SAKARYA
ÜNİVERSİTESİ
YAYINLARI

MÜHENDİSLİKTE YAPAY ZEKA VE UYGULAMALARI 4

EDİTÖRLER
PROF. DR. SEVINÇ GÜLSEÇEN
PROF. DR. MEHMEH MELIH İNAL
PROF. DR. ORHAN TORKUL
DOÇ. DR. MUHAMMED KÜRŞAD UÇAR

Sakarya 2022

SAÜ MEZUNLAR DERNEĞİ

MÜHENDİSLİKTE YAPAY ZEKA VE UYGULAMALARI 4

Sakarya 2022

MÜHENDİSLİKTE YAPAY ZEKA VE UYGULAMALARI 4

Editörler

Prof. Dr. Sevinç GÜLSEÇEN

Prof. Dr. Mehmet Melih İNAL

Prof. Dr. Orhan TORKUL

Doç. Dr. Muhammed Kürşad UÇAR

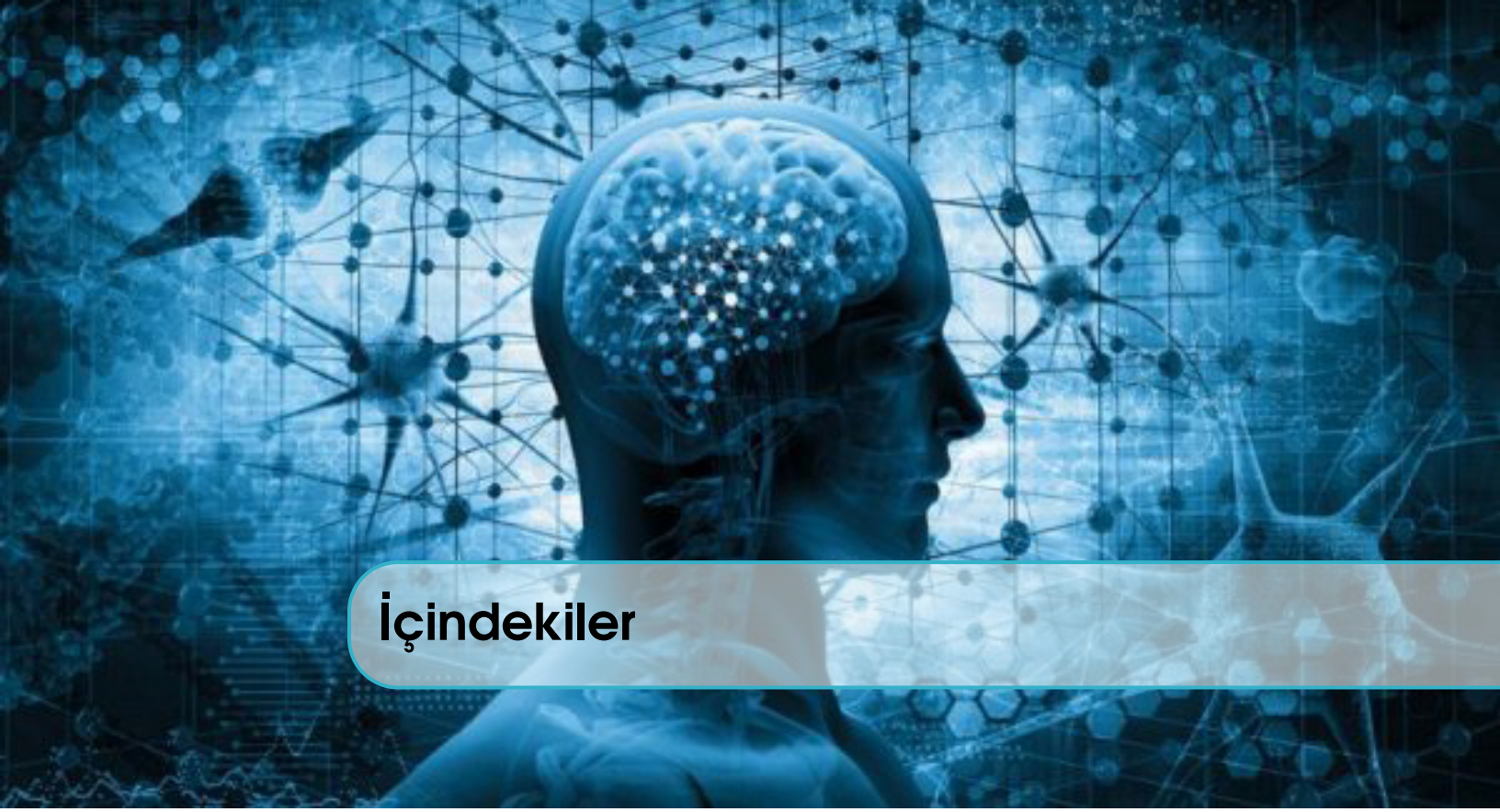
YAYIN NO: *230*
e-ISBN: *978-605-2238-57-8*
ALAN: *Mühendislik*

ADRES VE İLETİŞİM
Sakarya Üniversitesi, Bilimsel Yayınlar Koordinatörlüğü Esentepe / Sakarya / Serdivan
Tel: +90 264 295 7465
Fax: +90 264 295 5352
yayin@sakarya.edu.tr
www.sauyayinlar.sakarya.edu.tr

MÜHENDİSLİKTE YAPAY ZEKA VE UYGULAMLARI 4
EDİTÖRLER
Prof. Dr. Sevinç GÜLSEÇEN
Prof. Dr. Mehmet Melih İNAL
Prof. Dr. Orhan TORKUL
Doç. Dr. Muhammed Kürşad UÇAR

YAYIN EDİTÖRÜ
Doç. Dr. Mustafa GÜNERİGÖK
REDAKSİYON
Doç. Dr. Muhammed Kürşad UÇAR
DİZGİ
Doç. Dr. Muhammed Kürşad UÇAR
KAPAK TASARIM
Nukeloveer Studio
YAYIN TARİHİ
Aralık 2022

Gece destan yazan, sabahında devleti için hizmet eden kahraman milletimize ithaf olunur.



İçindekiler

ÖNSÖZ

Bir yıl aradan sonra sözümüze kaldığımız yerden devam ediyoruz. Bu yıl "Mühendislikte Yapay Zeka ve Uygulamaları 4" kitabı ile bir seriye devam etmek istiyoruz. Umarız ki bu tür hizmetler yetiştirdiğimiz öğrencilerimiz için faydalı olur ve her yıl bu kitabın devamını çıkarabiliriz.

Yapay Zeka Yaz Okulu (YAZSUM) ilk olarak 2017 yılında yüz yüze 88 farklı üniversiteden 550'den fazla katılımcı ile Sakarya Üniversitesi ev sahipliğinde gerçekleştirilmiştir. 2018 yılında detaylı içeriklerle bir kez daha hizmet etme fırsatı bulduk. 2020 yılında ise COVID-19 sebebiyle çevrimiçi platformları kullanarak 3500'den fazla katılımcı ile gerçekleştirdik. Eğitim kapsamında 96 saat eğitim verilmiştir. Bu rakam eğitmenlerimizi ve bizleri ziyadesiyle memnun etmiştir. 2021 yılında 6 yurtiçi 2 yurtdışı üniversite ortaklığıyla 3 günde 42 eğitimci ile 67 saatlik eğitim ile YAZSUM gerçekleştirildi.

Pandemi sürecinde teknolojik alt yapılarının önemi bir kez daha ortaya çıkmıştır. Bu süre zarfında sürece hazırlıklı olan kurum ve devletler ilerleyişini hız kesmeden devam ettirmektedir. Ülkemize ve kendimize ilim bakımından yatırım yapmak hayatımızın en önemli adımları olacaktır.

Elimizdeki bu kitap gerek teorik gerekse pratik uygulamalarla size yeni bir yol gösterici olmasını umuyoruz. Yapay zeka oldukça geniş bir konudur. Zifiri karanlıkta her tarafı aydınlatamasakta önümüzü görecek kadar kendimize ve çevremize ışık tutmayı umuyoruz.

Işığınızın hiç kaybolmaması dileğiyle.

Editörler
Aralık 2022

artificial intelligence

1. Topluluk Makine Öğrenmesi Yöntemleri

Topluluk Makine Öğrenmesi Yöntemleri ve Bir Uygulama

Deniz DEMİRCİOĞLU DİREN *

*Sakarya Üniversitesi, Uzaktan Eğitim Araştırma ve Uygulama Merkezi

1.1 Giriş

Öğrenme yeteneği, zekanın en temel özelliklerinden biridir. Öğrenme, bir ortamdaki deneyimden bilgi edinme yoluyla, bir çevredeki performansın iyileştirilmesi temeline dayanmaktadır. Bu yetenek, hem bilişsel psikoloji hem de yapay zeka için önemli bir ilgi alanı oluşturmaktadır (Langley, 1996). Yapay zekanın bir alt kümesi olan makine öğrenmesi, insanlarla aynı zekâ seviyesine sahip makineler geliştirmek için yoğun çalışmalar yapan bir daldır. Bu amaçla geliştirilen makine öğrenmesi, makinelerin örnek veri ya da geçmiş deneyimlerden öğrenerek karar almasını sağlayan zeki teknolojilerdir (Mohammed ve diğerleri, 2016; Bilgin, 2018). Makinelerin zeki olmasını sağlamak için bazı bilgisayar programları ve algoritmalar kullanılmalıdır. Bilgisayar biliminin diğer alanlarında olduğu gibi, bu algoritmaların performansı bazı kritik ölçütlere bağlıdır. Performansı arttırmak için algoritmalar verimli ve doğru tahmin yapacak şekilde tasarlanmalıdır (Mohri ve diğerleri, 2018).

Makine öğrenmesi asıl olarak bilgisayar bilimi ve istatistik alanlarının kesişimi ile oluşmaktadır. Genel bir bakış açısıyla, bilgisayar biliminin, problemleri çözen makinelerin nasıl yapıldığına, istatistik biliminin ise önceki verilerden nasıl ve hangi güvenilirlikle sonuçlar elde edilebileceğine odaklandığı söylenebilir. Makine öğrenmesi de bu iki bilim dalını temel alarak, bilgisayarların nasıl programlanarak hangi doğrulukla sonuçlar ürettiğine odaklanmaktadır. (Mitchell,2006).

Makine öğrenmesi çalışmalarında genellikle temel amaç an başarılı algoritmayı tespit ederek en iyi sonuçlara ulaşmaktır. Bu yöntemlerden biri de modellerin birleştirilmesi yani topluluk algoritmalarıdır (Alpaydın, 2017). Topluluk öğrenme fikri, birden fazla algoritmayı birlikte kullanarak onların tahminlerini bir araya getirmektir (Sewwell, 2008). Bu şekilde modelin başarısının artacağı düşünülmektedir. Çünkü tek bir hipotez üreten öğrenme algoritmaları üç temel sorunla karşı karşıya

kalmaktadır. Bunlar; (1) istatistiksel problem (2) hesaplama problemi ve (3) temsil problemidir. İstatistiksel problem ile, öğrenme algoritması, mevcut eğitim verilerinin miktarı için çok büyük bir hipotez alanını aradığı zaman karşılaşılır. Hesaplama problemi, öğrenme algoritması hipotez uzayında en iyi hipotezi bulmayı garanti edemediği zaman ortaya çıkmaktadır. Temsil problemi, hipotez uzayı gerçek fonksiyona iyi yaklaşımlar olan herhangi bir fonksiyon içermediği durumlarda oluşmaktadır. Bahsedilen bu sorunların topluluk algoritmaları ile çözülebileceği ifade edilmektedir (Dietterich, 2000). Bunların yanında topluluk algoritmalarını oluştururken dikkat edilmesi gereken konular vardır. Bunlar;

- Gruptaki bireysel sınıflandırıcılar arasındaki ara bağlantıların dikkate alınması
- Topluluk için çeşitli ve tamamlayıcı bireysel sınıflandırıcılardan oluşan bir havuz seçilmesi
- Topluluğu oluşturan sınıflandırıcıların güçlü yönlerinden faydalanılması
- Topluluğun nihai kararından sorumlu bir kombinasyon kuralının önerilmesi

olarak belirtilebilir (Krawczyk ve diğerleri, 2017).

Bu bölümde ilk olarak makine öğrenmesi ile ilgili temel kavramlar anlatılmaktadır. Daha sonra topluluk makine öğrenmesi algoritmalarının temel kavramları, amaç ve hedefleri, yaklaşımları, çeşitleri ve sağladığı faydalardan bahsedilmektedir. Konunun daha iyi anlaşılması açısından örnek bir veri seti üzerinde uygulama sunulmaktadır.

1.2 Makine Öğrenmesi

1.2.1 Makine Öğrenmesinin Hedefleri

Makine öğrenmesinin hedefleri araştırmacılara göre değişiklikler gösterse de Langley (1996) tarafından temel olarak dört farklı hedef belirlenmiştir.

İlk hedef insan öğrenmesine vurgu yapan mekanizmaları modellemektir. Bu çerçevede, genel olarak insan bilişsel mimarisinin bilgisiyle tutarlı olan öğrenme davranışlarını açıklamak için tasarlanmış öğrenme algoritmaları geliştirilmeye çalışılmaktadır. Bu kapsamda, problem çözme, doğal dil işleme, metin ve belge sınıflandırma, algı ve motor kontrol, öğrenme davranışını tahmini, öğretim materyallerinin tasarımı ve hastalık teşhisi gibi konulara çözüm üretmek hedef alınmaktadır.

İkinci hedef, öğrenme algoritmalarının özelliklerini ilişkilendiren genel ilkeleri keşfetmek ve bu manipülasyonun öğrenme üzerindeki etkisini gözlemlemektir. Bunu gerçekleştirmek için farklı algoritmaları karşılaştırmanın yanında tek bir algoritmanın farklı varyasyonları denenerek incelenebilir. Bazı deneyler doğal alanlardaki davranışı dikkate alırken, bazıları sentetik alanların özelliklerini sistematik olarak değiştirmektedir.

Makine öğrenmesinin bir diğer başarısı matematiksel çalışma alanı olarak ele alınmasıdır. Bu alandaki hedef öğrenme problemlerini çözmek için geliştirilmiş algoritmaların teoremlerini formüle etmek ve kanıtlamaktır.

Son hedef olarak ise makine öğrenmesinin gerçek hayat problemlerinde uygulanması yaklaşımıdır. Bu yaklaşım, tanılama, süreç kontrolü, zamanlama ve diğer alanlarda makine öğrenmesinin saha uygulamalarını ortaya çıkartmıştır.

1.2.2 Makine Öğrenmesi Sürecinin Akışı

Makine öğrenmesi sürecinde veri setine göre farklı ve spesifik uygulamalar olmaktadır. Farklı tahmin modelleri farklı süreçlerle yönetilebilir ancak her modelde gerçekleştirilen işlemlerin bazıları ortaktır. Makine öğrenmesi süreci Şekil 1.1'de gösterilen altı temel adımı içermektedir (Lantz, 2019).

Makine öğrenmesi süreci, problemin belirlenmesi ve seçimi ile başlamaktadır. Problem belirlendikten sonra, ortamdan probleme ait veriler toplanmaktadır. Bu veriler algoritmalar tarafından

Şekil 1.1: Makine Öğrenmesi Sürecinin Akışı

bilgi üretmek için kullanılmaktadır. Veriler ne kadar düzgün olursa üretilen bilgilerin de o derece doğru olması beklenmektedir. Bu nedenle, toplanan verilerdeki aykırı ve eksik değerlerin temizlenmesi, normalizasyon, veri dönüştürme gibi işlemleri içeren veri önişleme adımı gerçekleştirilmektedir. Veri ön işleme adımı sadece makine öğrenmesinde değil veri ile ilgili yapılacak her türlü analizde çok önemlidir. Çünkü verideki gürültülü örnekler sonuç üzerinde istenmeyen durumlara neden olabilir. Ayrıca veri analisti için anlamlı olmayan değişkenlerin, veri ile ilgili alan uzmanı tarafından dönüştürülmesi, verinin ve sonuçlarını anlamlı şekilde yorumlanmasını sağlayacaktır. Veri önişleme adımından sonra veri, analiz için hazır duruma gelmektedir. Veriler analiz için hazırlandığında, verilerden ne öğrenilmesi beklendiği ile ilgili fikir elde edilmiş olmaktadır. Spesifik makine öğrenmesinin görevi, uygun bir algoritmanın seçimini belirleyerek verileri bir model biçiminde temsil etmektir. Bundan sonraki adımda model eğitimi başlamaktadır. Model eğitilirken veri eğitim ve test olarak ayrılmaktadır. Algoritma eğitim verisi ile mevcut durumları öğrenerek daha önce karşılaşmadığı test verisine karşılık sonuç üretmektedir. Algoritmanın deneyimlerden ne kadar iyi öğrendiği yani sonuçların doğruluğunun değerlendirilmesi gerekmektedir. Bu adımda önemli nokta da değerlendirme kriterinin seçimidir. Model iyileştirme adımında ise modelin performansını artırmak için daha gelişmiş stratejiler kullanmaktadır. Bunun için de bazen ilave veri toplanması, bazen veri önişleme adımlarının gözden geçirilmesi bazen de yeni bir model ya da teknik gerekebilmektedir. Bu işlemler yapıldıktan sonra modelin performansı yeterli bulunursa model uygulamaya hazır olacaktır.

1.2.3 Makine Öğrenmesi Yaklaşımları

Bir problemin makine öğrenmesi ile çözümü için kullanılabilecek yaklaşımlar Şekil 1.2'deki gibi sınıflandırılabilir (Sarker, 2021). Bu yaklaşımlar veri setinin hedef değerinin olup olmamasına ve hedef değerin türüne göre farklılaşmaktadır. Standart öğrenme görevleri aşağıdaki gibi sınıflandırılmaktadır (Mohri ve diğerleri, 2018).

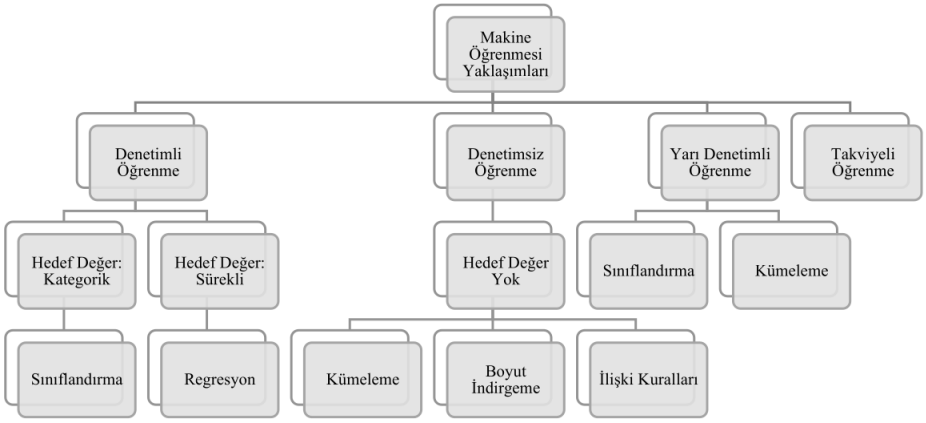
- Sınıflandırma
- Regresyon
- Sıralama
- Kümeleme
- Boyut azaltma

Daha açıklayıcı olmak için problem bir süreç olarak düşünülebilir. Süreç, Şekil 1.3'deki gibi girdi, işlem ve çıktı bileşenlerinden oluşmaktadır. Girdi, probleme etki eden bağımsız değişkenler, işlem makine öğrenmesi yöntemleri ile çözülecek olan model, çıktı ise problemin hedef yani bağımlı değişkenidir.

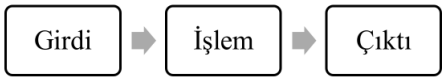
Yaklaşımlar ilk olarak çıktının olup olmamasına göre farklılaşmaktadır.

Buna göre;

1. Probleme ait veri setindeki örneklerin, girdi değişkenlerine karşılık çıktı değeri mevcutsa; yani algoritmalara hangi durumda hangi sonucun elde edileceği öğretiliyorsa, bu durumda denetimli ya da gözetimli öğrenme kullanılmalıdır (Nilsson, 1996)
2. Probleme ait veri setinde örneklerin, girdi değişkenlerine karşılık çıktı değeri mevcut değilse;



Şekil 1.2: Makine Öğrenmesi Sürecinin Akışı



Şekil 1.3: Makine Öğrenmesi Sürecinin Akışı

yani sisteme sadece giriş değişkenleri verilerek algoritmaların örneklerin benzerlikleri göre öğrenerek çıkarımlar yapması bekleniyorsa; bu durumda denetimsiz ya da gözetimsiz öğrenme kullanılmalıdır. Sisteme yeni veriler tanıtıldığında, verilerin sınıfını tespit etmek için önceden öğrenilen özellikler kullanılmaktadır (Mahesh, 2020)

3. Probleme ait veri setinde örneklerden az sayıda örneğin girdi değişkenlerine karşılık çıktı değeri bulunurken büyük çoğunluğunda bulunmuyorsa bu gibi durumlarda yarı-denetimli öğrenme kullanılmalıdır. Bu yaklaşımın amacı etiketli ve etiketsiz verilerin birleştirilmesinin öğrenme davranışını nasıl değiştirebileceğini anlamak ve böyle bir kombinasyondan yararlanan algoritmalar tasarlamaktır (Zhu ve Goldberg, 2009).
4. Problemde bir veri seti bulunmuyorsa, öğrenme dinamik bir ortamda deneme-yanılma etkileşimleri yoluyla davranışı öğrenen bir aracı ile gerçekleşiyorsa, bu durumda takviyeli öğrenme yaklaşımı kullanılmaktadır. Bu yaklaşımda iki ana strateji vardır. Birincisi, çevrede iyi performans gösteren birini bulmak için genetik algoritmalar ve genetik programlama gibi yöntemler ile arama yapmaktır. İkincisi, harekete geçmenin yararlarını tahmin etmek için istatistiksel teknikleri ve dinamik programlama yöntemlerini kullanmaktır (Kaelbling ve diğerleri, 1996).

Yaklaşımlar çıktı değişkeninin varlığına göre değerlendirildikten sonra çıktının türüne göre de değerlendirilmelidir. Çıktı türünden kastedilen çıktı değişkeninin sürekli ya da kategorik olmasıdır. Buna göre;

1. Denetimli öğrenme kullanılıyorsa ve çıktı değeri kategorikse bu durumda problemde sınıflandırma yaklaşımı kullanılmalıdır. Sınıflandırma problemlerinde kullanılabilecek algoritmalara örnek olarak k-en yakın komşu, naive bayes, destek vektör makineleri, karar ağacı vb verilebilir (Sarker, 2021).
2. Denetimli öğrenme kullanıyorsa ve çıktı değeri sürekli ise bu durumda problemde regresyon yaklaşımı kullanılmalıdır. Regresyon problemlerinde kullanılabilecek algoritmalar örnek olarak lineer regresyon, karar ağacı regresyonu, yapay sinir ağı regresyon vb verilebilir

(Sarker, 2021).

3. Denetimsiz öğrenme kullanıyorsa bu durumda hedef değer yoktur. Problemin ihtiyacına göre kümeleme, boyut indirgeme ya da ilişki kuralları kullanılabilir (Mahesh, 2020).
4. Yarı denetimli öğrenme kullanılıyorsa bu durumda hedef değişkeninin olduğu ve olmadığı örnekler birlikte olduğu için bu durumda hem sınıflandırma hem de kümeleme algoritmaları kullanılabilmektedir (Mohri ve diğerleri, 2018).

1.2.4 Makine Öğrenmesinde Performans Değerlendirme

Geliştirilmiş birçok farklı algoritma mevcuttur bu nedenle problem için en uygun ve başarılı algoritmanın hangisi olduğunu seçmek bazen çok karmaşık olabilmektedir. Bir modeli eğitildiğinden farklı veriler üzerinde test etmeyi amaçlayan performans değerlendirme, bu karmaşıklığı çözmek için önemli bir yol göstericidir. Ayrıca algoritmaların gelecekteki durumlarda öngörücü performansının ya da öğrenme performansının tarafsız bir tahminini sağlamaktadır (Raschka, 2018). Makine öğrenmesi algoritmalarının performansını değerlendirmek için kullanılan bazı kriterler bulunmaktadır. Hangi kriterin kullanılacağı problemin amacına göre temel olarak aşağıda sıralanan iki maddeye göre farklılık göstermektedir (Zheng, 2015).

- Sınıflandırma probleminde doğruluklarla ilgili ölçümler yapılmaktadır. Bunun için; doğruluk, duyarlılık, kesinlik, F ölçütü ya da kappa istatistiği gibi kriterler kullanılmalıdır.
- Regresyon probleminde hatalar ile ilgili ölçümler yapılmaktadır. Bunun için; ortalama kare hatası, karekök ortalama hata, mutlak hata ya da ortalama mutlak yüzde hata gibi kriterler kullanılmalıdır.

1.2.5 Sınıflandırma Performans Kriterleri

Sınıflandırma problemlerinde kullanılan algoritmanın başarısının tahmini için sıklıkla kullanılan araç Tablo 1.1. ile gösterilen hata matrisidir (Bilgin, 2018).

Tablo 1.1: Hata matrisi

	Gerçek Pozitif Sınıf	Gerçek Negatif Sınıf
Tahmin Pozitif Sınıf	GP	YN
Tahmin Negatif Sınıf	YP	GN

Burada;

- GP: test kümesindeki gerçek pozitif örnek sayısı,
- GN: test kümesindeki gerçek negatif örnek sayısı,
- YP: test kümesindeki yanlış pozitif örnek sayısı,
- YN: test kümesindeki yanlış negatif örnek sayısı

olarak ifade edilmektedir.

Doğruluk, hata oranı, kesinlik, duyarlılık ve F ölçütü ise hata matrisinden elde edilerek hesaplanabilecek ölçütlerdir. Ölçütlerin hesaplamaları sırasıyla aşağıda gösterilmektedir (Hossin ve Sulaiman, 2015);

Doğruluk: Doğru tahminlerin değerlendirilen toplam örnek sayısına oranını ölçmektedir. Eşitlik 1'deki gibi hesaplanmaktadır.

$$Doğruluk = \frac{t_p + t_n}{t_p + f_p + t_n + f_n} \tag{1.1}$$

Hata oranı: Yanlış tahminlerin değerlendirilen toplam örnek sayısına oranını ölçmektedir. Eşitlik 2'deki gibi hesaplanmaktadır.

$$HataOranı = \frac{f_p + f_n}{t_p + f_p + t_n + f_n} \tag{1.2}$$

Kesinlik: Pozitif bir sınıftaki toplam tahmin edilen modellerden doğru bir şekilde tahmin edilen pozitif kalıpları ölçmek için kullanılmaktadır. Eşitlik 3'deki gibi hesaplanmaktadır.

$$Kesinlik = \frac{t_p}{t_p + f_p} \tag{1.3}$$

Duyarlılık: Doğru bir şekilde sınıflandırılan pozitif kalıpların oranını ölçmek için kullanılmaktadır. Eşitlik 4'deki gibi hesaplanmaktadır.

$$Duyarlılık = \frac{t_p}{t_p + t_n} \tag{1.4}$$

F Ölçütü: Bu metrik, duyarlılık ve kesinlik değerleri arasındaki harmonik ortalamayı temsil etmektedir. Eşitlik 5'deki gibi hesaplanmaktadır.

$$FÖlütü = \frac{2 \times p \times r}{p + r} \tag{1.5}$$

Burada, her bir veri sınıfı; $t_(p_i) : C_i$ için doğru pozitif; $f_(p_i) : C_i$ için yanlış pozitif; $f_(n_i) : C_i$ için yanlış negatif; $t_(n_i) : C_i$ için doğru negatifdir.

Kappa istatistiği: Yalnızca şans eseri doğru bir tahmin olasılığını hesaba katarak değerlendirme yapan istatistik, iki değişken arasındaki anlaşmanın derecesini ölçer, çoğunlukla bir ölçümün değil, bir yargının sonucu olan verilerle ilgilenmektedir. Eşitlik 6 daki gibi hesaplanan değer, tahmin ve gerçek değerler arasındaki uyuma göre 1 değerine kadar değişmektedir. Mükemmel bir uyumu ifade eden 1 değerine ulaşmak çoğu zaman zor olmaktadır (Lantz, 2019).

Kappa istatistiği Eşitlik 6.daki gibi hesaplanmaktadır.

$$k = (Pr(a) - Pr(e))/(1 - Pr(e)) \tag{1.6}$$

Burada $Pr(a)$ ve $Pr(e)$ sırasıyla gerçek ve beklenen değerlerin, sınıflandırıcı ve gerçek değerler arasındaki uyum oranını ifade etmektedir.

Kappa değerleri, yaygın olarak şu şekilde yorumlanmaktadır (Lantz, 2019).

- Kötü tahmin = 0,20'den az
- Makul düzey tahmin = 0,20 – 0,40
- Orta düzey tahmin = 0,40 ila 0,60
- İyi tahmin = 0,60 ila 0,80
- Çok iyi tahmin = 0,80 – 1,00

1.2.6 Regresyon Performans Kriterleri

Ortalama Kare Hatası: Tahmin edilen çözümler ile istenen çözümler arasındaki farkı ölçmektedir. İyi eğitilmiş bir algoritmanın ortalama kare hatası (MSE) değerinin düşük olması beklenmektedir. MSE Eşitlik 7. deki gibi hesaplanmaktadır.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{1.7}$$

Burada y_i , i örneğinin tahmin edilen değeridir, $\hat{y}_i$, i örneğinin gerçek hedef değeridir ve n, toplam örnek sayısıdır (Hossin ve Sulaiman, 2015).

Karekök Ortalama Hata: Model performansını ölçmek için kullanılan standart bir istatistiksel ölçü olarak kullanılan bu değer MSE'nin kareköküdür (Pham,2019). Karekök ortalama hatası (RMSE) Eşitlik 8.deki gibi hesaplanmaktadır.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n-k}} \tag{1.8}$$

Ortalama Mutlak Hata: Düzgün dağılmış hataları tanımlamayı sağlayan ortalama mutlak hata (MAE) Eşitlik 9.daki gibi hesaplanmaktadır. Ancak iyi istatistik ölçütleri, yalnızca bir performans ölçüsü değil, aynı zamanda hata dağılımının bir temsilini de sağlamalıdır. Bu nedenle Model hatalarının tek tip bir dağılımdan ziyade normal bir dağılıma sahip olması muhtemel olduğundan, RMSE, bu tür bir veri için MAE'den daha iyi bir ölçüm sunmaktadır.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i) \tag{1.9}$$

Modelin performansı değerlendirilip istenen düzeyde bulunmazsa performansı yükseltmek için bazı çözüm yaklaşımları mevcuttur. Bu yaklaşımlar:

- Daha iyi öğrenme teknikleri tasarlamak
- Eğitim verileri üzerinde bir tür dönüşüm uygulamak
- Sınıflandırıcı tarafından verilen tahminleri değiştirmek veya ayarlamak

şeklinde gruplandırılabilir (Silva-Palacaios ve diğerleri, 2017). İlk grupta yer alan ve birden fazla algoritmayı birlikte kullanma temeline dayanan topluluk algoritmalarının kullanımı, son yıllarda artarak güncel bir çalışma alanı olarak karşımıza çıkmaktadır (Re ve Valentini, 2012). Bu nedenle uygulamada topluluk algoritmaları incelenmiştir.

1.3 Topluluk Algoritmaları

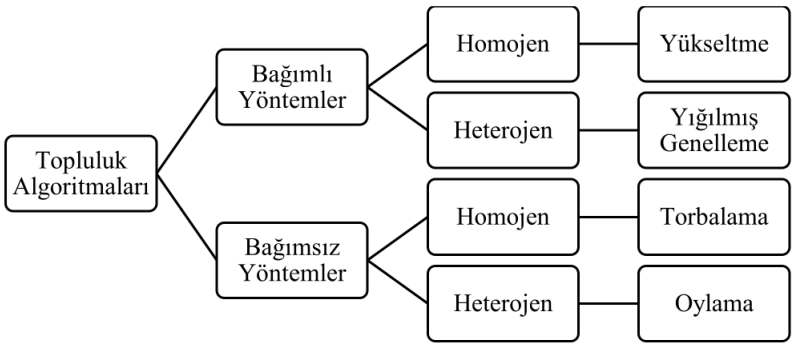
Topluluk algoritmaları, herhangi bir önemli karar vermeden önce çeşitli görüşler aramaya yönelik insan sosyal öğrenme davranışını anımsatan istatistiksel ve hesaplamalı öğrenme prosedürleridir (Matteo ve Giorgio, 2012). Topluluk algoritmaları ilk olarak bireysel algoritmaların ya da karar vericilerin değişkenliğini azaltmak ve dolayısıyla performansı arttırmak için geliştirilmiş olsa da özellik seçimi, güven tahmini, eksik özellik, artımlı öğrenme gibi çeşitli makine öğrenmesi sorunlarına çözüm üretmek için de kullanılmaktadır. Genel olarak sınıflandırıcıların olası iki hatası

bulunmaktadır. Birincisi önyargı yani sınıflandırıcının doğruluğu, ikincisi ise varyans yani sınıflandırıcının kesinliğidir. Topluluk sistemlerinin amacı, nispeten sabit (veya benzer) önyargıya sahip birkaç sınıflandırıcı oluşturmak ve daha sonra varyansı azaltmak için çıktılarını ortalama alarak birleştirmektir. Değişkenliğin azaltılması, sinyalin her örneğinin etrafındaki bir komşu örnek tarafından ortalamasının alındığı, yüksek frekanslı (yüksek varyanslı) gürültünün azaltılması olarak düşünülebilir (Zhang ve Ma, 2012)

Birçok araştırmacı topluluk algoritmaları ile ilgili araştırmalar yaparak topluluk algoritmalarının, öğrenme problemlerinde algoritmaların bireysel olarak kullanılmasından daha güvenilir ve daha doğru tahminler elde etmesini sağlayan makine öğrenmesi kümeleri olduğu sonucuna varmışlardır (Wolpert 1992; Breiman 1996; Alpaydın 2017). Sınıflandırıcı toplulukları kullanmanın ana motivasyonu, tüm görevler için uygun tek bir sınıflandırıcı olmaması gerçeğidir. Genellikle belirli bir sorunu çözmek için elde bir sınıflandırıcı havuzu bulunmaktadır. Bunlarla inşa edilen topluluk yöntemleri yeni gelen örnekleri tahmin etmek için tahminleri birleştirilen bir dizi bireysel bileşen sınıflandırıcıdır. Bu yöntemlerin, tahmin doğruluğunu arttırmayı ve/veya karmaşık bir öğrenme problemini alt problemlere ayırmayı sağladığı belirtilmektedir (Krawczyk, 2017). Ayrıca topluluk yapısının genelleme yeteneği sayesinde bir algoritmanın tek olarak kullanılmasından daha güçlü olması beklenmektedir. Tabi ki bu öngörü kesinlik ifade etmemektedir (Zhou, 2012). Bunula birlikte birkaç sınıflandırıcının çıktısını birleştirmek, yalnızca fikir ayrılıkları mevcutken yararlı olmaktadır. Açıkçası, birkaç özdeş sınıflandırıcıyı birleştirmenin bir fayda sağlaması öngörülmemektedir (Maclin ve Opitz, 1997).

1.3.1 Topluluk Algoritmalarının Çeşitleri

Topluluk algoritmaları geliştirilirken kurulum prensibine göre Şekil 1.4'deki gibi farklılaşmaktadır. Temel olarak öncelikle algoritmaların birleştirilme prensibi ele alınmalıdır. Bu açıdan iki farklı yaklaşım bulunmaktadır. Birincisi algoritmaların sıralı olarak bağlanması ikincisi ise algoritmaların birbirine paralel olarak bağlanmasıdır. Sıralı bağlanma bağımlı, paralel bağlanma ise bağımsız olarak isimlendirilebilir. Algoritmaların bağlanma prensipleri dışında bir de bağlanan algoritmaların türüne göre değişkenlik mevcuttur. Bu da aynı algoritmaların birleştirilmesi ya da farklı algoritmaların birleştirilmesi olarak ikiye ayrılmaktadır. Eğer aynı algoritmalar birleştiriliyorsa homojen, farklı algoritmalar birleştiriliyorsa bu durumda da heterojen olarak adlandırılmaktadır (Zhou 2012; Gowda 2018).



Şekil 1.4: Topluluk Algoritmalarının Çeşitleri

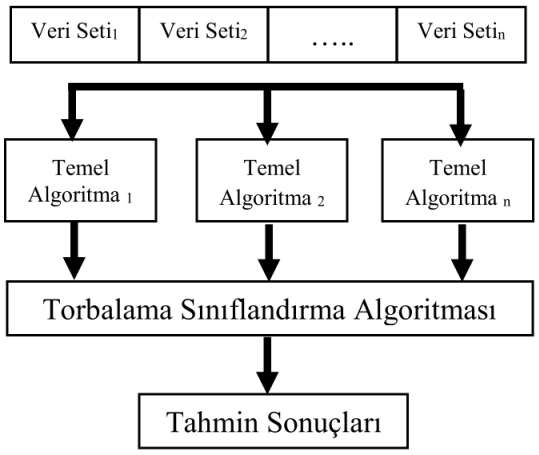
Orijinal topluluk yöntemi Bayes ortalamasıdır, ancak daha yeni algoritmalar torbalama, yükseltme ve yığılmış genellemeyi içermektedir (Dietterich, 2000). Şekil 1.4'de popüler topluluk

algoritmaları ve oluşturulma prensipleri gösterilmiştir. Topluluk algoritmalarının türleri, kullanılan bireysel algoritmaların homojen (aynı) ya da heterojen (farklı) olmasına ve bu algoritmaların bağımlı (sıralı) ya da bağımsız (paralel) olmasına göre değişmektedir.

1.3.2 Torbalama Topluluk Yöntemi

Breiman (1994) tarafından geliştirilen torbalama algoritması, aynı algoritmayı birden fazla kez birleştirerek bir model üreten en basit ve eski topluluk yöntemlerindendir (Breiman, 1994). Temel birleştirme prensibi olarak Şekil 1.5'de gösterildiği gibi algoritmaların paralel olarak bağlanmasını benimseyen bu yöntemde, kararlar oylama ile verilmektedir (Gowda ve diğerleri,2018).

Torbalama algoritmasının öğrenme sistemi hakkında farklı varsayımlar mevcuttur. Örneğin torbalama algoritması, öğrenme sisteminin "kararlı" olmamasını gerektirir, böylece eğitim setindeki küçük değişiklikler farklı sınıflandırıcılara rehberlik edebilmelidir. Bunun yanında Breiman "zayıf tahmin edicilerin torbalama yoluyla daha kötü tahminlere dönüştürülebileceğini" de belirtmiştir (Quinlan, 1996).



Şekil 1.5: Torbalama Yöntemi (Maclin ve Opitz,1997)

Tablo 1.2'de torbalama yönteminin adımları verilmektedir. Bu adımlara göre, torbalama yöntemi, her sınıflandırıcıyı eğitim setinin rasgele yeniden dağıtımıyla eğiterek topluluk için bireyler oluşturan bir yöntemdir. Her sınıflandırıcının eğitim seti, S örnekle değiştirilerek rastgele çizilerek oluşturulur (burada S orijinal eğitim setinin boyutudur); orijinal örneklerin çoğu sonuçtaki eğitimde tekrarlanarak ayarlanır, diğerleri dışarıda bırakılabilir. Topluluktaki her bir bireysel sınıflandırıcı, eğitim setinin farklı bir rastgele örneklemesi ile oluşturulur (Maclin ve Opitz, 1997).

1.3.3 Yükseltme Topluluk Yöntemi

Bir diğer topluluk yöntemi olan yükseltme algoritması sıralı bağlanma prensibine göre aynı algoritmaların birlikte kullanıldığı bir yöntemdir. Bu yöntemde Şekil 1.6'da gösterildiği gibi her algoritma bir önceki algoritmanın çıktısını girdi olarak kullanmaktadır (Lantz, 2019). Algoritmalar kendinden önceki algoritmanın performansından etkilenmektedir. Çünkü bir önceki sınıflandırma da yapılan hatalara daha fazla ağırlık vermektedir. Bu şekilde mevcut grup performansının zayıf olduğu örnekleri daha doğru bir şekilde tahmin edebilen yeni sınıflandırıcılar üretmeye çalışılmaktadır. Böylece hataları azaltmak ve performansı yükseltmek hedeflenmektedir (Rokach, 2010).

Tablo 1.2: Torbalama Algoritması

Algoritma: Torbalama

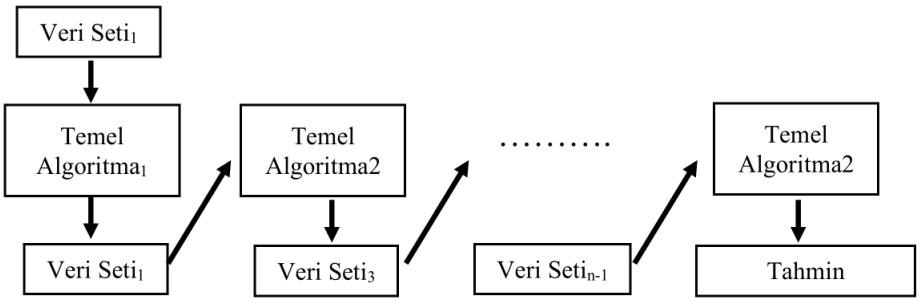
Girdiler: Eğitim seti S; denetimli öğrenme algoritması, ***TemelSınıflandırıcı***
topluluk boyutunu belirten T tamsayısı; önyüklenmiş eğitim verileri oluşturmak için yüzde R
Tekrar yap $t = 1, ..., T$

1. S'nin $\%R$'sini rastgele çizerek önyüklenmiş bir S_t kopyası alın
2. ***TemelSınıflandırıcı*** S_t ile çağırın ve h_t hipotezini (sınıflandırıcı) elde edin.
3. Topluluğa h_t ekleyin, $\varepsilon \leftarrow \varepsilon \cup h_t$

Son
Topluluk Kombinasyonu: Basit Çoğunluk Oylaması- Verilen etiketlenmemiş örnek x

1. x üzerinde $\varepsilon = \{h_1, ..., h_t\}$ grubunu değerlendir
2. Eğer $h_t \omega_c$ sınıfını seçerse, $v_{t,c} = 1$, değilse 0.
3. Her sınıfın aldığı toplam oyu elde edin.

$$V_c = \sum_{t=1}^{T} v_{t,c}, c = 1, ..., C$$



Şekil 1.6: Yükseltme topluluk yöntemi

Yükseltme algoritmasında her bir sınıflandırıcı için kullanılan eğitim seti, bir önceki sınıflandırıcının performansına bağlı iken torbalama algoritmasında eğitim setinin yeniden örneklenmesi önceki sınıflandırıcının performansından bağımsızdır (Maclin ve Opitz,1997). Genellikle çalışmalarda yükseltme yönteminin en bilinen türü olan Adaboost kullanılmaktadır (Zhou, 2012)

Tablo 1.3: Yükseltme algoritması

Algoritma2. Yükseltme

Girdi: Eğitim verisi =) $\{x_i, y_i\}, i = 1, ..., N, y_i \in \{\omega_1, ..., \omega_2\}$ denetimli öğrenme
TemelSınıflandırıcı; topluluk boyutu T.
Başlangıç Değeri: $D_i(i) = {}^1/_N$
Tekrar et $t = 1, 2, ..., T$:

1. D_t dağılımından S_t eğitim alt kümesini oluşturun.
2. **TemelSınıflandırıcıyı** S_t üzerinde eğit, $h_t : X \to Y$
3. h_t nin hatasını hesapla:

$$\varepsilon_t \sum_i I\left[[h_t(x_i \neq y_i)]\right] D_i(x_i)$$

Eğer $\varepsilon_t > {}^1/_2$ **iptal et**

4. Ata

$$\beta_t = {}^{\varepsilon_t}/_{(1-\varepsilon_t)}$$

5.Örnekleme dağılımını güncelle

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \begin{cases} eğer\, h_t(x_i) = y_i, \beta_t \\ değilse, 1 \end{cases}$$

burada $Z_t = \sum_i D_t(i)$, D_{t+1}'nin uygun bir dağılım fonksiyonu olmasını sağlamak için bir normalleştirme sabitidir.

Son
Ağırlıklı çoğunluk oylaması: Etiketlenmemiş z örneği verildiğinde, her sınıfın aldığı toplam oyu alın.

$$V_c = \sum_{t:h_t(z)=w_c} \log\left(\frac{1}{\beta_t}\right), c = 1, ..., C$$

Çıktı: En yüksek V_c'ye sahip sınıf.

Bu yöntem de torbalama gibi, orijinal N eğitim örneklerinden olasılıksal olarak (yerine koyma ile) örnekler seçerek T+1 sınıflandırıcısı için N boyutunda bir eğitim seti seçmektedir. Ancak torbalamadan farklı olarak, bir örnek seçme olasılığı eğitim setinde eşit değildir. Bu olasılık, o örneğin önceki T sınıflandırıcıları tarafından ne sıklıkla yanlış sınıflandırıldığına bağlıdır.

Tablo 1.3. de yükseltme yönteminin adımları verilmektedir. Bu adımlara göre, başlangıçta her bir örneğin seçilme olasılığı $1/N$ olarak belirlenir. Daha sonra, topluluğa her eğitilmiş sınıflandırıcı eklendikten sonra bu olasılıkları yeniden hesaplanır. ε_t, bir sonraki deneme sınıflandırıcısı T için yanlış sınıflandırılmış örnek olasılıklarının toplamı olsun. Bir sonraki deneme için olasılıklar, T'nin yanlış sınıflandırılan örneklerinin olasılıklarının β_t faktörü ile çarpılması ve ardından bu olasılıkların toplamları 1'e eşit olacak şekilde yeniden normalleştirilmesiyle üretilir. Yöntem $C_1, ..., C_t$ sınıflandırıcılarını, C_t'nin $\log(\beta_t)$ ağırlığına sahip olduğu ağırlıklı oylama kullanarak birleştirir.

1.3.4 Oylama

Farklı algoritmaların birleştirilmesi ile oluşturulan bu algoritmada Tablo 1.4'de sunulan bazı birleştirme kuralları mevcuttur.

Tablo 1.4: Oylama birleştirme kuralları

Kural	Birleştirme Fonksiyonu
Toplam	$y_i = \sum_{j=1}^{L} d_{ij}$
Ağırlıklı Toplam	$y_i = \sum_j w_j d_{ji}$ burada; $w_j \geq 0$, $\sum_j w_j = 1$
Toplam	1
Ortanca	$y_i = medyan_j d_{ji}$
En Küçük	$y_i = \min_j d_{ji}$
En Büyük	$y_i = \max_j d_{ji}$
Çarpım	$y_i = \prod_j d_{ij}$

Bu kurallar; oybirliği, oy çokluğu ya da çoğunluk oyu şeklinde olabilir. Oy birliğinde, tüm sınıflandırıcılar aynı kararı vermektedir. Sınıflandırıcıların yarısından fazlası aynı kararı verdiğinde oy çokluğu ile karar verilmiş olur. Çoğunluk oyunda ise karar en fazla oy alandan yana kullanılır. Bunların yanında, bazı sınıflandırıcıların diğerlerinden daha doğru olma ihtimalinin olduğu biliniyorsa böyle bir durumda bu sınıflandırıcıya daha fazla ağırlık vermek iyi performanslar sağlayabilir (Polikar,2012). Bu yöntemde çıktılar genel olarak Eşitlik.10'daki gibi birleştirilebilir (Alpaydın, 2017).

$$y_i = \sum_j w_j d_{ji} \tag{1.10}$$

Burada, $d_j i$ değeri, j modelinin i. sınıf için verdiği oy, w_j değeri oyun ağırlığıdır. $w_j \geq 0$ ve $\sum_j w_j = 1$'dir.

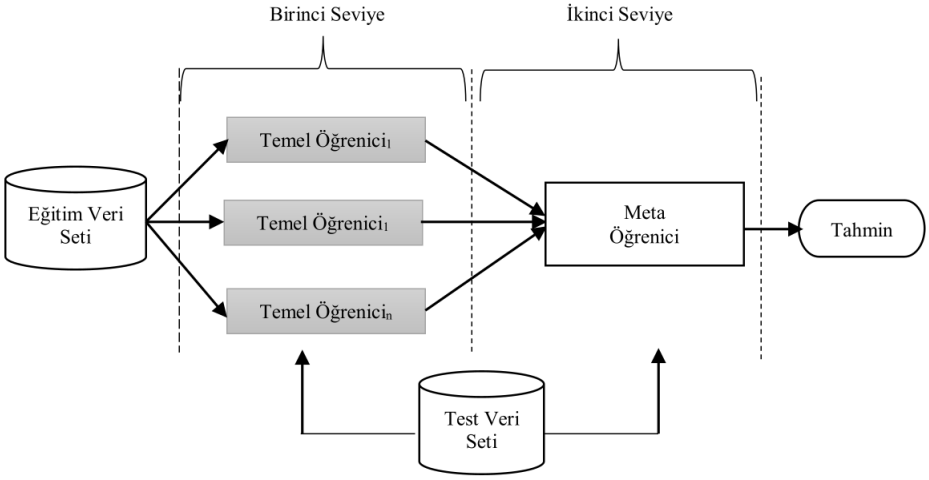
1.3.5 Yığılmış Genelleme Topluluk Yöntemi

Wolpert (1992) tarafından ayrı türde öğrenme algoritmalarının güçlerini bir araya getirmek için geliştirilmiş olan bu yöntem, meta-öğrenme tabanlıdır. Şekil 1.7'de görüldüğü gibi iki seviyeli çalışma prensibi olan yöntemde, birinci seviye temel öğrenici ikinci seviye ise meta öğrenici olarak isimlendirilmektedir (Zhou, 2012). Meta öğrenici temel öğrenicilerin tahminleri ile eğitilmektedir. Yani meta öğrenici birinci seviyedeki temel öğrenicilerin oluşturduğu veri setinden öğrenerek tahminde bulunmaktadır (Onan, 2018). Son sınıflandırma tahmininde hataların azaltılması amaçlanmaktadır (Oza, 2008).

Tablo 1.5'de sunulan yığılmış genelleme yöntemi üç ana adımı içermektedir (Tang ve diğerleri, 2014). Bunlar,

Adım1: Orijinal eğitim veri setine dayalı olarak, birinci seviye sınıflandırıcıları öğren. Temel sınıflandırıcıları öğrenmek için birkaç seçenek mevcuttur. Bunlar aşağıdaki gibidir;

- Bağımsız sınıflandırıcıları öğrenmek için yeniden örnekleme tekniği uygulanabilir.
- Yükseltme yönteminde kullanılan stratejiler uygulanabilir, örneğin ağırlık dağılımına uygun veriler üzerinde adaptif öğrenme tabanlı sınıflandırıcılar
- Çeşitli temel sınıflandırıcılar (homojen sınıflandırıcılar) oluşturmak için bir öğrenme algoritmasındaki parametreler ayarlanabilir.
- Temel sınıflandırıcılar (heterojen sınıflandırıcılar) oluşturmak için farklı sınıflandırma yöntemleri ve/veya örnekleme yöntemleri uygulanabilir.



Şekil 1.7: Yığılmış Genelleme Yöntemi

Adım2: Temel sınıflandırıcıların çıktısı üzerinde yeni bir veri seti oluşturun. Burada, birinci seviye sınıflandırıcıların çıktı tahmini etiketleri yeni bir özellik olarak kabul edilir ve orijinal sınıf etiketleri yeni veri setinde etiketler olarak tutulur. D' deki her örneğin x_i, y_i olduğu varsayılırsa, yeni veri kümesinde $x_i^{'} = \{h_1(x_i), h_2(x_i), ..., h_T(x_i)\}$ karşılık gelen bir örnek $\left\{x_i^{'}, y_i\right\}$ oluşturulur.

Adım3: Yeni oluşturulan veri kümesine dayalı olarak ikinci düzey bir sınıflandırıcı öğrenin. İkinci seviye sınıflandırıcıyı öğrenmek için herhangi bir öğrenme yöntemi uygulanabilir.

İlk olarak ikinci seviye sınıflandırıcı oluşturulur ve birinci seviye sınıflandırıcıları birleştirmek için kullanılabilir. Yeni bir x örneği için, yöntemin tahmini sınıf etiketi $h^{'} = \{h_1(x), h_2(x), ..., h_T(x)\}$'dir, burada $h_1, h_2, \ldots, h_T$ birinci düzey sınıflandırıcı ve $h^{'}$ ise ikinci düzey sınıflandırıcıdır.

Tablo 1.5: Yığılmış genelleme algoritması

Algorithm: Stacking
Girdi: Eğitim verisi $D = \{x_i, y_i\}_{i=1}^{m} (x_i \in R^n, y_i \in Y)$
Çıktı: Topluluk sınıflandırıcısı H
1: Adım 1: Birinci seviye sınıflandırıcıları öğren
2: $t \leftarrow 1$ den Tye kadar **tekrar yap**
3: D 'ye dayalı bir temel sınıflandırıcı h_t' yi öğren.
4: **tekrar sonu**
5: Adım 2: D verisinden yeni veri kümeleri oluşturun
6: $t \leftarrow 1$ den $m$ye kadar **tekrar yap**
7: Yeni bir $\left\{x_i^{'}, y_i\right\}$ veri seti oluştur, burada $x_i^{'} = \{h_1(x_i), h_2(x_i), ..., h_T(x_i)\}$
8: **tekrar sonu**
9: Adım 3: İkinci düzey bir sınıflandırıcı öğrenin
10: Yeni oluşturulmuş veri setinden yeni bir sınıfladırıcı $h^{'}$ oluşturun
11: Sonucu döndür: $H(x) = h^{'} = \{h_1(x), h_2(x), ..., h_T(x)\}$

Çalışma prensibinden de anlaşılacağı gibi yığılmış genelleme yöntemi genel bir çerçevedir. Birinci veya ikinci seviye sınıflandırıcılar oluşturmak için farklı öğrenme yaklaşımları eklenebilir veya hatta yaklaşımlar bir araya getirilebilir. Torbalama ve Yükseltme ile karşılaştırıldığında, Yığılmış

genelleme yöntemi, oylama yerine temel sınıflandırıcıların nasıl birleştirileceğini öğrenmektedir.

1.3.6 Topluluk Yöntemlerinin Faydaları

Topluluk yöntemleri algoritmaları birleştirerek sınıflandırma ya da tahmin doğruluklarını arttırmayı sağlamanın yanında aşağıda belirtilen beş önemli konuda da fayda ve iyileştirmeler sağlamaktadır (Polikar, 2012).

1. Artımlı Öğrenme: Mevcut verileri ve önceki hipotezleri kullanarak bir hipotezi sıralı olarak güncellemeye çalışan bu kavram algoritmanın mevcut bilgiyi unutmama ve yeni verilere yol gösterme yeteneğini ifade eder. Topluluk tabanlı sistemler, aynı zamanda kararlılık-plastisite ikilemine dengeli bir çözüm sağlayan artımlı öğrenme için sezgisel bir yaklaşım sağlar.
2. Veri birleştirme: Veri analizi uygulamalarındaki yaygın bir sorun, genellikle heterojen veriler sağlayan farklı veri kaynaklarından gelen bilgileri bir araya getirmektir. Topluluk sistemleri bu tür problemler için çözüm sağlar. Bireysel sınıflandırıcılar her bir veri kaynağı üzerinde eğitilebilir ve daha sonra uygun bir birleştirici aracılığıyla birleştirilebilir.
3. Özellik seçimi ve eksik verilerle sınıflandırma: Gerçek dünya uygulamalarında eksik, gürültülü ve bozuk veriler oldukça yaygındır. Topluluk tabanlı sistemler de bireysel sınıflandırıcılar, tüm özellik setinin farklı alt kümeleri ile eğitilir. Bu nedenle bu yaklaşım aynı zamanda özellik seçimi ve çeşitlilik geliştirme için de kullanılabilir.
4. Kavram kayması: Topluluk tabanlı algoritmalar, kavram kayması problemlerine alternatif bir yaklaşım sağlar. Bu algoritmalar genellikle üç kategoriden birine aittir: (1) Sabit bir grubun kombinasyon kurallarını veya oy ağırlıklarını güncellemek; (2) Çevrimiçi bir öğrenici kullanarak mevcut topluluk üyelerinin parametrelerini güncellemek (3) Gelen her veri kümesiyle bir topluluk oluşturmak için yeni üyeler eklemek.
5. Güven tahmini: Bir topluluktaki sınıflandırıcıların önemli bir çoğunluğu kararları üzerinde hemfikirse, topluluğun kararına tek sınıflandırıcı kararına göre daha fazla güvenilir olduğu düşünülmektedir. Ancak bu güvenilirliğin anlamlı olması için sınıflandırıcı kararlarının bağımsız olması gerekmektedir.

1.4 Uygulama

Çalışmada kullanılan "Car Evaluation Database" veri seti UCI Machine Learning Repository sitesinden alınmıştır (`https://archive.ics.uci.edu/ml/datasets/car+evaluation`). Veri kümesi 1997 yılında Marko Bohanec tarafından elde edilmiştir. Araba değerlendirme ile ilgili toplamda 1728 tane örnek içeren veri setinde yedi adet nitelik bulunmaktadır. İlk altı nitelik girdi değişkenlerini, yedinci nitelik ise sınıf etiketlerini yani çıktı değişkenini ifade etmektedir. Girdi değişkenleri; satın alma maliyeti, bakım maliyeti, kapı sayısı, insan kapasitesi, bagaj kapasitesi ve güvenliktir. Bu girdilere karşılık çıktı değişkeni ise çok iyi, iyi, kabul edilebilir ve kabul edilemez olarak dört kategorilidir. Örneklerden 1210 tanesi kabul edilmez, 384 tanesi kabul edilebilir, 69 tanesi iyi ve 65 tanesi ise çok iyidir. Veri setindeki değişkenler ve değer aralıkları Tablo 1.7. de gösterilmektedir. Değişkenlere bağlı olarak veri setinden 10 tane örneği içeren değerler ise Tablo 1.8. de sunulmaktadır.

Veri setinde girdilere karşılık hedef değerleri bulunmaktadır. Bundan dolayı problem sınıflandırma yaklaşımı ile ele alınacaktır. Kurulan modeller sayesinde, bir kişi araba alacağı zaman bu değişkenlere bağlı olarak ne karar vereceği tahmin edilecektir. Uygulamadaki amaç, bu tahmini en iyi şekilde gerçekleştirebilmektir. Bu amaca ulaşmak için de topluluk algoritmaları kullanılacaktır. İlk önce sınıflandırma algoritmalarından en bilinenlerinden olan karar ağacı, naive bayes ve k-en yakın komşu (k-nn) algoritmaları bireysel olarak modellenecektir (Warner 1998). Ardından bu

Tablo 1.6: Veri seti değişken tanımları

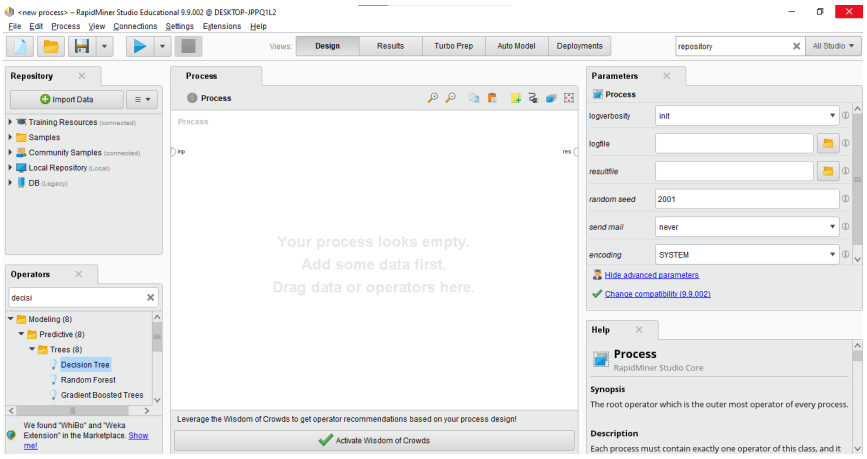
	Değişkenler	Açıklama	Değer
1	sm	Satın alma maliyeti	Çok yüksek, yüksek, orta, düşük
2	bm	Bakım maliyeti	Çok yüksek, yüksek, orta, düşük
3	ks	Kapı sayısı	2, 3, 4, 5 ve daha çok
4	ık	İnsan kapasitesi	2, 4, daha çok
5	bk	Bagaj kapasitesi	Küçük, orta, büyük
6	güv	Güvenlik	Düşük, orta, yüksek

Tablo 1.7: Araba değerlendirme veri seti örnek

	sm	bm	ks	ık	bk	güv	sınıf
1	çyüksek	orta	5 ve daha çok	4	küçük	yüksek	Kabul edilebilir
2	yüksek	çyüksek	2	2	küçük	düşük	Kabul edilemez
3	yüksek	çyüksek	2	2	küçük	orta	Kabul edilemez
4	çyüksek	orta	5 ve daha çok	4	küçük	yüksek	Kabul edilebilir
5	orta	düşük	2	2	küçük	düşük	Kabul edilemez
6	düşük	orta	5 ve daha çok	4	büyük	orta	İyi
7	düşük	orta	5 ve daha çok	4	orta	yüksek	Çok iyi
8	orta	orta	3	2	küçük	düşük	Kabul edilemez
9	yüksek	orta	4	4	büyük	yüksek	Kabul edilebilir
10	yüksek	düşük	2	2	küçük	düşük	Kabul edilemez

yöntemler torbalama ve yükseltme algoritmaları ile topluluk olarak kullanılarak sınıflandırma modelleri kurulacaktır. Sonuç olarak modellerin sınıflandırma performansları karşılaştırılarak topluluk yöntemlerinin sınıflandırma doğruluğu üzerindeki etkileri incelenecektir.

Uygulamada Rapidminer Studio 9.9 veri madenciliği ve makine öğrenmesi programı kullanılmıştır. İşlem operatörlerini bir bilgi akış sistemi ile birbirine bağlayarak çalışan program birçok veri formatıyla çalışmayı desteklemektedir. Programın ana uygulama ekranı aşağıda Şekil 1.8' de görüldüğü gibidir.

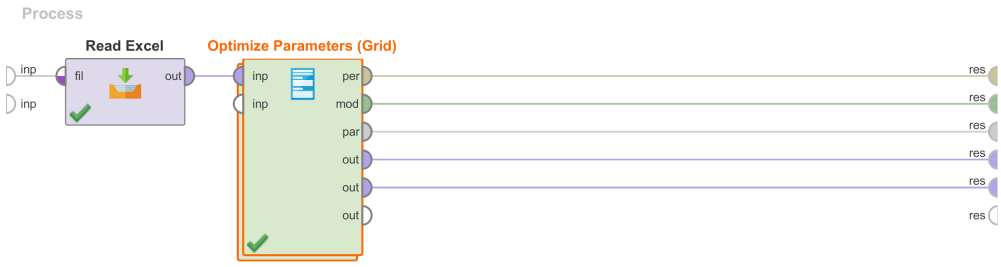


Şekil 1.8: Rapidminer Ana Ekran

Ekranda en üstte araçların bulunduğu alan mevcuttur. Alt kısım ise beş bölümden oluşmaktadır. "Repository" hazır örnek veriler ya da kullanıcı tarafından yüklenen verilerin ve depoların bulunduğu kısımdır. "Operators" penceresi veri madenciliği ile ilgili kullanılabilecek algoritma ve tekniklerin bulunduğu işlemler bölümüdür. Bu bölümde veri yükleme, veri tanımlama, veri ön işleme, modelleme, değerlendirme gibi işlemler gerçekleştirilebilir. "Parameters" kısmı ise "Operators" altındaki işlemlerin çalışma performanslarını belirlemeye yarayan parametrelerin ayarlanmasını sağlayan penceredir. "Help" programla ilgili destek alınabilecek yardım kısmıdır. Son olarak ortadaki beyaz ekran ise modellerin kurulduğu boş alandır.

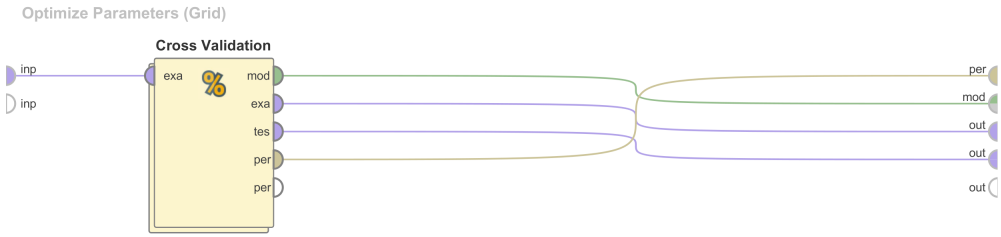
Uygulama yapmak için ilk olarak veri setinin yüklenme işlemi gerçekleştirilmektedir. Veri yükleme işlemi için; "operators $\rightarrow$ data access $\rightarrow$ files $\rightarrow$ real excel" adımları ile işlem kutusu beyaz alana sürüklenerek eklenebilir. Ardından ilgili yerden veri çekilerek parameters bölümünden veri seti programa tanıtılır. Uygulamadaki veri setinde eksik ve aykırı değer bulunmamaktadır. Tüm değişkenler, amaçla ilişkili ve anlaşılır olduğu için veri dönüştürme işlemine de ihtiyaç duyulmamaktadır. O nedenle veri önişleme adımı atlanarak modeller kurulmuştur.

Çalışmada modellerdeki algoritmaların parametrelerinin seçimi için Şekil 9'da görülen optimizasyon (optimization-grid) tekniği kullanılmıştır. Burada veri tanıtma ve optimizasyon ile bunlar arasındaki bağlantılar ve çıktıları görülmektedir. Bu bağlantılar sayesinde işlem kutucukları arasında bilgi akışı sağlanmaktadır. Veri tanıtma işleminde, kullanılacak değişkenler, değişkenlerin türleri ve hedef değerinin hangisi olduğu gibi bilgiler modelleme sürecine tanıtılmış olmaktadır. Veriler ilk haliyle sisteme girilip modelin ihtiyaçlarına göre tanıtılarak optimizasyon işlem kutusuna iletilmektedir. Algoritma ve işlemlerin parametre seçimleri optimizasyon tekniği kullanılarak gerçekleştirildiği için sezgisel yöntemler kullanıldığında göz ardı edilen seçim kombinasyonları da değerlendirilmiş olmaktadır. Ancak buna karşın çözüm sürecinin uzayacağı gerçeği unutulmamalıdır.



Şekil 1.9: Optimizasyon işlemi

Bunun yanında uygulamada yapılan testin hatasını en doğru şekilde tespit edebilmek için model kurma aşamasında k-katlamalı çapraz doğrulama (k-fold cross validaiton) tekniği kullanılmıştır. Çapraz doğrulama işlem kutusu Şekil 1.10'da sunulduğu gibi optimizasyon tekniği işleminin içerisine tanımlanmıştır. Optimizasyon işlem kutusu içerisine işlemlerin tanımlanması gereken bir yapıdadır. Çapraz doğrulama işlem kutusunda, model, test ya da performans sonuçları çıktı olarak bağlanmıştır. Kurulan modellere göre veri işlenerek sonuçlar elde edilmektedir. Çapraz doğrulama işlem kutusu da içine model ve işlemler tanımlanması gereken bir yapıdadır. Şekil 1.11'de sunulan model, çapraz doğrulama işlem kutusu çift tıklanarak oluşturulmuştur.



Şekil 1.10: Veri tanımlama ve çapraz doğrulama bağlantıları.

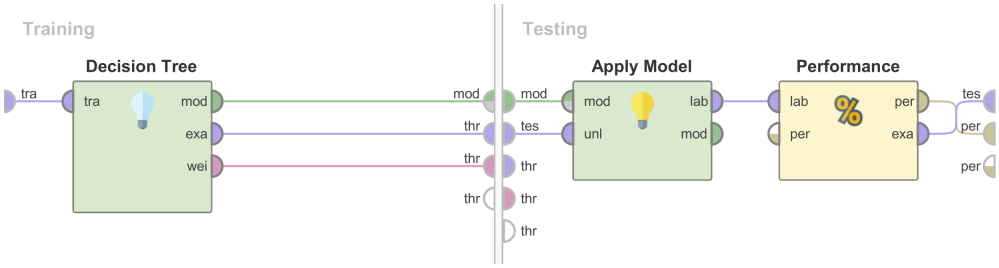
Çapraz doğrulama yönteminde veri seti belirli sayıda parçaya bölünür. Bir parça test veri seti olarak ayrılır diğer parçalar eğitim için ayrılır ve model çalıştırılır. Ardından ikinci parça test olarak ayrılır diğer parçalar eğitim için kullanılır. Bu süreç, tüm veri seti hem test hem de eğitim aşamalarına katılana kadar devam eder. Performans başarısı her aşamada elde edilir ve nihai başarı, her aşamanın ortalamasından oluşur. Veri setinin kaç parçaya bölüneceğini ifade eden k-kat sayısı genellikle 10 olarak alınır (Refaeilzadeh, 2009). Benzer şekilde çalışmada kat sayı 10 olarak alınmıştır. Çapraz doğrulamada, veri setini örneklemek için kullanılan kriterler vardır. Bunlar; doğrusal (linear sampling), rastgele (shuffled sampling), tabakalı (Stratified sampling) ve otomatik (automatic samling) örneklemedir.

İlk olarak makine öğrenmesi algoritmalarından sık kullanılan (Warner, 1998) karar ağacı, naive bayes ve k-en yakın komşu algoritmaları bireysel olarak ele alınmıştır. Ardından algoritmalar homojen ve heterojen olarak paralel ve seri bağlama şeklinde birleştirilmiştir. Tüm modellerin değerlendirme sonuçları karşılaştırılarak topluluk algoritmalarının sağladığı iyileştirmeler ve başarı oranları incelenmiştir.

1.4.1 Yöntem

Bireysel Algoritmalarla Modeller

Karar Ağacı: Yaygın kullanılan makine öğrenmesi yöntemlerinden olan karar ağacı algoritması düğümlerden oluşan bir yapıya sahiptir. Kök, ara ve yaprak düğümlerden oluşan ağaç yapısındaki bu algoritma kök düğümün belirlenmesi için belirli kriterlere göre veri setini alt parçalara ayırarak çalışmaktadır. Eğer-ise kural yapısı ile temsil edilebildiği için de kullanımı ve anlaşılması kolay ve pratiktir (Mitchell 2014; Agrawal 1993). Çapraz doğrulama içerisine tanımlanan karar ağacı modeli Şekil 1.11'de sunulmaktadır. Model eğitim ve test olarak iki kısımdan oluşmaktadır. Eğitim kısmında algoritma, test kısmında ise modelin uygulanması ve performans ölçümü gerçekleştirilmektedir. Ayrıca diğer bireysel algoritmalara ait modellerde aynı yöntemlerle kurulmuştur.



Şekil 1.11: Karar ağacı model yapısı

Karar ağacına ait kullanılan parametreler Tablo 8'de gösterilmiştir.

Tablo 1.8: Karar ağacı parametreleri

Parametreler	Değer	Doğruluk%
Çapraz Doğrulama	Tabakalı örnekleme	93.46
Karar Ağacı	Kazanç oranı	
Maximum Derinlik	10	
Güvenilirlik	0.1	
Minimum Kazanç	0.01	

Naive Bayes: Öğrenme algoritmaları arasındaki pratik algoritmalardan biri olan naive bayes algoritması bayesci yaklaşımı temel almaktadır. Örneklem değeri ve öncü bilgiyi bir araya getirmeyi hedeflemektedir (Alpaydın, 2012). Bu algoritmada Laplace düzeltme (correction) kriteri kullanılmıştır. Bunun dışında herhangi bir parametre bulunmamaktadır (Anwar, 2014). Naive bayes algoritmasında kullanılan parametreler Tablo 1.9'da gösterilmektedir.

Tablo 1.9: Naive Bayes algoritma parametreleri

Parametreler	Değer	Doğruluk%
Çapraz Doğrulama	Tabakalı örnekleme	86.17
Laplace Düzeltme	Hayır	

k-En Yakın Komşu: Yeni örneği mevcut örneklerin benzer özelliklere göre uygun sınıfa atamayı amaçlayan k-en yakın komşu yöntemi oldukça güçlü bir algoritmadır (Lantz, 2019). Özellik

benzerliği için en yakında bakılacak komşu sayısını ifade eden “k” değerini kullanıcı belirlemektedir. Bu sayı birden büyük ve tek sayı olmalıdır ayrıca çok büyük sayılar etkinliği azaltacağı için tercih edilmemelidir (Nilsson, 1996). En yakın komşunun yerini belirlemek için ise örneklerin benzerliğini hesaplayan Öklid, Minskowski veya Manhattan uzaklığı gibi ölçüm değerleri bulunmaktadır (Bilgin, 2018).

k-en yakın komşu algoritmasına ait kullanılan parametreler Tablo10’da gösterilmiştir.

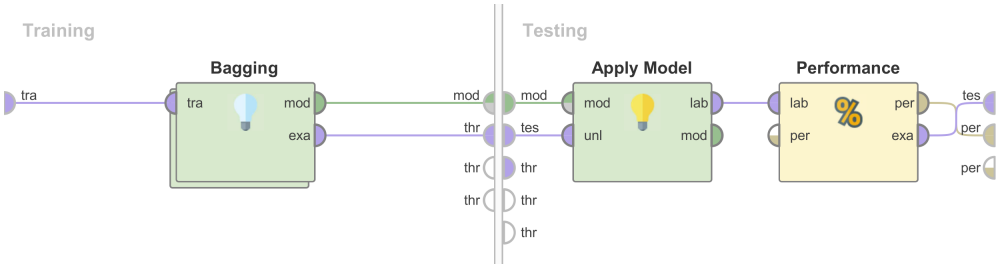
Tablo 1.10: K-en yakın komşu parametreleri

Parametreler	Değer	Doğruluk%
Çapraz Doğrulama	Rastgele örnekleme	89.07
k sayısı	11	
Measured type	Mixed measured	
Mixed measured	Mixed ecludian distance	

Topluluk algoritmaları ile modeller

Bireysel algoritmalarda olduğu gibi topluluk algoritmalarının parametreleri de optimizasyon ile belirlenmiştir. İki topluluk yönteminde de belirlenen parametre, yineleme (iterasyon) sayısı yani kaç tane algoritmanın bağlanması gerektiğidir.

Torbalama yöntemi: Torbalama algoritmasının programda uygulanması Şekil 1.12’de görüldüğü gibidir. Modeller çapraz doğrulama içerisine kurulmuştur. Topluluk algoritmaları, içerisine bireysel algoritma tanımlanması gereken bir yapıdadır. Şekil 1.13’de topluluk algoritmasının iç yapısı görülmektedir. Yükseltme algoritma yapısı da aynı şekildedir. Sadece torbalama işlem kutusu yerine yükseltme işlemi konulacaktır. Ayrıca hangi bireysel algoritma ile model kurulacaksa topluluk algoritmasının iç yapısına o algoritmaya ait işlem kutusu eklenmeli ve uygun bağlantılar gerçekleştirilmelidir.



Şekil 1.12: Torbalama topluluk algoritması genel model yapısı

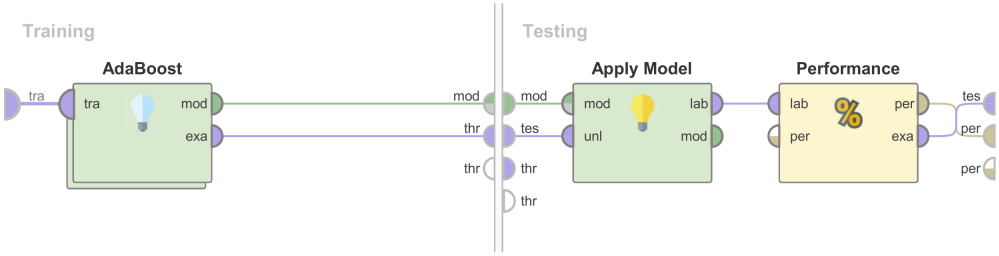
Şekil 1.13: Torbalama topluluk modelinin iç yapısı

Torbalama yönteminin karar ağacı, naive bayes ve k-nn algoritmaları ile kurulan modellere ait parametereler ve sonuçta elde edilen doğruluk performansları Tablo 1.11'de sunulmaktadır.

Tablo 1.11: Torbalama Algoritması Parametre Değerleri ve Doğruluklar

Algoritma	Kriterler	Değer	Performans
Torbalama-Karar Ağacı	Çapraz doğrulama örnekleme türü:	Otomatik	95.14
	Torbalama yineleme sayısı:	31	
	Karar ağacı kriteri:	Bilgi kazanımı	
Torbalama-K-nn	Çapraz doğrulama örnekleme türü:	Rastgele	89.07
	Torbalama yineleme sayısı:	1	
	k sayısı:	11	
Torbalama-Naive bayes	Çapraz doğrulama örnekleme türü:	Rastgele	86.64
	Torbalama yineleme sayısı:	41	
	Laplace düzeltme:	YANLIŞ	

Yükseltme Yöntemi: Yükseltme yöntemi ile kurulan model yapısı Şekil 1.14'de sunulmaktadır. Burada kullanılacak parametre torbalama algoritmasına benzer olarak kaç tane algoritmanın birleştirileceğini belirten kriterdir. Ayrıca çapraz doğrulama ve kullanılan bireysel algoritmalara ait kriterlerin parametreleri mevcuttur.



Şekil 1.14: Yükseltme topluluk algoritması genel model yapısı

Yükseltme algortiması ile kurulan modellerdeki parametreler ve elde edilen doğruluk performansları Tablo 1.12'de sunulmaktadır.

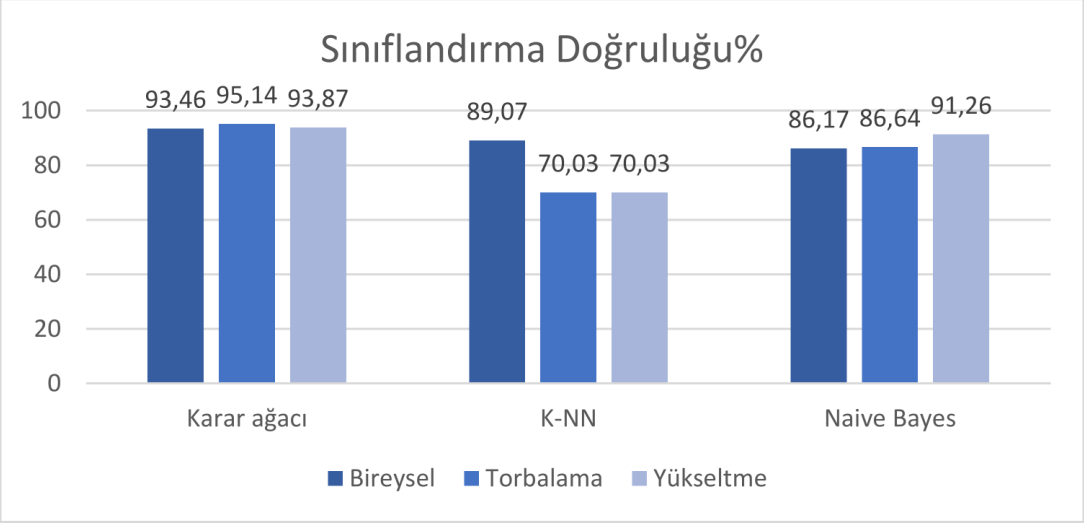
1.5 Sonuçlar ve Tartışma

Topluluk algoritmalarının faydalarını incelemek için gerçekleştirilen araba değerlendirme konusundaki sınıflandırma uygulamasında ilk olarak bireysel algoritmaların başarı performansları incelenmiştir. Bireysel algoritmalar arasından karar ağacı, k- en yakın komşu ve naive bayes kullanılmıştır. Ardından aynı bireysel algoritmalar paralel ve sıralı bağlanma prensibiyle yani torbalama ve yükseltme makine öğrenmesi algoritmaları ile birleştirilmiştir. Tüm algoritma ve yöntemlerle kurulan modellerin performansları incelenmiştir. Sonuç olarak performansların çoğunlukla iyileştiği görülmüştür. Farklı algoritmaları kullanılarak oluşturulan modellerin başarı oranları Şekil 1.15'de gösterilmektedir.

Optimum parametre seçimleri yapıldıktan sonra elde edilen performans oranlarına ait sonuçlar incelendiğinde, bireysel algoritmaların arasında sınıflandırma başarı performansı en yüksek olan

Tablo 1.12: Yükseltme Algoritması Parametre Değerleri ve Doğruluklar

Algortima	Kriterler	Değer	Doğruluk
Yükseltme-Karar Ağacı	Çapraz doğrulama örnekleme türü: Yükseltme yineleme sayısı: Decision Tree	Rastgele 11 Gini-index	93.87
Yükseltme-K-nn	Çapraz doğrulama örnekleme türü: Yükseltme yineleme sayısı: k-nn	Rastgele 1 11	89.07
Yükseltme-Naive Bayes	Çapraz doğrulama örnekleme türü: Yükseltme yineleme sayısı: Laplace düzeltme:	Tabakalı 41 DOĞRU	91.26



Şekil 1.15: Sınıflandırma modellerin tahmin doğrulukları

algoritma 93,46% oranı ile karar ağacı olarak elde edilmiştir. Ardından 89,07% ile k-NN ve en son sırada ise 86,17% başarı oranı ile naive bayes olarak görülmektedir.

Bireysel değerlendirmenin ardından farklı birleştirme yaklaşımları ile topluluk öğrenme algoritmalarının performansları değerlendirilmiştir.

Değerler incelendiğinde bireysel algoritmaların kullanılmasına benzer olarak torbalama topluluk algoritmasında da yine karar ağacının en yüksek başarı oranına sahip olduğu ayrıca bireysel karar ağacı kullanımına göre daha başarılı olduğu görülmektedir. En başarılı algoritmanın 95,14% başarı oranı ile torbalama-karar ağacı olduğu ardından torbalama-k-nn algoritmasının geldiği görülmektedir. Ancak torbalama k-nn algoritmasının başarı oranının 87,07% olarak bireysel k-nn algoritması ile aynı değerde olduğu görülmektedir. Yani bu veri seti için k-nn algoritmasını paralel olarak bağlamak başarıyı arttırmamıştır. Bu durum topluluk algoritmalarının genellikle iyi yönde ilerlediğini ancak bu durumun kesin doğru olmadığını göstermektedir. Başarı oranı en düşük olan algoritma bireysel algoritmalarda olduğu gibi naive bayes algoritmasıdır. Bu algoritma torbalama yöntemi arasında en düşük başarı oranına sahip olsa da 86,64% ile bireysel olarak kullanılmasına göre az da olsa başarı elde edilmiştir.

Yükseltme yöntemi incelendiğinde ise, başarı sıralamasının bireysel ve torbalama yöntemine göre aynı olarak bulunmaktadır. Karar ağacının başarısının 93,87% başarı ile bireysele göre daha yüksek ancak torbalama yöntemine göre daha az başarılı olduğu, K-nn algoritmasının başarısının bireysel, torbalama ve yükseltme yöntemlerinin hepsinde 89,07% oranı ile aynı seviyede kaldığı, Naive bayes algoritmasının ise 91,26% bireysel ve torbalama yöntemine göre artmış olduğu görülmektedir.

Uygulamada kullanılan veri setinde topluluk algoritmalarının sonuçlara başarılı yönde etki ettiği söylenebilmektedir.

1.6 Kaynaklar

Agrawal, R., Imieliński, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. In Proceedings of the 1993 ACM SIGMOD international conference on Management of data (pp. 207-216).

Alpaydın, E., Yapay Öğrenme, (2017). 3. baskı. Boğaziçi Üniversitesi.

Anwar, H., Qamar, U., & Muzaffar Qureshi, A. W. (2014). Global optimization ensemble model for classification methods. The Scientific World Journal, 2014.

Bilgin, M., (2018). Veri Biliminde Makine Öğrenmesi Makine Öğrenmesi Teorisi ve Algoritmaları, Papatya Bilim, 2. Baskı.

Breiman, L. (1996). Stacked regressions. Machine learning, 24(1).

Dietterich, T. G. (2000, June). Ensemble methods in machine learning. In International workshop on multiple classifier systems(pp. 1-15). Springer, Berlin, Heidelberg.

Gowda, S., Kumar, H., Imran, M.,(2018). Ensemble Based Learning with Stacking, Boosting and Bagging for Unimodal Biometric Identification System. 12th International Conference on Recent Trends in Engineering and Technology

Hossin, M., & Sulaiman, M. N. (2015). A review on evaluation metrics for data classification evaluations. International journal of data mining & knowledge management process, 5(2), 1. ISO 690

Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. Journal of artificial intelligence research, 4, 237-285.

Krawczyk, B., Minku, L. L., Gama, J., Stefanowski, J., & Woźniak, M. (2017). Ensemble learning for data stream analysis: A survey. Information Fusion, 37, 132-156.

Langley, P. (1996). Elements of machine learning. Morgan Kaufmann.

Lantz, B. (2019). Machine learning with R: expert techniques for predictive modeling. Packt publishing ltd.

Maclin, R., & Opitz, D. (1997). An empirical evaluation of bagging and boosting. AAAI/IAAI, 1997, 546-551.

Mahesh, B. (2020). Machine Learning Algorithms-A Review. International Journal of Science and Research (IJSR).[Internet], 9, 381-386.

Mitchell, T. M. (2006). The discipline of machine learning (Vol. 9). Pittsburgh: Carnegie Mellon University, School of Computer Science, Machine Learning Department.

Mitchell, T. M., (2014). Machine Learning, M. Mcgraw-hill Science. Engineering/Math, 1, 27. Mohammed, M., Khan, M. B., & Bashier, E. B. M. (2016). Machine learning: algorithms and applications. Crc Press.

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2018). Foundations of machine learning. MIT press.

Nilsson, N. J. (1996). Introduction to machine learning. An early draft of a proposed textbook.

Onan, A. (2018). Particle Swarm Optimization Based Stacking Method with an Application to Text Classification. Academic Platform Journal of Engineering and Science, 6(2), 134-141.

Oza, N. C., & Tumer, K. (2008). Classifier ensembles: Select real-world applications. Information fusion, 9(1), 4-20.

Polikar, R. (2012). Ensemble learning. In Ensemble machine learning (pp. 1-34). Springer, Boston, MA.

Quinlan, J. R. (1996). Bagging, boosting, and C4. 5. In Aaai/iaai, Vol. 1 (pp. 725-730).

Raschka, S. (2018). Model evaluation, model selection, and algorithm selection in machine learning. arXiv preprint arXiv:1811.12808.

Re, M., & Valentini, G. (2012). Ensemble methods. Advances in machine learning and data mining for astronomy, 563-593. Refaeilzadeh, P., Tang, L., Liu, H., (2009). C Cross-Validation, Springer, Boston, 1-3, 2009.

Rokach, L. (2010). Ensemble-based classifiers. Artificial intelligence review, 33(1), 1-39.

Sewell, M. (2008). Ensemble learning. RN, 11(02), 1-34.

Silva-Palacios, D., Ferri, C., & Ramírez-Quintana, M. J. (2017). Improving performance of multiclass classification by inducing class hierarchies. Procedia Computer Science, 108, 1692-1701.

Tang, J., Alelyani, S., & Liu, H. (2014). Data classification: algorithms and applications. Data Mining and Knowledge Discovery Series, 37-64.

Warner, R. M. (1998). Spectral analysis of time-series data. Guilford Press.

Wolpert, D. H. (1992). Stacked generalization. Neural networks, 5(2).

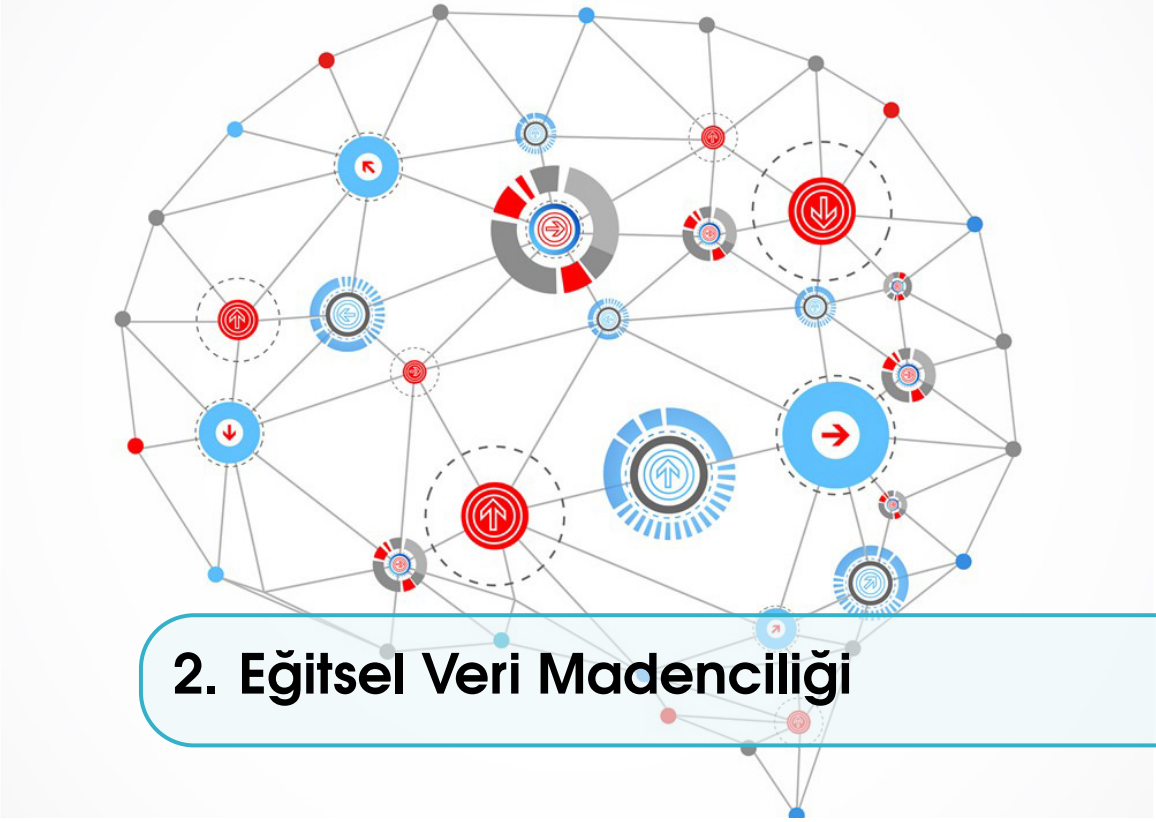
Zhang, C., & Ma, Y. (Eds.). (2012). Ensemble machine learning: methods and applications. Springer Science & Business Media.

Zheng, A., (2015) Evaluating Machine Learning Models, O'Reilly Medis, Inc, 2015.

Zhou, Z. H. (2012). Ensemble methods: foundations and algorithms. Chapman and Hall/CRC.

Zhu, X., & Goldberg, A. B. (2009). Introduction to semi-supervised learning. Synthesis lectures on artificial intelligence and machine learning, 3(1), 1-130.

`https://archive.ics.uci.edu/ml/datasets/car+evaluation`



2. Eğitsel Veri Madenciliği

Eğitsel Veri Madenciliği ve Kural Çıkarımı Örneği

Deniz DEMİRCİOĞLU DİREN*, Mehmet Barış HORZUM†

***Sakarya Üniversitesi, Uzaktan Eğitim Araştırma ve Uygulama Merkezi †Sakarya Üniversitesi, Eğitim Fakültesi, Bilgisayar ve Öğretim Teknolojileri Eğitimi Bölümü**

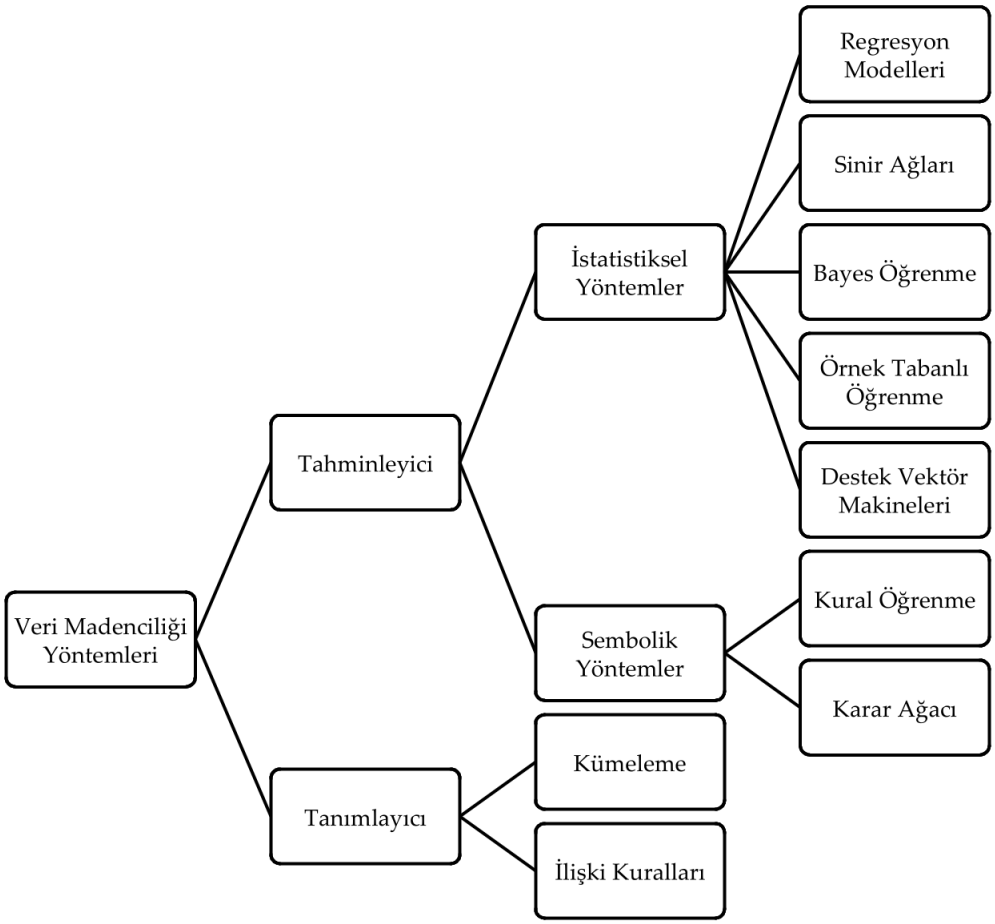
2.1 Giriş

Geniş bir uygulama alanı bulunan veri madenciliği, bir problem ile ilgili toplanan verilerden yararlı bilgiler elde etme, temizleme, işleme, analiz etme ve sonuç üretme sürecidir (Aggarwal,2015). Veri madenciliği süreci temel olarak aşağıdaki adımları takip ederek ilerlemektedir. Bu adımlar tekrarlanabilir ve gerektiğinde geriye doğru hareket edebilir (Chung ve Gray, 1999):

1. Son kullanıcının ihtiyaç ve hedeflerine göre bir problem belirlenir.
2. Probleme göre veri seti oluşturulur.
3. Veriler temizlenir ve ön işlemeye tabi tutulur.
4. Değişken sayısı azaltılır.
5. Veri madenciliği tekniği seçilir.
6. Veri madenciliği algoritması seçilir.
7. Örüntü modelleri incelenir (Bu adım gerçek veri madenciliğidir).
8. Çıkarılan model yorumlanır. İhtiyaç olursa önceki adımlara dönülebilir.
9. Sonuçlar birleştirilir ve raporlanır.

Hangi tür veri analizi yapılırsa yapılsın ilk iki adım ortaktır. Problemin hedefleri belirlenerek uygun veri seti oluşturulduktan sonra üçüncü adıma geçirilir. Bu adımda veri, temizleme ve önişleme sürecinden geçirilerek analiz için bir adım daha uygun hale getirilir. Ön işleme uygulaması eksik verilerin ele alınması, gürültülü verilerin temizlenmesi ya da normalizasyon gibi adımları içermektedir. Ardından dördüncü adımda boyut indirgeme veya dönüştürme işlemleri uygulanarak gereksiz değişkenler azaltılır. Analiz için tamamen hazır hale gelen veri seti için beşinci adımı takip ederek problemin hedefine uygun olan veri madenciliği tekniği seçilir. Seçilecek veri madenciliği tekniği

Şekil 2.1'de görüldüğü gibi tahminleyici ya da tanımlayıcı olabilir.



Şekil 2.1: Veri Madenciliği Yöntemleri (Garica ve diğerleri,2015))

Altı ve yedinci adımlar ise verilerde bilgi keşif sürecinin kalbi gibidir. Asıl veri madenciliği bu sürece göre burada gerçekleştirilir. Bu süreçte tüm aşamalar birbirine bağlıdır ve süreç sonunda veri bilgiye dönüşmektedir (Garcia ve diğerleri, 2015).

Sekizinci adımda modellerin doğruluk uygunlukları bir analist tarafından değerlendirilir. Eğer uygun bulmazsa önceki adımlar tekrarlanır ve yeni duruma göre model tekrar çalıştırılarak yeniden değerlendirilir. Eğer uygun bulunursa bir sonraki adıma geçirilir ve sonuçlar raporlanarak süreç tamamlanır. Bahsedilen adımlardan anlaşılacağı üzere veri madenciliği ustaca yapılan bir analiz ve neredeyse hatasız bir metodoloji seçimi gerektirmektedir. Bu durum da birçok farklı disiplininin bir araya gelmesini gerektirmektedir. Dolayısıyla bu çeşitlilik günümüzde veri madenciliği teknik ve yöntemlerinin çok farklı alanlarda kullanılarak fayda elde edilmesini sağlamaktadır. Bu çalışma alanlarından biri de eğitim alanıdır.

2.2 Eğitsel Veri Madenciliği

2005 yılında Romero & Ventura tarafından kavram olarak kullanılmış olan Eğitsel Veri Madenciliği, eğitim ortamlarından gelen karmaşık ve büyük ölçekli verileri keşfetmek için yöntemler geliştiren ve bu yöntemleri öğrencileri ve içinde öğrendikleri ortamları daha iyi anlamak için kullanan disiplindir. Kendi başına önemli bir çalışma alanı olan eğitsel veri madenciliği, Baker ve Yacef (2009) tarafından beş sınıfa ayrılmaktadır.

1. Tahmin (Sınıflandırma ve Regresyon)
2. Kümeleme
3. İlişki Madenciliği (Birliktelik Kuralı, Korelasyon ve Sıralı Örüntü Madenciliği)
4. İnsan karar alma süreci için verinin damıtılması
5. Modellerle keşif

Eğitsel veri madenciliği en çok uzaktan eğitim sistemleri, öğrenme yönetim sistemleri, web tabanlı uyarlanabilir sistemlerde ve zeki öğrenme sistemlerinde kullanılabilmektedir. Bunun en temel sebebi bu tür eğitim sistemlerinin teknoloji tabanlı olması ve teknoloji sayesinde veri depolama olanaklarının daha fazla olmasıdır. Kullanıcıların sistem hareketlerinden elde edilen verilerinden, kümleme, sınıflandırma, ilişki kuralları ya da görselleştirme gibi yöntemlerle bilgi çıkarımı yapılmaktadır (Romero, Ventura, Pechenizkiy & Baker 2010). Amaçlanan, bu sistem içerisindeki kullanıcıların verilerinin, işlemlerinin ve hareketlerinin depolanması, izlenmesi ve yorumlanmasıdır.

2.3 Eğitsel Veri Madenciliği ile İlişkili Alanlar

Disiplinler arası bir çalışma alanı olan eğitsel veri madenciliği eğitim, istatistik ve bilgisayar bilimi alanlarının birleşmesiyle oluşmaktadır Ayrıca bu üç alanın kesişimi, bilgisayar tabanlı eğitim, veri madenciliği, makine öğrenme ve öğrenme analitiği gibi alanları da oluşturmaktadır (Romero & Ventura, 2013).

Şekil 2.2: Eğitsel veri madenciliği ile ilişkili alanlar (Romero ve Ventura,2013))

Şekil 2.2 incelendiğinde, bilgisayar bilimleri istatistik bilimi ile birleşerek veri madenciliği ve makine öğrenme alanını, eğitim bilimi ile birleşerek ise bilgisayar tabanlı eğitim alanını oluşturmaktadır. Eğitim ve istatistik bilimi birleştiğinde de öğrenme analitikleri alanı oluşmaktadır. Eğitsel

veri madenciliği bu alt alanlarla da yakından ilişkilidir. Farklı disiplinleri bir arada barındırarak etkileşimle çalışan eğitsel veri madenciliği alanı bu sayede yenilikçidir.

2.4 Eğitsel Veri Madenciliğinin Bileşenleri

Eğitim alanında çalışmalar gerçekleştiren eğitsel veri madenciliği en temel olarak dört bileşenden oluşmaktadır (Dahiya, 2018). Bunlar;

1. Paydaşlar
2. Veri
3. Öğrenme Ortamı
4. Yöntem ve Teknikler

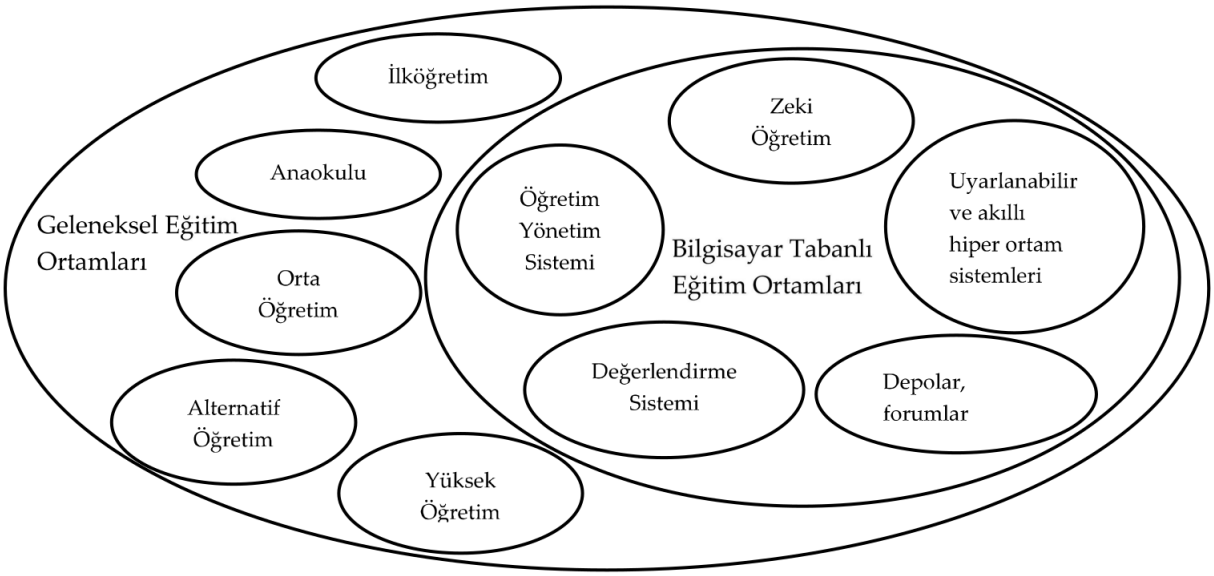
Eğitsel veri madenciliği süreçlerindeki paydaşlar, öğrenen, öğretici, yöneticiler, öğretim tasarımcıları ve ailelerden oluşmaktadır. Kısacası süreçten ve sonuçlardan etkilenecek olan kullanıcılardır. Eğitsel veri madenciliği de mühendislik ve birçok alanda kullanılan sistematik yapı olan girdi, süreç ve çıktı değişkenlerini kullanarak işlem yapmaktadır. Girdi ve süreç değişkenleri etkileyen değişkenler olarak isimlendirilirken çıktı değişkeni ise etkilenen değişkendir. Değişkenler ile ilgili örnek vermek gerekirse; girdi değişkenleri, yaş, cinsiyet, önceki eğitim, giriş puanları gibi öğrencilere ait özellikler olabilir. Süreç değişkenleri, derse katılma, eğitime devam ya da devamsızlık, motivasyon, isteklilik gibi faktörler olabilir. Bunlara karşın çıktı değişkeni asıl elde edilmek istenen değişken, hedef ya da sonuçtur. Örneğin, öğrenci başarısı, performansı ya da öğrencilerin okulu terk etmesi olabilir. Yani bu şekilde yapılandırılmış ve yapılandırılmamış veri olarak ayrılabilen veriler öğrencilerle ilgili toplanan değerleri içermektedir.

Sistematiği ve kullanılan değişkenlerin yanında eğitsel veri madenciliğinin ortam ve çevresi de önemlidir. Bunlar genel olarak veriye ulaşmanın daha kolay mümkün olduğu ortamlardır. Öğrencilerin öğrenmesine destek olan öğrenme yönetim sistemleri ve yönetim bilgi sistemleri bu ortamların en temellerindendir. Verilerin depolandığı veri tabanları, web günlükler, anket ve sosyal ağlar da bu ortamlardan birkaçı olarak sayılabilir. Yani bütüncül olarak veri toplayıp analiz yapılarak sonuçlar üretilebilecek yapılar ortamları oluşturmaktadır. Eğitim ortamları türlerine göre Şekil 2.3'de gösterildiği gibi geleneksel ve bilgisayar tabanlı eğitim olarak ayrılmaktadır (Romero ve Ventura, 2013). Yöntem ve teknikler ise madencilik yapmak için kullanılan araçlardır. Bunlar ihtiyaca göre; sınıflandırma, kümeleme, regresyon ya da ilişki analizleri olabilir ve bu bunları gerçekleştirmek için uygun algoritmaların seçilmesi gerekmektedir.

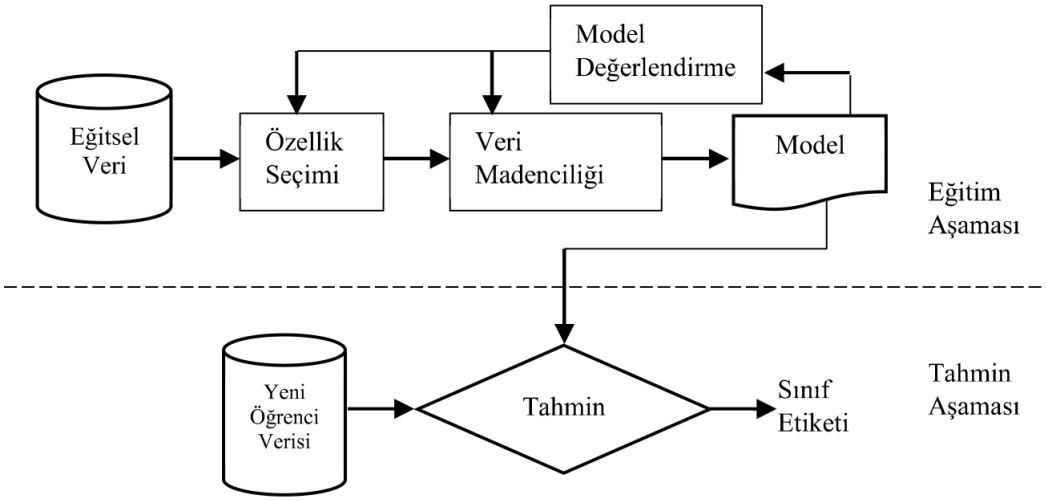
Bu bileşenler kullanılarak gerçekleştirilen eğitsel veri madenciliği sürecinde, öğrenme ortam ve çevrelerinden toplanan veriler çeşitli makine öğrenme yöntem ve teknikleri ile sonuç ve çıkarımlar elde edilerek paydaşlara yol göstermek hedeflenmektedir. Temel olarak hedeflenen, çıktı değişkeni ile girdi ve süreç değişkenleri ilişkilendirerek sonuç elde etmektir.

2.5 Eğitsel Veri Madenciliği Akışı

Eğitsel veri madenciliğinin bütüncül akışı Şekil 2.4'de görüldüğü gibi eğitim ve tahmin olmak üzere ikiye ayrılmaktadır. Eğitim aşamasında ilk olarak veri ön işleme ve öznitelik seçimi ile veri madenciliği teknikleri aracılığıyla model değerlendirilmektedir. Eğer kullanıcı için uygun başarı elde edildiyse tahmin aşamasına geçirir aksi durumda önceki adımlar tekrarlanır ya farklı teknik ya da parametreler denenerek istenen başarı düzeyine ulaşmak amaçlanır. Tahmin aşamasında ise yeni öğrenci verisi sisteme verilerek öğrencinin sınıfının yani başarı ya da başarısız olma durumunun tahmin edilmesi sağlanmaktadır.



Şekil 2.3: Geleneksel ve Bilgisayar Tabanlı Eğitim Ortamları ve Sistemleri



Şekil 2.4: Eğitsel Veri Madenciliği Akışı (Ünal,2020)

2.6 Eğitsel Veri Madenciliği Uygulama Alanları ve Süreçleri

Temel bileşenleri paydaşlar, veriler, öğrenme ortamı, teknikler ve yöntemler olan eğitsel veri madenciliğinin temel uygulama alanları ve süreçler aşağıda sıralanmıştır (Romero ve diğerleri, 2011).

1. Öğrenci modellerinin geliştirilmesi
2. Alanın bilgi yapısının modellerini keşfetmek veya geliştirmek
3. Pedagojik desteği incelemek
4. Öğrenme sistemleri tasarlamak
5. Paydaşlarla iletişim kurmayı sağlamak

Eğitsel veri madenciliğinin en temel noktası elde edilen verilerden sonuç çıkartarak bilgiye ulaşmaktır. İlk uygulama alanı olarak belirtilen 'Öğrenci Modellerinin Geliştirilmesi', öğrenci modelleri, öğrencinin özellikleri veya durumu hakkındaki bilgileri temsil eder. Eğitsel veri madenciliği yöntemleri, gerçek zamanlı olarak ilgili öğrenci niteliklerinin yelpazesini modellemeyi sağlayabilir. Modellenen öğrenci özellikleri, öğrencinin öğrenme stilleri ve tercihler, bilişsel yönler, duygusal özellikler ve üst-bilişsel özellikleri olabilir. Öğrenme stilleri ve tercihlere örnek olarak videolu öğrenme, yazılı dokümanla öğrenme ya da etkileşimli öğrenme gösterilebilir. Bilişsel özellikler öğrencinin bilme olgusunu ve zihinsel süreçleri ele alır, duygusal ve duyuşsal özellikler ise öğrencinin ders motivasyonu, derse olan ilgisi ve sevgisi olabilir. Bu özelliklerde yapılan modelleme sayesinde öğrenme performans tahmini ve öğrenci davranışlarının analizi gibi konularda sonuç üretilebilir.

Chrysafiadi ve Virvou (2013) tarafından öğrenci özellikleri ile ilgili yapılan çalışmalar incelenmiştir. Çalışmada öğrenci özelliklerinin analizi yapılması için bilişsel teoriler, bulanık mantık, bayes ağları, stereotip, kısıtlar teorisi ve makine öğrenme yöntemlerinin kullanıldığı görülmüştür. Eğitsel veri madenciliği çalışmalarını gerçekleştirmek için kullanılan makine öğrenme yöntemlerinin öğrenci özelliklerine göre kullanım oranları Tablo 2.1'de sunulmuştur.

Tablo 2.1: Makine Öğrenme Yöntemlerinin Öğrenci Özelliklerine Göre Kullanım Oranları

	Öğrenci Özellikleri						
	Bilgi	Hatalar	Öğrenme Stilleri	Bilişsel Yönler	Duygusal Özellikler	Motivasyon	Üst Bilişsel Özellikler
Kullanım Oranı	13.33	10.53	10.26	15.15	16.67	30	0

Tablo 2.1 incelendiğinde çalışmalarda makine öğrenme yöntemlerinin en çok motivasyon ile ilgili konularda kullanıldığı görülmektedir. Bunun yanında duyuşsal özellikler ve diğer bilişsel yönleri tahmin etmede ve bilgiyi modelleme de makine öğrenmesi teknikleri daha yoğun kullanılmıştır. Bunların aksine üst bilişsel özellikleri modellemek için makine öğrenme yöntemleri hiç kullanılmamıştır. Bu konuda en çok bayes ağları yöntemi kullanılmıştır. İkinci sırada sunulan uygulama alanı "Alan Bilgi Yapısı"dır. Alan bilgi yapısının modellerini keşfetmek veya geliştirmek için makine öğrenme algoritmaları ile verilerden doğru alan yapısı modellerini keşfedebilen otomatik yaklaşımlar geliştirilebilir. Üçüncü uygulama alanı "Pedagojik Destek"tir. Hangi gruba hang tür pedagojik desteğin etkili olduğunu belirlemek için kullanılan öğrenme ayrıştırması yöntemi, öğrencinin ileri dönemdeki başarısını, öğrencinin o noktaya kadar aldığı her türlü pedagojik desteğin miktarı ve türü ile ilşkilendirmektedir. Bu amaçla, öğrencinin eğer uzman desteğine ihtiyacı varsa öğretici ile eğer akran desteğine ihtiyacı varsa bu durumda ihtiyaç duyulan konu hakkında bilgisi olan bir akran ile iletkileşime geçmesine yönelik destek sunulmaktadır. Ayrıca bazı öğrenciler dinleyerek bazı öğrenciler uygulayarak bazıları ise örnek çözerek öğrenmektedir bu nedenle sunulan pedagojik destek farklılaşmalıdır. En uygun modelde her tür pedagojik destek için göreli ağırlıklar, öğrenmeyi teşvik etmek için her tür desteğin göreli etkinliğini çıkarmak için kullanılabilir.

Dördüncü uygulama alanı "Öğrenme Sistemleri Tasarlamak"tır. Makine öğrenme algoritmaları ile öğrenmeyi etkileyen temel faktörlerin daha derinden anlaşılmasına yönelik eğitim teorilerini geliştirme ve genişletme hedeflerine ulaşılabilir. Sistem tasarımı, alan bilgisi yapısında keşfedilen modeller ile pedagojik destek sistemi birleşerek gerçekleştirmekte ve öğrenciye sunulmaktadır.

Beşinci uygulama alanı ise "Paydaşlarla İletişim Kurmayı Sağlamak"tır. Eğitsel veri madenciliği teknikleri kullanılarak veriler ile ilgili elde edilen kuralla ve tahminler paydaşlara yol göstererek ders ve bölüm yöneticilerine ve eğitimcilere yardımcı olmak sağlanabilmektedir. Paydaşlar en genel bağlamda öğrenci, öğretici, yönetici ve ailelerden oluşmaktadır. Çalışmalar da daha çok öğrenci üzerine odaklanarak öğrenci unsurlarının ele alındığı görülmektedir. Ancak öğrenci verilerinin yanında öğretici ve aile paydaşlarının da ön planda tutulması gerektiği ve bunlar ile ilgili çalışmalar yapılması gerektiği de unutulmamalıdır. Öğretmenlerin ders işleyiş yöntemleri, derste kullanılan materyaller, ders hazırbulunuşluğu gibi faktörlerdir. Yöneticilerin yönetimsel faaliyetlerde aldığı kararlar ve öğrencinin ailesinin tutumlarının da öğrenci başarısına etkilerinin de çok önemli olduğunun bilincinde olup ortam ve verilerin bu paydaşların açısından da incelenmesi gerekmektedir.

2.7 Eğitsel Veri Madenciliğinde Ele Alınan Değişken ve Konular

Eğitsel veri madenciliğinde modelin tahmin doğruluğunun yüksek olmasının yanında hangi değişkenlerin kullanıldığı ve bunların ilişkileri ve paydaşlar üzerindeki etkileri de çok önemlidir. Bu alanda sıkça çalışılan konular temel olarak aşağıdaki dört başlık altında toplanabilir (bkz. Rodrigues ve diğerleri 2018, Baradwaj ve Pal 2012, Fernandes ve diğerleri 2019, Agrusti ve diğerleri 2019, Dekker ve diğeri 2009).

1. Akademik başarı,
2. Öğrenme süreci,
3. Okuldan ayrılan öğrenciler,
4. Öğrenci performansı

Akademik başarı konusunda bir öğrencinin öğretim sisteminde başarılı olup olamayacağı ile ilgili çalışılmaktadır. Bu alanda etkili olan değişkenler belirlenerek öğrencilere sunulacak destek hizmetlerinin de belirlenmesi sağlanabilmektedir. Öğrenme süreci konusunda ise öğrencinin sistemdeki hareketleri yani neler yaptığı, sistemde ne kadar ve nasıl vakit geçirdiği gibi konularla ilgili çalışılmaktadır. Bu öğrencilerin nasıl yönlendirmesi gerektiği ile ilgili fikir vermektedir. Eğitsel veri madenciliği çalışmalarındaki bir diğer önemli nokta ise bırakma oranlarıdır. Bunun için okuldan ayrılan, bırakan, terk eden öğrencilerin incelenmesi gerekmektedir. Bu konu özellikle de uzaktan eğitim sistemlerinde önemle çalışılmaktadır çünkü daha sık rastlanmaktadır ve çözülmesi gereken bir sorun olarak görülmektedir. Bu çalışma konusu ile hem yüz yüze hem de uzaktan eğitimi bırakan öğrenciler incelenerek sebepleri bulunup ortadan kaldırılmaya çalışılmaktadır. Öğrenci performansı konusunda ise öğrencilerin hangi derste hangi faaliyetleri ne şekilde gerçekleştirerek başarılı olduğunu araştırarak ortaya çıkartmak hedeflenmektedir.

2.8 Eğitimde Kural Çıkarımı

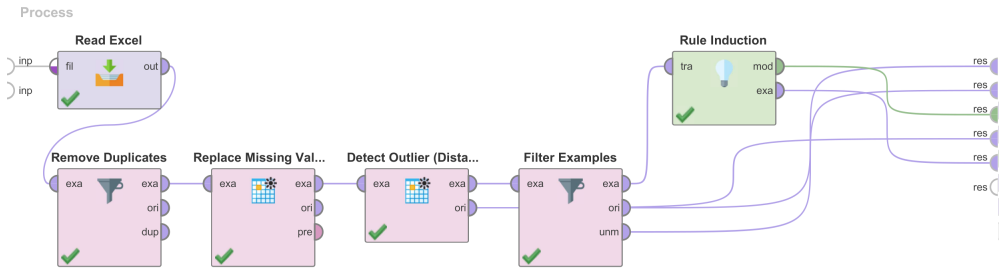
Eğitsel veri üzerinden anlamlı sonuçlara ulaşabilmek için kullanılacak yöntemlerinden biri de tahminleyici veri madenciliği algoritmalarından olan kural çıkarımı (rule induction)dır. Kural çıkarımı, temel olarak bilgi-teorik yaklaşımlara dayalı olarak özellik uzayı içindeki bölümlere veri atayarak belirsizliği (entropi) azaltma işlemidir (Luan, 2002). Bu işlem gerçekleştirilirken karar ağaçlarını veya benzer bilgi yapılarını kullanarak örnekleri karar ağacının dallarına göre sıralanır

veya genellikle ya hep ya hiç eşleşme süreci ile koşulları örnekle eşleşen ilk kuralı bulunur. Sınıflar veya tahminler hakkındaki bilgiler, kuralların eylem taraflarında veya ağacın yapraklarındadır. Kural tümevarım çerçevesindeki öğrenme algoritmaları, genellikle, bilgi yapısına dahil edilmek üzere nitelikleri seçmek için tipik olarak istatistiksel bir değerlendirme işlevi kullanarak açgözlü bir arama gerçekleştirir. Çoğu yöntem, eğitim verilerini yinelemeli olarak ayrık kümelere böler ve her kümeyi mantıksal koşulların bir birleşimi olarak özetlemeye çalışır (Langley ve Simon, 1995). Kural çıkarımı uygulamadan önce verinin hazırlanması önemli bir adımdır. Örneğin sayısal verilerde kural tümevarımı uygulaması için ya kategorileştirme ya da dönüştürme işlemleri yapılmalıdır. Yani kural çıkarımı uygularken de ilk olarak ön işleme bir ön işleme olarak gerçekleştirilmelidir (Busse, 2003). Elde edilen kurallar bir öğrenme yönetim sisteminde başarılı veya başarısız olmaya iten faktörleri belirlemek, öğrencilerin davranışlarını anlamak ve bu anlayıştan yola çıkarak, öğretme ve öğretme sürecini iyileştirebilecek eylemleri önermeyi sağlamaktadır.

2.9 Uygulama

Uygulama çalışmasında, UCI veri deposundan "öğrenci performansı (student performance)" veri seti kullanılarak kural çıkarımı gerçekleştirilmiştir. Bağımsız ya da girdi ve süreç değişkenleri Tablo 2.2'de gösterilmiştir. İlk 21 bağımsız değişken girdileri son 9 değişken ise süreç değişkenleridir. Öğrenci Başarı Durumu yani bağımlı ya da çıktı değişkeni çok iyi, iyi, yeterli ve başarısız olarak kategorileştirilmiştir.

Örnek uygulamada, araştırma sorusu "Başarıya etki eden değişkenler ile öğrencilerin ders başarı performansları ile arasında nasıl ilişkiler vardır?" şeklindedir. Uygulama Rapidminer 9.9 ile gerçekleştirilmiştir. İlk olarak veri setinde Şekil 2.5'deki gibi ön işleme adımları uygulanarak ardından kural tümevarım algoritması ile öğrenci başarısı ile etki eden değişkenlerin kuralları elde edilmiştir.



Şekil 2.5: Rapidminer ekran görüntüsü

2.10 Kurallar

Öğreticilere ve yöneticilere öğrenci başarısını etkileyen faktörler ile ilgili fikir vermek için eğitsel veri madenciliği ve makine öğrenme yöntemlerinden olan kural çıkarımından faydalanılmaktadır. Elde edilen kurallar modeli temsil etmektedir. Ancak burada ki önemli nokta her kuralın kullanılabilir veya anlamlı olmadığını bilmektir. Sadece anlamlı ve işe yarar olan kurallar yorumlanmalıdır aksi durum zaman kaybı ve karmaşıklığa neden olacaktır. Çalışmada toplamda 15 tane kural elde edilmiştir. Ancak bunlardan üç tanesi bir önceki kuralın parçası niteliğinde iki tanesi de anlamlı olarak bulunmamıştır. Yorumlanabilir durumdaki altı tane kural aşağıda sıralanmıştır.

Tablo 2.2: Örnek Uygulama Bağımsız Değişkenleri

No	Bağımsız Değişkenler	Değişken Tanım	Değişken Türü
1	Okul	Öğrencinin okulu	İkili
2	Cinsiyet	Öğrencinin cinsiyeti	İkili
3	Yaş	Öğrencinin yaşı	Sayısal
4	Adres	Öğrencinin ev adres türü	İkili
5	Aile Büyüklüğü	Ailedeki birey sayısı	İkili
6	Aile Durumu	Ebeveynin birlikte yaşama durumu	İkili
7	A_Eğitim	Annenin Eğitimi	İkili
8	B_Eğitim	Babanın Eğitimi	Sayısal
9	A_İş	Annenin işi	Nominal
10	B_İş	Babanın işi	Nominal
11	Neden	Bu okulu seçme nedeni	Nominal
12	Veli	Öğrenicinin velisi	Nominal
13	Seyahat Süresi	Evden okula seyahat süresi	Sayısal
14	Çalışma Süresi	Haftalık çalışma süresi	Sayısal
15	Hata	Geçmiş sınıf hatalarının sayısı	Sayısal
16	E_Destek	Ekstra eğitim desteği	İkili
17	A_Destek	Aile eğitim desteği	İkili
18	ÜcretliDers	Ekstra ücretli kurslar	İkili
19	Etkinlik	Müfredat dışı etkinlikler	İkili
20	Kreş	Anaokuluna gitme durumu	İkili
21	YüksekÖğrenim	Yükseköğrenim almak isteme durumu	İkili
22	İnternet	Evde internet olma durumu	İkili
23	Romantik	İlişki durumu	İkili
24	A_İlişkiKalite	Aile ilişkilerinin kalitesi	Sayısal
25	BoşZaman	Okuldan sonra boş zaman	Sayısal
26	Gezi	Ailelerle dışarı çıkmak	Sayısal
27	Sağlık	Mevcut sağlık durumu	Sayısal
28	Devamsız	Okula devamsızlık gün sayısı	Sayısal
29	D1	Birinci dönem başarı notu	Sayısal
30	D2	İkinci dönem başarı notu	Sayısal

- Kural 1: İkinci dönem başarı notu $10,5 < D2 \leq 14,5$ ve aile arası ilişkileri mükemmele yakın olan kız öğrencilerin başarı durumu iyidir.
- Kural 2: Öğrencinin ikinci dönem başarı notu D2>14,5 ve okula seyahat süresi 1.5 saatten az ise öğrenci başarısı iyidir.
- Kural 3: Birinci dönem başarı notu D1>10,5 ve ikinci dönem başarı notu $D2 \leq 13,5$ olan erkek öğrencilerden babası yükseköğretim mezunu olan olanlar dönem sonunda yeterli başarıya ulaşmaktadır.
- Kural 4: "X" okulunda okuyan 16.5 yaşından küçük tüm öğrenciler dersten kalmıştır.
- Kural 5: Haftalık çalışma süresi 1.5 saatten az olan ve duygusal bir ilişkisi olan tüm öğrenciler dersten kalmıştır.
- Kural 6: Şehirde yaşan öğrencilerden ikinci dönem başarı puanı 10.5 dan az olan, devamsızlığı 2 günden çok alan ve arkadaşlarıyla sık gezmeğe çıkan öğrencilerin tamamı dersten kalmıştır.

2.11 Kurallardan elde edilen bulgular

Kural çıkarımı gerçekleştirilerek birtakım kurallar elde edilmiştir ancak önemli olan bu kuralları yorumlamak ve bunlara yönelik alınacak tedbir, önlem ve politikaları belirlemektir. Eğitsel veri madenciliği bu amaçla kullanılarak, kurallar aşağıdaki gibi yorumlanabilir;

- Sınıftaki öğrenciler arasından ikinci dönem başarısı 10,5 ile 14,5 arasında olan ve aile ilişkisi çok iyi olan kız öğrenciler derslerde iyi başarı elde etmektedir.
- İkinci başarı notu 14ten yüksek ve seyahat süresi kısa olan öğrenciler de iyi başarı elde etmektedir.
- Bir okuldaki öğrencilerden 16.5 yaşından küçük olan tüm öğrenciler başarısı olmuştur.

Bu kurallardan aile durumu iyi olan öğrencilerin başarısının iyi olduğu, okula seyahat süresinin başarı üzerinde etkisi olduğu ve seyahat süresinin kısa olmasının öğrenci başarısını olumlu etkilediği anlaşılmaktadır. Bu durumda aileler için öneriler mümkün olduğu kadar yakın okula çocuklarını göndermek önerilebilir. Ailelerin yanında öğretmenler açısından aile durumu iyi olan çocukların başarılı olduğu ancak tersine aile durumu iyi olmayanlar için tedbir almaları gerektiği ifade edilebilir. Ayrıca okul ya da bakanlığın alması gereken karar çocukların en yakın okula kaydolmasına yönelik politika üretilmesi gerektiği vurgulanabilir.

İlk iki kuralda başarı notu sadece birinci döneme bağlı iken önemli bulgulardan biri okula seyahat süresi ve aile iletişimi iken üçüncü kuralda birinci dönem notu ve baba mesleği eklenmiştir. Buna göre babası yükseköğretim mezun olan erkek öğrenciler yeterli düzeyde başarı elde etmektedir. Buradan yola çıkılarak babası yükseköğretim mezunu olmayan öğrencilerin derslerde sıkıntı yaşayabileceği tahmin edilmektedir. Buna yönelik olarak öğretmen hem ailelere hem de öğrencilerine yönelik önlemler alması gerekmektedir.

Bu noktaya kadar olan kurallardan çıkan sonuçlar paydaşlar için yol gösterir. Bu önerilere göre öğretmen, yönetici ve ailelere öneriler sunulur. Örneğin, Öğrencilerin başarı notlarına göre öğretmenlere öneriler sunulabilir. Başarı notu düşük olan öğrencilere ilgili dönemden ders tekrarı, ek not ya da kaynak paylaşımı yapılabilir.

Aile içi sorunlar yaşandığı için başarısı düşük olan öğrencilerin ailelerine aile sorunları öğrencilere yansıtmamaları önerilebilir. Bunu yanında okula seyahat süresi uzun olan öğrencilerin ailelerine bu konuda çözüm bulmaları önerilebilir. Bu konu da yöneticilere de okula uzak mesafeden öğrencilerin kaydını almaması önerilebilir.

Ayrıca 16.5 yaşından küçük çocukların başarısız olduğu, okul yöneticilerine ise bu okulda belli bir yaş olgunluğunda olmayan öğrencilerin başarısız olduğu bildirilerek bu yaş grubuna ilave bir

destek verilmesi önerilebilir. Örneğin, seçmeli derslerde bir model geliştirilerek ders a ve b grubu olarak bölünebilir. A grubunda 16,5 dan küçükler olarak bunlar için özel bir eğitim geliştirilebilir ya da bu dersi hiç almayabilirler.

Sonuç olarak eğitsel veri madenciliği ile tespit ettiğimiz konularda politika geliştirip paydaşlara öneride bulunmanın alandaki temel nokta olduğu görülmektedir. Sonuçlar ile ilgili kararlar almak, faaliyetler gerçekleştirmek ve iyileştirmeler yapmak paydaşlara bağlıdır.

Eğitsel veri madenciliğinde elde edilen bulgular genellikle öneri geliştirme ve politika üretme boyutunda kalmaktadır. Eğitsel veri madenciliği ile elde edilen veriler ışığında öğrenme analitikleri ismini verdiğimiz yazılımlar yoluyla veriler ışığında müdahaleli süreçler oluşabilmektedir. Bu yönüyle bölümde farkındalık amacıyla kısa şekilde öğrenme analitiklerinden bahsedilecektir.

2.12 Öğrenme Analitikleri ve Kullanım Amacı

Öğrenme Analitiği, öğrenmeyi ve öğrenmenin gerçekleştiği ortam ve süreçleri anlamak ve iyileştirmek amacıyla öğrenenler ve bağlamları hakkında verilerin ölçülmesi, toplanması, analiz edilmesi ve raporlanmasıdır. Bir araştırma ve öğretme alanı olarak Öğrenme Analitiği, öğrenme, analitik ve insan merkezli tasarımın birleştiği noktada yer alır (SOLAR, 2012). En sık kullanım amacı, öğrencinin akademik başarısının tahmini yani bir dersten kalma veya eğitimini bırakma riski altında olan öğrencilerin belirlenmesidir.

2.13 Öğrenme Analitiğinin Hedefleri

Öğrenme Analitiği Araştırmaları Derneği (SOLAR) tarafından öğrenme analitiği alanı için belirlenen en popüler hedefler aşağıdaki gibi sıralanabilir:

- Öğrencilerin yaşam boyu öğrenme becerilerinin ve stratejilerinin gelişimini desteklemek,
- Öğrencilere öğrenmeleri ile ilgili kişiselleştirilmiş ve zamanında geri bildirim sağlanmak,
- Kendini yansıtmayı destekleyerek öğrenci farkındalığını geliştirmek,
- İşbirliği, eleştirel düşünme, iletişim ve yaratıcılık gibi önemli becerilerin gelişimini desteklemek,
- Pedagojik yeniliklerin başarısı hakkında ampirik kanıtlar sağlayarak kaliteli öğrenmeyi ve öğretmeyi desteklemektir.

2.14 Öğrenme Analitiklerinin Eğitimde Kullanılmasının Faydaları / Etkilediği Alanlar?

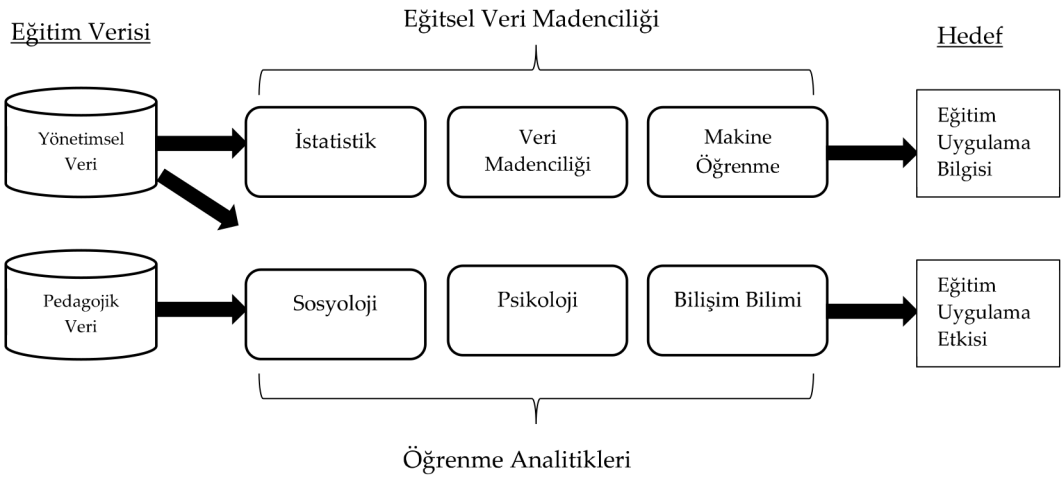
Eğitim verilerden yola çıkarak eğitimsel farklı yapı ve modeller geliştirmeyi hedefleyen öğrenme analitikleri alanının kullanılmasının fayda sağladığı alanlar aşağıdaki gibi sıralanabilir (Avella ve diğerleri,2016):

- Hedef dersler/programlar belirleme,
- Öğretim programı iyileştirme,
- Öğrenci öğrenme sonucu, davranışı ve süreci,
- Kişiselleştirilmiş öğrenme,
- Eğitim sonrası istihdam,
- Öğrenme analitiği uygulayıcıları ve araştırma topluluğudur.

Tüm bu süreçler için kullanılabilen akademik analitik, eğitsel öneri sistemi, zeki öğrenme sistemleri, uyarlanabilir öğrenme sistemleri, akademik erken uyarı sistemleri gibi öğrenme analitikleri türlerinden yararlanılmaktadır.

2.15 Öğrenme Analitiği ve Eğitsel Veri Madenciliği İlişkisi

Eğitsel veri madenciliği eğitim verilerinden anlamlı sonuçlar elde etmek olarak tanımlanırken öğrenme analitikleri ise elde edilen bu bulguların kullanılması olarak yorumlanmaktadır. Eğitsel veri madenciliği ve öğrenme analitikleri ile ilgili farklılıklar ve ilişkiler Şekil 2.6'da gösterilmektedir. Eğitsel veri madenciliğinde yönetimsel veriler istatistik, veri madenciliği ve makine öğrenme yöntemleri kullanılarak eğitim için paydaşlara öneriler sunulmaktadır. Öğrenme analitikleri ise hem yönetimsel hem de pedagojik verileri sosyoloji, psikoloji ve bilişim bilimleri alanlarından yararlanarak eğitimde uygulama etkisine dönüştürmeyi amaçlar (Ahmadu, 2017).



Şekil 2.6: Eğitsel veri madenciliği ve Öğrenme Analitikleri İlişkileri (Strecht ve diğ. 2014)

2.16 Eğitsel Veri Madenciliği ile İlgili Olası Yaşanabilecek Sorunlar

Eğitsel veri madenciliği çalışmaları gibi bir konu ile ilgili toplanmış veriler üzerinde çalışmalar yapılıyorsa bu durumda veri ile ilgili bazı sorunlar yaşanabilmektedir. Bunlardan en temel olanlar aşağıda sıralanmıştır:

1. Kişisel Verilerin Korunumu: Verilerin toplanması sırasından bazen bir bazen birden çok veri tabanı kullanılabilir. Birden çok veri tabanından veri eşleştirmek için kimlik numarası gibi kişisel özellik taşıyan belirleyici bilgilere ihtiyaç duyulabilir. Bu gibi çok özel verilerin kullanılması ciddi sorunlara neden olabilir.
2. Etik: Bazı durumlarda toplanan veriler çok özel kişisel veriler içeriyor olabilir bu gibi bir durumda bu verilerin kullanılması ya da paylaşılması etik dışı olabilir.
3. Verilerin yönlü olması: Veriler dolayısıyla elde edilen sonuçlar kişileri yönlendirecek şekilde düzenlenmiş olabilir. Bunlar gerçek dışı olursa hatalı bilgi ve önerilere neden olabilir.
4. Eğitsel verilerde duyuşsal ve psikomotor özellikler: Öğrenme davranışı bilişsel, duygusal ve psikomotor bileşenleri bir araya geldiğinde ortaya çıkmaktadır. Değişkenler net durumları

belirtiyorsa (örneğin yaş,cinsiyet) bunlarla ilgili yorum yapmak çıkarımlarda bulunmak nispeten kolaydır ancak duygu durumu ya de el becerileri gibi özelliklerle ilgili değişkenlerin ölçümlerini yapmak oldukça zordur. Eğitsel verilerde ise ağırlıklı olarak başarı gibi bilişsel özellikler ölçülerek değerlendirilmektedir.

5. Nitel veriler: Sayısal değişkenleri analiz ederek yorumlamak biraz zordur çünkü anlam kayıpları yaşanabilmektedir. Eğitsel veri madenciliğinde sayısal değerler genellikle kodlanarak çalışılmaktadır. Aslında eğitsel veri madenciliğinde nicel veriler üzerinden işlem yapılmaktadır. Nitel ve derinlemesine olan veriler çok fazla dikkate alınamamaktadır.

Veriler ile ilgili yaşanan sıkıntıların yanında eğitsel veri madenciliği teknikleri ve öğrenme analitikleri kullanılarak istenen faydaların sağlanabilmesi için bazı zorlukların aşılması gerekmektedir. Bunlar:

1. Anlamlı ve işe yarar verilerin toplanması ve depolanması
2. Verilerin devamlılığının olması
3. Geliştirilen politikaların anlaşılabilir kullanılabilir olması
4. Günlük yaşantıda işe yarar olması

2.17 Kaynaklar

Aggarwal, C. C. (2015). Data mining: the textbook. Springer.

Agrusti, F., Bonavolontà, G., & Mezzini, M. (2019). University Dropout Prediction through Educational Data Mining Techniques: A Systematic Review. Journal of E-Learning and Knowledge Society, 15(3), 161-182.

Ahmadu, A. S., Boukari, S., Garba, E. J., & Danjuma, K. J. (2017). Simulation of the Framework for Evaluating Academic Performance (FEAP) using WEKA. International Journal of Scientific & Engineering Research, 8(10), 1201-1217

Avella, J. T., Kebritchi, M., Nunn, S. G., & Kanai, T. (2016). Learning analytics methods, benefits, and challenges in higher education: A systematic literature review. Online Learning, 20(2), 13-29.

Baker, R. S., & Yacef, K. (2009). The state of educational data mining in 2009: A review and future visions. Journal of educational data mining, 1(1), 3-17.

Baradwaj, B. K., & Pal, S. (2012). Mining educational data to analyze students' performance. arXiv preprint arXiv:1201.3417.

Chung, H. M., & Gray, P. (1999). Data mining. Journal of management information systems, 16(1), 11-16.

Chrysafiadi, K., & Virvou, M. (2013). Student modeling approaches: A literature review for the last decade. Expert Systems with Applications, 40(11), 4715-4729.

Dahiya, V. (2018). A survey on educational data mining. International Journal of Research in Humanities, Arts and Literature, 6(5), 23-30.

Dekker, G. W., Pechenizkiy, M., & Vleeshouwers, J. M. (2009). Predicting Students Drop Out: A Case Study. International Working Group on Educational Data Mining.

Fernandes, E., Holanda, M., Victorino, M., Borges, V., Carvalho, R., & Van Erven, G. (2019). Educational data mining: Predictive analysis of academic performance of public school students in the capital of Brazil. Journal of Business Research, 94, 335-343.

García, S., Luengo, J., & Herrera, F. (2015). Data preprocessing in data mining (Vol. 72). Cham, Switzerland: Springer International Publishing.

Grzymala-Busse, J. W. (2003). A comparison of three strategies to rule induction from data with numerical attributes. Electronic Notes in Theoretical Computer Science, 82(4), 132-140.

Langley, P., & Simon, H. A. (1995). Applications of machine learning and rule induction. Communications of the ACM, 38(11), 54-64.

Luan, J. (2002). Data Mining and Knowledge Management in Higher Education-Potential Applications.

Rodrigues, M. W., Isotani, S., & Zarate, L. E. (2018). Educational Data Mining: A review of evaluation process in the e-learning. Telematics and Informatics, 35(6), 1701-1717.

Romero, C., López, M. I., Luna, J. M., & Ventura, S. (2013). Predicting students' final performance from participation in on-line discussion forums. Computers & Education, 68, 458-472.

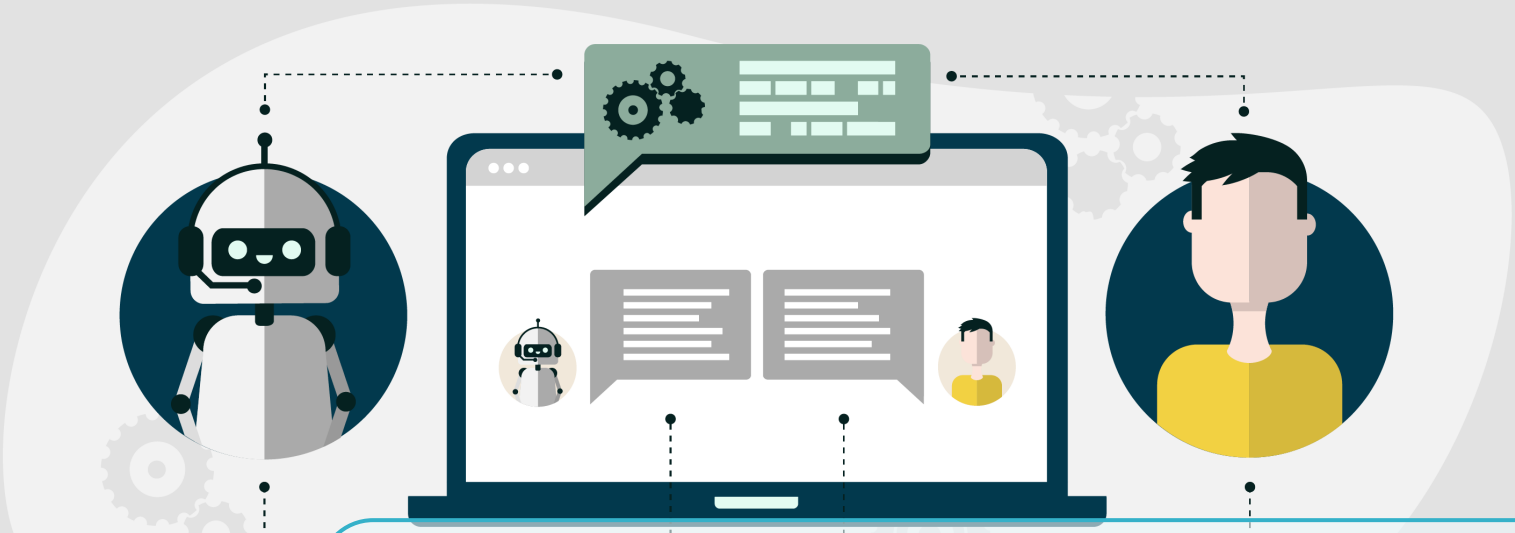
Romero, C., Ventura, S., Pechenizkiy, M., & Baker, R. S. (Eds.). (2010). Handbook of educational data mining. CRC press.

Strecht, P., Moreira, J. M., & Soares, C. (2014). Educational data mining: preliminary results at university of porto.

Ünal, F. (2020). Data mining for student performance prediction in education. Data Mining-Methods, Applications and Systems.

`https://www.solaresearch.org/about/what-is-learning-analytics`

`https://archive.ics.uci.edu/ml/datasets/student+performance#`



3. Üretim Sistemleri İçin Dijital İkiz Tasarımı

Üretim Sistemleri İçin Dijital İkiz Tasarımı

Muhammet Raşit CESUR*, Elif CESUR†

***Endüstri Mühendisliği, İstanbul Medeniyet Üniversitesi, Türkiye, rasit.cesur@medeniyet.edu.tr**

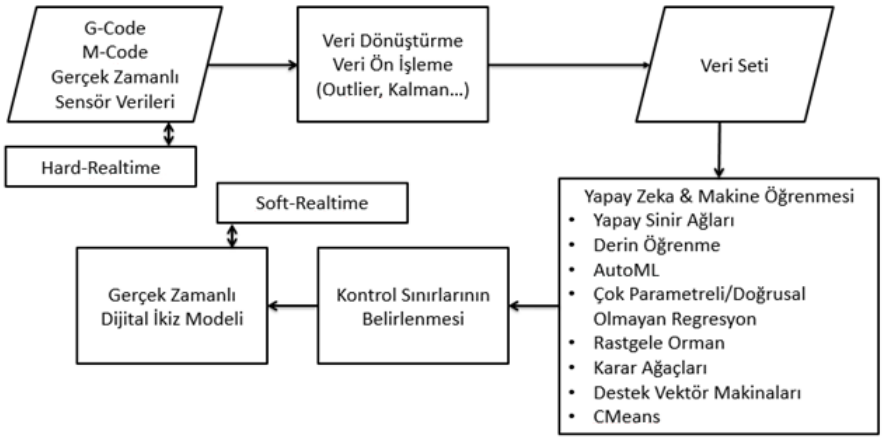
†Endüstri Mühendisliği, İstanbul Medeniyet Üniversitesi, Türkiye, elif.karakaya@medeniyet.edu.tr

3.1 Giriş

Dijital ikiz temel olarak gerçek bir sistemin dijital kopyası olarak adlandırılabilir. Bu bağlamda dijital ikiz teknolojisinin en önemli katkısı, dijital ikizi üretilen sistem veya malzemenin gerçek ortamda hiç çalışmadan sanal ortamda test edilebilir veya çalıştırılabilir olmasıdır. Dijital ikiz sayesinde ekstra maliyete katlanmadan, zaman ve kaynak kaybı olmadan gerçek bir işlem sanal ortamda icra edilebilmektedir. Dolayısıyla, hiç yapılmamış veya çalıştırılmamış, geçmiş verisi olmayan, bir işe veya sisteme ait veriler dijital ikiz teknolojisi sayesinde sanal ortamda elde edilebilir. Dijital ikiz teknolojisini simülasyondan ayıran en önemli özelliği de geçmiş veri veya geçmiş veriye dayalı bir dağılımın parametrelerine ihtiyaç duymamasıdır. Dijital ikizin geçmiş veriye dayanmadan gerçek hayatta oluşan sonuçlar ile özdeş çıktılar üretebilmesini sağlayan temel yapı dijital ikizi oluşturulacak sistemin çalışma sistematiğinin modelidir. Bu model sayesinde, bir sistem belirlenen parametrelerle (hız, miktar vb.) dijital ortamda çalıştırılır ve gerçek ortamda benzer parametrelerle çalıştırıldığında benzer sonuçların (enerji tüketimi, işlem süresi vb.) elde edildiği gözlemlenecektir. Bu sayede bir motorun çalışma süresi boyunca tüketeceği yakıttan, ilk defa üretilecek bir ürünün üretim süresinin hesaplanması ve çizelgelenmesinden, kestirimci bakım çalışmalarına kadar geniş bir alanda uygulama yapma imkanı olmaktadır.

Dijital ikizin oluşturulması bir modele bağlı olarak gerçekleştirileceğinden dijital ikiz oluşturma sürecinde veri ön işleme, modelin oluşturulması ve modelin doğrulanması süreçlerinin olması gerekmektedir. Bu süreçlere ek olarak veri toplama ve entegrasyon süreci ile kontrol ve geri besleme adımları eklendiğinde dijital ikizin oluşturulması ve çalıştırılması için gerekli akış Şekil 1'de görüldüğü gibi tasarlanmış olacaktır. Dijital ikiz modeli oluşturulurken ilk aşamada dijital

ikizi oluşturulacak sistemin çalıştırılması esnasında, sistemin çalışmasına yönelik veriler sensör yardımıyla toplanarak gerçek zamanlı olarak dijital ikiz yazılımına iletilmelidir. Toplanan veriler gürültü giderme ve düzeltme işleminden geçirilerek analize hazır hale getirilmelidir. Geliştirilecek dijital ikiz modelinin ihtiyaçlarına göre veri setine normalizasyon işlemi de uygulanabilir. Veri seti hazırlandıktan sonra girdi olarak kullanılan sensör verileri ile hesaplanmak istenen çıktılar arasında ilişki kuracak bir makine öğrenmesi veya yapay zekâ modeli geliştirilecektir. Geliştirilen model ile sistemin çalışma sistematiği modellenmiş olacaktır. Bir sonraki aşama ise modelin test edilerek doğrulanmasıdır. Geliştirilen modelin gerçek hayatta elde edilen sonuçlarla benzer sonuçlar elde ettiği anlaşıldığında dijital ikizin uygulanması ve geri besleme sağlaması aşamalarına geçilir.



Şekil 3.1: Dijital ikiz tasarlama ve uygulama süreci

Dijital ikiz modeline örnek olarak bir CNC freze tezgâhının dijital ikizinin oluşturulması verilebilir. Tezgâhın dijital ikizi tezgâhta üretilecek bir ürünün G-Code komutlarına bağlı olarak işlem süresini, toplam enerji tüketimini ve sensör verilerine dayanarak tezgâhta arıza veya arıza başlangıcı olup olmadığını analiz edecektir. Bu sayede tezgâhta hiç üretilmemiş olan bir ürünün üretim süresi ve üretim boyunca toplam enerji tüketimi önceden hesaplanabilecek, üretim boyunca tezgâhın arızalanma riski önceden analiz edilebilecektir.

3.2 Model Oluşturma Süreci

Standart bir model oluşturma süreci dijital ikiz teknolojisinin yaygınlaştırılması açısından bir gereklilik olmakla birlikte, dijital ikiz teknolojisinin geniş bir alanda uygulanmasına olanak sağlayacak kapsamda tasarlanması önemlidir. Bundan dolayı, dijital ikiz oluşturma süreci tüm sistemlerden sensör verisi alınarak bu veriler arasındaki ilişkinin çözülmesi ve sisteme geri beslenme sağlanması biçiminde tasarlanmıştır.

3.2.1 Veri Toplama

Veri toplama aşamasında analiz edilecek sisteme doğrudan bağlanacak sensörler yoluyla veri elde edilebileceği gibi, sistemin bilgisayarı ve kontrol kartları arasındaki iletişime sızılarak bilgi elde etmek mümkündür. İletişime sızılması sniffer yazılımları aracılığıyla sistem bilgisayarındaki seri portların veya eternetin dinlenmesi olabileceği gibi, Proxy arayüzler oluşturarak seri portun, ortadaki adam saldırısı ile de eternet kartının dinlenmesi yoluyla olabilir.

3.2.2 Veri Dönüştürme

Veri dönüştürme aşamasında veri üzerinde yapılan işlemlerle verilerdeki eksikliklerin ve hataların giderilmesi veya verinin değiştirilmesi yoluyla standart hale getirilmesi sağlanır. Veri dönüştürme işlemi veri temizleme, veri tamamlama, veri birleştirme ve veri dönüşümü aşamalarını içermektedir. Veri temizleme aşamasında aykırılıkları saptayarak temizlemeye ve tutarsızlıkları gidermeye odaklanılmaktadır. Bu işlemler gürültü giderme olarak da adlandırılmaktadır. Ayrıca veri temizleme aşamasında veri setinin benzer sonuçlar üretecek biçimde küçültülmeye çalışılması da değerlendirilir. Veri tamamlama aşamasında eksik veriler istatistiksel dağılımlar veya eğri uydurma yoluyla üretilmektedir. Veri birleştirmede farklı veri kaynağındaki veriler birleştirerek bir veri seti haline getirilir. Veri dönüştürme aşamasında, veriyi analiz yöntemlerinde kullanıma uygun hale getirmek veya analiz için gerekli parametre değerlerinin oluşturulabilmesi için veriye çeşitli işlemler uygulanabilir. Bu işlemlerden en sık kullanılanı normalizasyondur.

3.2.3 Modelin Oluşturulması

Model oluşturma işlemi, dijital ikiz teknolojisi açısından değerlendirildiğinde, sistemden toplanan değerler ve ölçülen sonuçlar arasında ilişki kuracak yapıyı ortaya çıkarma işlemidir. Bu işlem girdilerle çıktılar arasında bir bağıntı kurma işlemi olduğundan dijital ikiz modeli eğri uydurma yöntemleri veya yapay zekâ ile tahmin modelleri kullanılarak geliştirilmektedir.

Eğri Uydurma

Eğri uydurma, birbiri ile ilişkili iki veya daha fazla noktalar kümesi arasındaki ilişkiyi temsil edecek fonksiyonun bulunmasıdır. Eğri uydurma işleminin sonucunda doğrusal veya doğrusal olmayan ilişkiler tespit edilir. İlişkinin tespit edilmesinde seriler (örn. Fourier Serisi), Vandermonde sistemi, en küçük kareler yöntemi ve Newton interpolasyonu gibi yöntemler kullanılır. Bulunan fonksiyonun noktalar arasındaki ilişkiyi ne kadar güçlü temsil ettiğini anlamak için fonksiyonun üreteceği sonuçlarla ilişki kurulmak istenen nokta kümesi karşılaştırılır. Karşılaştırma işlemi sonucunda belirtme katsayısı (R2), ortalama mutlak yüzde hata (MAPE) ve ortalama kareli hata (MSE) kullanılır.

En Küçük Kareler Yöntemi

Doğrusal ve polinomal regresyonun çözümünde kullanılan bu yönteme en küçük kareler yöntemi de denir. Denklem 3.1'de matris formu verilen X*A = Y bağıntısının çözümü ile A matrisi bulunarak $a_0 + a_1x + a_2x^2 + ... + a_px^p$ formundaki polinomunun katsayıları bulunur. Katsayıları bulunan polinom birinci dereceden bir polinom ise doğrusal regresyon, polinomun derecesi iki veya daha büyük ise doğrusal olmayan regresyon denkleminin katsayıları elde edilmiş olur.

$$\begin{pmatrix} n & \sum_{i=0}^{n} x_i & \sum_{i=0}^{n} x_i^2 & \cdots & \sum_{i=0}^{n} x_i^p \\ \sum_{i=0}^{n} x_i & \sum_{i=0}^{n} x_i^2 & \sum_{i=0}^{n} x_i^3 & \cdots & \sum_{i=0}^{n} x_i^{p+1} \\ \sum_{i=0}^{n} x_i^2 & \sum_{i=0}^{n} x_i^3 & \sum_{i=0}^{n} x_i^4 & \cdots & \sum_{i=0}^{n} x_i^{p+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sum_{i=0}^{n} x_i^p & \sum_{i=0}^{n} x_i^{p+1} & \sum_{i=0}^{n} x_i^{p+2} & \cdots & \sum_{i=0}^{n} x_i^{2p} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_p \end{pmatrix} = \begin{pmatrix} \sum_{i=0}^{n} y_i \\ \sum_{i=0}^{n} x_i y_i \\ \sum_{i=0}^{n} x_i^2 y_i \\ \vdots \\ \sum_{i=0}^{n} x_i^p y_i \end{pmatrix} \tag{3.1}$$

Python'da en küçük kareler yöntemi kullanılarak eğri uydurmak için numpy ve numpy.linalg kütüphanelerinden faydalanılabilir. Kütüphaneler aşağıda verildiği biçimde tanımlanır.

```
import numpy.linalg as l
import numpy as np
```

Girdi olarak kullanılacak olan X matrisi ve çıktı değerlerini temsil eden Y matrisi tanımlandıktan sonra X1 ve Y1 matrisleri Denklem 3.1'de verilen bağıntılar kullanılarak aşağıdaki gibi oluşturulur.

```
X=np.array([3, 5, 7, 9])
Y=np.array([45, 113, 213, 345])
X1=np.array([[len(X), sum(X), sum(X**2)],
[sum(X), sum(X**2), sum(X**3)],
[sum(X**2), sum(X**3), sum(X**4)]])
Y1=[sum(Y), sum(X*Y), sum(Y*X**2)]
```

$XA = Y$ bağıntısında tutarsızlık oluşmaması için bağıntı $X^T XA = X^T Y$ haline getirilir. Denklemin çözümü için her iki taraf da $X^T Y$ ifadesinin tersiyle çarpılarak polinomun katsayılarını içeren A matrisinin değerleri bulunur. İlgili hesaplamanın kaynak kodu aşağıda verilmiştir.

```
X2=np.matmul(np.transpose(X1), X1)
Xinv=l.inv(X2)
Y2=np.matmul(np.transpose(X1), Y1)
A=np.matmul(Xinv,Y2)
```

En küçük kareler yöntemi ile eğri denklemi bulunurken, gerçek noktalar ile bulunan denklem kullanılarak hesaplanan noktalar arasındaki kareli hatayı en aza indirecek katsayılar belirlenir. Bu yöntemin avantajı kareli hatayı en aza indirmenin yanında, çözüm esnasında veri boyutuna bakılmaksızın, boyutu polinomun derecesine bağlı bir matris üzerinde hesaplamalar yapılmasıdır.

Vandermonde Sistemi

Vandermonde sisteminde katsayı belirleme işlemi, girdi ve çıktı arasında bağıntı kurulurken üretilmek istenen polinomun derecesine ve veri sayısına bağlı olarak oluşturulan bir temel matris ile çıktı matrisine bağlı olarak yapılır. Denklem 3.2'de verilen temel matris, ilk sütun x^0'ı temsil ettiği için ilk sütun değerleri 1, ikinci sütun x^1'i temsil ettiği için ikinci sütun değerleri girdi matrisi ile aynı, üçüncü sütun x^2'yi temsil ettiği için üçüncü sütun değerleri girdi matrisinin karesi olacak biçimde kuvvetler birer artırılarak oluşturulur.

$$\begin{pmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^p \\ 1 & x_2 & x_2^2 & \cdots & x_2^p \\ 1 & x_3 & x_3^2 & \cdots & x_3^p \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^p \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_p \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{pmatrix} \tag{3.2}$$

Vandermonde sisteminin çözümü için en küçük kareler yönteminde verilen kaynak kodda X1 ve Y1 değişkenlerinin değerleri aşağıda verildiği biçimde atanmalıdır.

```
X1=np.column_stack((np.ones(len(X)),X,X**2))
Y1=Y
```

Newton İnterpolasyonu

Denklem 3.3'te verilen polinomunun katsayılarının bulunmasında Denklem 3.4'te verilen Newton temel matrisi kullanılır. Polinomun katsayıları Newton İnterpolasyonu ile bulunurken sadece

polinomun derecesinden bir fazla veri kullanıldığı için, Newton İnterpolasyonu tüm verileri gözden geçirerek bir sonuç üretmez. Bu yüzden Newton İnterpolasyonu hesaplama esnasında alınan örneklemi temsil eder. Örnek olarak alınacak matrisin büyüklüğü polinomun derecesine bağlı olduğu için çözümü kolay olur.

$$f(x) = a_0 + \sum_{i=1}^{p} a_i \prod_{j=1}^{i} x - x_{j-1} \tag{3.3}$$

$$\begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & x_1 - x_0 & 0 & \cdots & 0 \\ 1 & x_2 - x_0 & (x_2 - x_0)(x_2 - x_1) & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_p - x_0 & (x_p - x_0)(x_p - x_1) & \cdots & \prod_{i=0}^{p} (x_p - x_i) \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_p \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{pmatrix} \tag{3.4}$$

Newton İnterpolasyonu'nun çözümü yapılırken en küçük kareler yöntemi çözümünün kaynak kodunda X1 ve Y1 değişkenlerinin değerleri aşağıda verildiği gibi atanır.

```
X1=np.array([[1,0,0],
[1, X[1]-X[0], 0],
[1, X[2]-X[0], (X[2]-X[0])*(X[2]-X[1])]])
Y1=Y
```

A matrisi hesaplandıktan sonra, bulunan değerler aranan polinomun katsayısı değil, Denklem 3.3'te verilen polinomun katsayılarıdır. Bu polinom açık hale getirildiğinde katsayılar aşağıda verilen kaynak kodda yer aldığı gibi bulunur.

```
a0=A[0]-A[1]*X[0]+A[2]*X[0]*X[1]
a1=A[1]-A[2]*(X[0]+X[1])
a2=A[2]
```

SciPy ile Eğri Uydurma

Python ile eğri uydurma işlemleri için scipy.optimize kütüphanesinden curve_fit fonksiyonu kullanılmaktadır.

```
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
import numpy as np
```

curve_fit fonksiyonu ile katsayıları hesaplanacak eğrinin bir şablonunun tanımlanması gerekmektedir. Bu şablon aşağıdaki kaynak kodda yeni bir fonksiyon olarak tanımlanmıştır. Şablon fonksiyonun ilk parametresi girdi değerleri, sonraki değerleri ise polinomun katsayılarıdır.

```
def func(x, a0, a1, a2):
return a0+ a1 * x + a2*x**2
```

Eğri uydurma işleminde kullanılacak olan girdi ve çıktılar aşağıdaki kaynak kodda hesaplanmıştır.

```
X=np.array([3, 5, 7, 9])
Y=func(X, 3, 2, 4)
```

Şablon fonksiyon ile hesaplanan girdi ve çıktı değerleri curve_fit fonksiyonunda parametre olarak kullanılır. curve_fit fonksiyonu tanımlanan şablon fonksiyonda parametre olarak kullanılan katsayı değerlerini tahmin ederek popt değişkenine atamaktadır. curve_fit fonksiyonunun hesaplayacağı katsayıların belirlenecek alt ve üst limitler arasında olması isteniyorsa, fonksiyon curve_fit(func, X, Y, bounds=(0, [3., 3., 3])) şeklinde bounds parametresi ile kullanılır.

```
popt, pcov = curve\_fit(func, X, Y, bounds=(0, [3., 3., 3]))
```

Yapay Sinir Ağları ve Derin Öğrenme

Eğri uydurma yöntemleri bir girdi ve bir çıktı arasında ilişki kurmaya yararken yapay sinir ağları bir veya daha fazla girdi ile bir veya daha fazla çıktı arasında ilişki kurmada kullanılabilir. Yapay sinir ağlarının çalışma mekanizması sıralı biçimde katmanlara gelen değerlerin Denklem 3.5'te verilen f aktivasyon fonksiyonu ile doğrusal regresyona benzer bir fonksiyonun bilekesinde girdi olarak kullanılmasına dayanmaktadır. Elde edilen sonuç son katmanda ise çıktı değilse sıradaki katmana girdi olacaktır. Denklem 3.5'te görülen x değeri girdi w değeri ise ağırlıktır.

$$f\left(bias + \sum_{i=0}^{n} x_i w_i\right) \tag{3.5}$$

TensorFlow kullanılarak yapay sinir ağı oluşturmak için aşağıda verilen kaynak kodda yer alan kütüphaneler tanımlanır.

```
from keras import backend as K
from keras import layers as L
import keras as ke
import tensorflow.compat.v1 as tf
```

TensorFlow kütüphanesine ait dosyaları hafızaya yükleyerek aktifleştirmek için Session() fonksiyonu kullanılır. Bu fonksiyonla üretilen TensorFlow oturum nesnesi set_session fonksiyonu ile Keras'a aktarılır. Keras, TensroFlow kullanımını kolaylaştırmak için geliştirilen bir kütüphane olduğu için yaygın olarak kullanılmaktadır.

```
sess = tf.Session()
K.set_session(sess)
```

TensorFlow kütüphanesi aktifleştirildikten sonra modelin tasarımına başlanabilir. Bunun için, öncelikle girdi ve çıktı matrisleri placeholder fonksiyonu ile tanımlanır. Tanımlanan matrislerin sütun sayları girdi ve çıktı parametrelerinin sayısıdır. Bu değerler input_count ve output_count değişkenleriyle temsil edilmektedir.

```
input_count=3
output_count=1
X = tf.placeholder(tf.float32, [None, input_count]) ...
        #create variable objects
Y = tf.placeholder(tf.float32, [None, output_count])
```

Girdi ve çıktı değişkenleri tanımlandıktan sonra katman tasarımına geçilir. Keras ile oluşturulabilecek katmanlar tam bağlantılı katman (Dense, SimpleDense), temel katman nesnesi (Layer), kıvrım katmanı (Conv1D, Conv2D, Conv3D vb.), sadeleştirme katmanı (AveragePooling1D, MaxPool1D vb.), eleme katmanı (Dropout, GaussianDropout vb.), girdi katmanı (Input) ve devirli yapay

sinir ağı katmanı (SimpleRNN, AbstractRNN vb.) olarak sıralanabilir. Girdi katmanı Dense katmanı olarak tanımlanabilir. Fakat tanımlama esnasında girdi değişkeni olarak tanımlanan X, Dense fonksiyonuna parametre olarak verilir. Dense katmanı tam bağlantılı olduğu için sahip olduğu düğüm değerlerinin tamamını bir sonraki katmana iletir. Bundan dolayı yapay sinir ağı ve derin öğrenme tasarımında yaygın olarak kullanılmaktadır. Mevcut uygulamada dört Dense ve bir Dropout katmanı kullanılmıştır. Gizli katman olarak kullanılan katman sayısının üç veya daha fazla olması klasik yapay sinir ağıyla derin öğrenmeyi birbirinden ayıran en önemli noktalardan bir tanesi olduğundan, tasarlanan ağ bir derin öğrenme ağıdır. Ayrıca aşağıda verilen kaynak kodda yer aldığı üzere, gizli katman çıktıları üzerinde çeşitli işlemler yapılarak veya katman çıktıları farklı makina öğrenmesi algoritmalarıyla işlendikten sonra sıradaki katmana gönderilerek hibrit uygulamalar geliştirilebilir.

Yapılan tasarımda aktivasyon fonksiyonu olarak sigmoid fonksiyonu kullanılmıştır. Bunun dışında tanh, Softmax, ReLU, Leaky ReLU ve Swish fonksiyonlarıyla birlikte çeşitli fonksiyonlar aktivasyon fonksiyonu olarak kullanılabilir. Aktivasyon fonksiyonu girdi değerleri ve katmanlarda yapılan hesaplamaların sonuçlarının değer aralığıyla davranışlarına göre belirlenmelidir. Son olarak da Dropout oranı 0.5 olarak belirlenmiş ve tasarım sonlandırılmıştır. Dropout ağın ezberlemesine (over-fitting) önlem olarak kullanıldığından belirlenecek oran da ezberleme eğilimine göre artırılıp azaltılabilir. Bu oranyaygın olarak 0.05 ile 0.5 arasında belirlense de daha yüksek ve daha küçük değerler atanması da muhtemeldir.

```
input_layer = L.Dense(10, activation='sigmoid')(X)
hidden_layer1 = L.Dense(20, activation='sigmoid')(input_layer**2)
hidden_layer2 = L.Dense(20, activation='sigmoid')(input_layer)
hidden _layer3 = L.Dense(20, activation='sigmoid')...
        ((hidden _layer1+ hidden _layer2)/2)
hidden _layer4 = Dropout(0.5)(hidden _layer3)
output_layer = L.Dense(output_count, activation='sigmoid') ...
        (hidden _layer4)
```

Ağın tasarımı tamamlandıktan sonra eğitim (training) aşamasına geçilir. Eğitim Denklem 3.5'te görülen ağırlık (wi) ve bias değerlerinin hesaplanması işlemidir. Ağın en iyi performansı vermesini sağlayacak ağırlık ve bias değerlerini bulmak ise bir optimizasyon işlemi sonucunda mümkün olacaktır. Bundan dolayı, eğitim işlemi TensorFlow kütüphanesinde tanımlı olan Adam, SGD, Adamax veya Adadelta gibi optimizasyon algoritmaları kullanılmaktadır. Optimizasyon işleminde amaç fonksiyonu olarak tanımlanan bir kayıp (loss) fonksiyonunun değeri en aza indirilmeye çalışılmaktadır. Kayıp fonksiyonu olarak MeanSquaredError, MeanAbsoluteError, CrossEntropy ve daha birçok tanımlı fonksiyon kullanılabilmekle birlikte özel kayıp fonksiyonu tasarlamak da mümkündür. Kayıp fonksyonu seçerken ya da tasarlarken dikkat edilmesi gereken en önemli husus fonksiyonun optimizasyon algoritması ve aktivasyon fonksiyonu çıktılarına uyumlu olmasıdır. Mevcut uygulamada AdamOptimizer kullanılmış ve öğrenme oranı (learning rate) 0.01 olarak belirlenmiştir. Öğrenme oranının olması gerekenden küçük olması yetersiz öğrenmeye, büyük olması ise ağın performansının beklenen seviyenin altında olmasına neden olur. Öğrenme işlemi sonucunda

```
abs_diff=tf.abs(Y - output_layer)
loss = tf.reduce_sum(abs_diff**2)
train_step = tf.train.AdamOptimizer(0.01).minimize(loss)
```

Kayıp fonksiyonu da belirlendikten sonra eğitim aşamasına geçilebilir. Eğitim aşamasında tekrar sayısı ağın optimal öğrenmesi açısından önemli bir parametredir. Tekrar sayısı belirlendikten sonra

global_variables_initializer ile modelin optimizasyonunda ihtiyaç duyulacak olan global değişkenleri tanımlayacak olan bir Factory nesnesi üretilir.

```
epoc_count=2000
init = tf.global_variables_initializer()
```

Eğitim sürecine başlanırken önce global değişkenler hafızaya alınır. Ardından tekrar sayısı kadar çalışacak bir döngünün içinde Çok tekrar ezberlemeye (over-fitting), az sayıdaki tekrar ise yetersiz öğrenmeye (under-fitting) neden olabilir. Öğrenmenin durumunu anlamak için her bir eğitim iterasyonu esnasında eğitim seti ile elde edilen kayıp fonksiyonu değeri ile test seti kullanılarak elde edilen kayıp değerleri karşılaştırılır. Şekil 4'te görülen kayıp değerlerinden test eğitim eğrisi ile test seti eğrisi birlikte azalırken ağ yetersiz öğrenme durumundadır. Eğitim seti eğrisi azalırken test seti eğrisi artma eğilimi gösteriyorsa ezberleme başlamış demektir. Şekil4'teki grafiğe göre 2000 tekrarın üstü ezberlemeye neden olduğundan optimal tekrar sayısı 2000 civarında aranmalıdır.

```
with sess.as_default() as sess:
init.run()
for i in range(epoc_count):
train_step.run(feed_dict={X: train_X,
        Y: train_Y})
mse = loss.eval(feed_dict={X: train_X, Y: train_Y})
mse2 = loss.eval(feed_dict={X: test_X, Y: test_Y})
yPred = sess.run(output_layer, feed_dict={X: test_X})
```

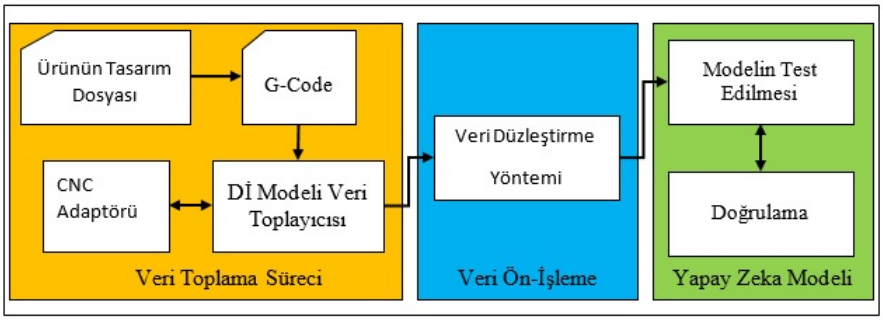
3.3 Dijital İkiz Uygulaması

Dijital İkiz (Dİ) olarak adlandırılan gerçek üretim sisteminin dijital kopyası, gerçek bir üretim ortamında meydana gelen tüm durumları analiz etme yeteneğine sahiptir ve sistemdeki tüm işlemlerin sonuçlarını tahmin etmeye çalışmaktadır. Bahsedilen modelleme seviyesine ulaşmak için, Dİ modeli, bir sistemin dinamiklerini taklit eden alt modellerden oluşur ve bu dinamiklerini modellemek, veri bilimi perspektifinde, diferansiyel modelleme, eğri uydurma, doğrusal/doğrusal olmayan/polinom regresyon, makine öğrenme yöntemleri ve yapay zekâ olmak üzere birkaç yolla mümkün olmaktadır. Dijital İkiz yaklaşımı ile üretim sistemlerinin tüm bileşenlerinin modellenmesinde tabi ki mümkündür. Üretim sistemlerinin temel bileşenlerinden biri tezgâhlardır ve tezgâhların çoğunda parçaları işlemek için kartezyen eksenler bulunur. Bu bölümde, kartezyen eksenli CNC tezgâhlarının Dİ modellerini oluşturmak için standartlaştırılmış bir yapıyı önerilmiş ve Dİ modelinin, tezgâhta gerçekleştirilecek görevin işlem süresini ve enerji tüketimini tahmin etmesi amaçlanmıştır.

Dİ uygulamasına başlamadan önce literatürde yapılan uygulamalardan bahsedilmiş ve bu çalışmanın farkının ortaya konulması uygun görülmüştür. Bilimsel literatürdeki birçok çalışma, CNC tezgâhların işleme başlamadan önce atanan işlerin sonuçlarını ve performans göstergelerini tahmin eden bir Dİ modeli önermektedirler (Yang, 2021; Roy, 2020; Liu, 2021). Bu çalışmalarda, Dİ modelleri yapay zekâ (AI) veya makine öğrenmesi modelleri kullanılarak geliştirilmiştir (Yang, 2021). Tezgâhta arıza riskini analiz etmek için Gri İlişkiler Yöntemi (Gray Relations) kullanmış ve derin öğrenme yönteminin (Deep Learning) önerilen Dİ modeline daha fazla katkı sağlayacağı vurgulanmıştır. Diğer bir çalışmada ise (Roy, 2020), makine parçalarının kullanılabilirlik düzeyi belirlemek ve farklı sınıflara atayabilmek için Destek Vektör Makinesi (Support Vector Machine) algoritması kullanılmıştır. Başka bir çalışma ise CNC tezgâhının takım kafasına ivmeölçer ve

jiroskop yerleştirerek Dİ ile kalite kontrol sürecini incelemiştir (Liu, 2021). Sensör verileri, Kayan Pencere algoritması (Sliding Window Algorithm) ile katı model verileriyle eşleştirilir ve eşleşen veriler AI tarafından analiz edilir. Yapay zekânın kullanıldığı Dİ çalışmalarında gözlemlenen ortak nokta, üretim alt bileşenlerinin Dİ modeli oluşturularak, öncelikle bileşenin çalışma mekanizmasını çıkarmak ve gerekli sonuçları üretmeye çalışmaktır.

Önerilen çalışmada ise, kartezyen eksenli CNC tezgâhlarında lineer hareket komutlarının (G0, G1) operasyon süresini tahmin eden bir Dİ modeli geliştirilmektedir. Dİ modeli, herhangi bir CNC adaptörünü (sürücü kartı) modellemeye amaçlar. Uygulamanın ilk aşaması veri toplamadır; ikinci aşamada, Yapay zekâ modelini eğitmek için veri ön işleme yapılmaktadır ve sonrasında yapay zekâ tahmin modelinin geliştirilmesi gelmektedir. Son olarak, Şekil 2'de görüldüğü gibi yapay zekâ modelinin bir doğrulaması yapılmıştır. Ancak bu uygulamanın diğer çalışmalardan farkı, üretim sistemleri için tasarlanmış bir Dİ modelin pratik uygulamalarını içermesidir.

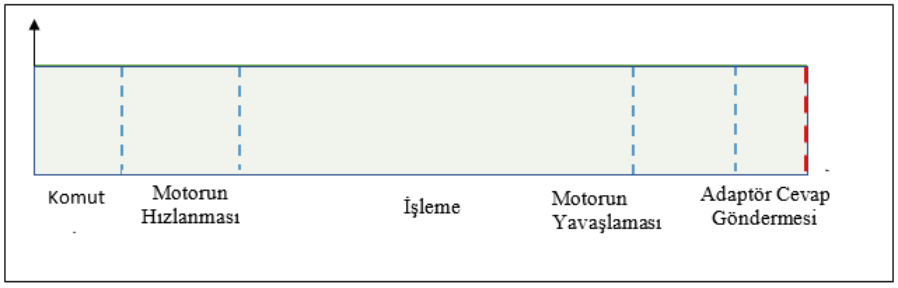


Şekil 3.2: Dijital ikiz modelinin yapısı

3.3.1 Önerilen Metot

Tezgâh üzerinde üretilmesi planlanan ürünün G-code dosyası Dİ modeline yüklenir. Dİ modeli G-Code dosyasını satır satır CNC adaptöre gönderir ve işlemin süresini ölçer. Bu işlem süresi beş bölümden oluşmaktadır; birincisi G-Kodu gönderme süresi, ikincisi motorun hızlanma süresi, üçüncüsü motor hareket süresi, dördüncüsü motorun yavaşlama süresi ve sonuncu süre ise Şekil 3'te görüldüğü gibi CNC adaptörün sürecin sonu hakkında Dİ modeline cevap verme süresidir. İkinci prosedür, veri ön işlemedir. Veri iletişimindeki zaman gecikmeleri nedeniyle zaman ölçümlerine yumuşatma işlemi uygulanır. Bir yumuşatma yöntemi olarak hareketli ortalama yöntemi uygulanabilir. Veri ön işlemeden sonra hem doğrusal/güç regresyonu hem de çok katmanlı algılayıcı kullanılarak Dİ modeli oluşturulmaya çalışılır. Çok katmanlı algılayıcıda tek gizli katman tercih edilir. Öğrenme algoritması olarak geri yayılım seçilmiştir ve aktivasyon fonksiyonu olarak hiperbolik tanjant fonksiyonu kullanılmıştır.

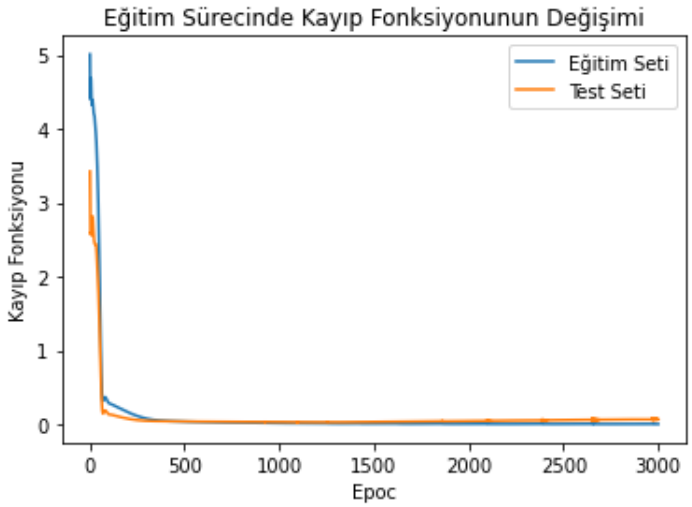
GRBL projesi ile uyumlu bir CNC adaptörünün dijital ikizini oluşturmak için literatürde önerilen Dİ model oluşturma prosedürleri uygulanmıştır. Doğrusal hareketin ilk adım işleme süresinde, farklı hız ve hareket mesafesi kombinasyonları için komutlar ölçülmüştür. İkinci adımda, süre verileri Şekil 3'te görüldüğü gibi hareketli ortalama ile düzeltilmiştir. Son adımda ise 3 Dİ modeli geliştirilmiştir. İlk model, 0.99 R2 ve 6.83x10-5 ortalama mutlak yüzde hatası (MAPE) ile sabit çalışma hızında G0 komutu veya G1 komutu ile muhteşem performans sergileyen doğrusal regresyondur. Ancak hız parametreleri dinamik hale geldiğinde doğrusal model iyi performans göstermemektedir. Bu durumda bir güç regresyon modeli kurulmaya çalışılmıştır. Güç regresyon modeli 0.98 R2 ile



Şekil 3.3: Bir G-Code komutunun icra çevrimi

kurulmuş ve işlem süresini %10.62 ortalama mutlak yüzde hatası ile tahmin etmiştir. Güç regresyon modelinin performansı lineer regresyondan daha iyi olsa da, genel olarak dijital ikiz hedefine ulaşmak için hata seviyesinin düşürülmesi gerekmektedir.

Son olarak, TensorFlow aracılığıyla Adams Optimizer ile çok katmanlı bir algılayıcı modeli geliştirilmiştir. Dİ modeli küçük veri setleri ile geliştirilmeye ihtiyaç duyulduğu için ile öğrenme oranı (learning rate) 0.01 olarak ayarlanmıştır. Model 10 nöronlu tam bağlantılı bir girdi katmanı, 20'şer nöronlu üç tam bağlantılı gizli katman, bir Dropout gizli katmanı ve bir çıktı katmanından oluşmaktadır. Hız, yer değiştirme miktarı ve toplam yer değiştirmenin hıza oranı giriş katmanına bağlıdır. Gizli katman için sigmoid aktivasyon fonksiyonu, çıkış katmanı için softplus aktivasyon fonksiyonu kullanılmaktadır. Kayıp fonksiyonu olarak hataların karelerinin toplamı (Sum of Squared Error), ve optimizasyon metriği olarak ortalama mutlak hata (Mean Absolute Error) kullanılmaktadır. Kayıp fonksiyonunun değeri Şekil 4'te görüldüğü gibi eğitim verilerinde 4.8061285'ten 0.0016557573'ye, test verilerinde 3.2398317'den 0.0052765277'ye düşürülmüştür. Sonuç olarak model 1 R2 değeri ve 0.03894 MAPE ile kurulmuştur.



Şekil 3.4: Kayıp Fonksiyonu

3.3.2 Sonuçlar

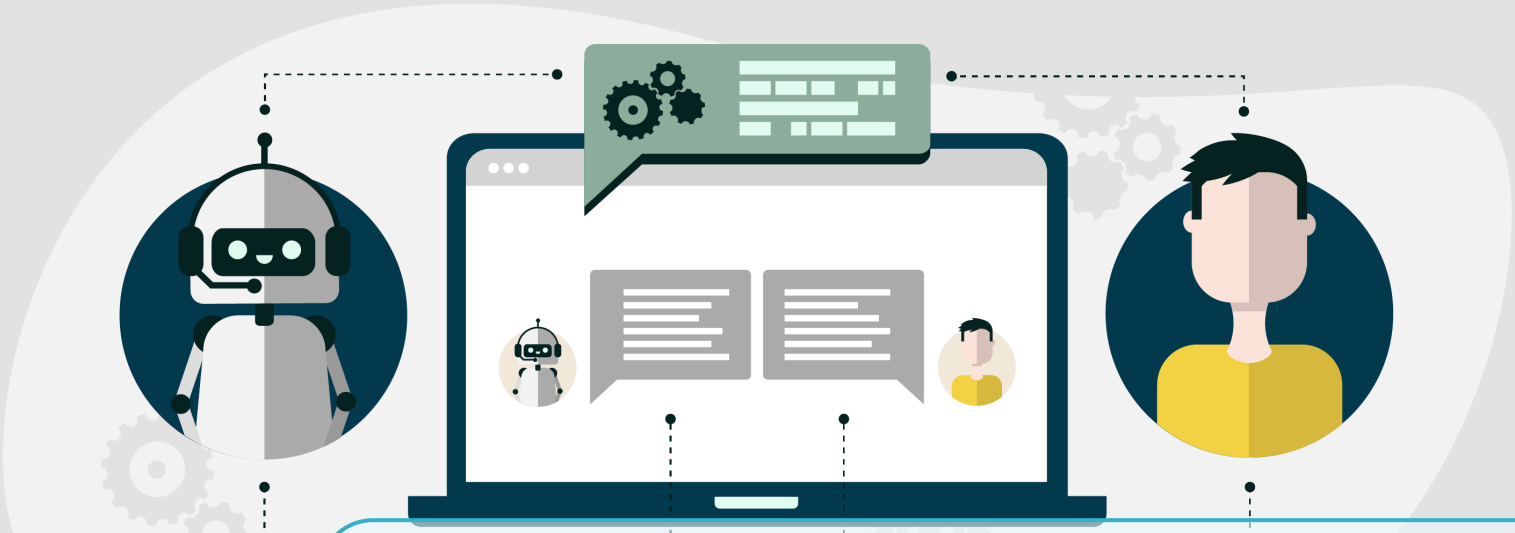
Dinamik ve esnek bir üretim sistemine ulaşmak için sanayi devrimi sırasında Dİ yaklaşımı kullanılmaya başlanmıştır. Ancak gerek literatürde gerekse imalat sektörlerinde yeterli teorik ve pratik uygulama bulunmamaktadır. Bu noktada, bu çalışma, imalat tarafında kullanılan bir CNC tezgahının Kartezyen ekseninin dijital ikizinin bir ön çalışmasını sağlamayı amaçlamaktadır. Öncelikle veri ön işleme aşamasında CNC tezgahların G Kodundan elde edilen veriler hareketli ortalama yöntemleri kullanılarak düzgünleştirilir. Daha sonra işlem süreleri çeşitli hareket hızları ve hareket miktarları ele alınarak tahmin edilmeye çalışılır. Görevin tamamlanma süresini bulmak için iki yöntem kullanılır 1) Regresyon yöntemi 2) Çok Katmanlı Yapay Sinir Ağı. Sabit hız koşullarında çalışma sürelerini hesaplamak için regresyon yönteminin yeterli olduğu çok açıktır. Model farklı hız değerleri altında çalışacak şekilde genişletildiğinde ise, regresyonun mantıklı bir sonuç üretememektedir. Bu nedenle, bir CNC tezgahının doğrusal hareket süresini tahmin etmek için çok katmanlı algılayıcı kullanılarak orijinal bir model geliştirilmesi tavsiye edilmiştir.

3.4 Kaynaklar

R. B. Roy et al., “Digital twin: current scenario and a case study on a manufacturing process,” Int. J. Adv. Manuf. Technol., vol. 107, no. 9–10, pp. 3691–3714, 2020.

S. Liu, S. Lu, J. Li, X. Sun, Y. Lu, and J. Bao, “Machining process-oriented monitoring method based on digital twin via augmented reality,” Int. J. Adv. Manuf. Technol., vol. 113, no. 11–12, pp. 3491–3508, 2021.

Y. Wang, W. Ren, Y. Li, and C. Zhang, “Complex product manufacturing and operation and maintenance integration based on digital twin,” Int. J. Adv. Manuf. Technol., 2021.



4. Dijital İkiz Yapay Zeka İlişkisi

Dijital İkiz Yapay Zeka İlişkisi

Zerrin AYVAZ REİS*

***İstanbul Üniversitesi-Cerrahpaşa, Hasan Ali Yücel Eğitim Fakültesi-BÖTE (Bilgisayar ve Öğretim Teknolojileri Eğitimi Bölümü**

4.1 Giriş

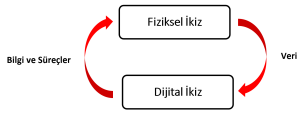
Yapay zeka (AI), nesnelerin interneti (IoT) ve Blok zincir gibi birçok teknolojiyle entegre edilebilen dijital ikiz üzerine çalışmalar hızla devam etmektedir. Gartner'in 2019 yılında seçtiği ilk 10 stratejik trend arasında yapay zeka, blok zincir, kuantum programlama gibi teknolojilerin arasında dijital ikiz de yer almaktadır. Yapay zeka, detayını anlamamız için verileri daha başarılı bir şekilde parçalayabilmekte ve bu süreçte nesnelerin interneti de önemli bir rol oynamaktadır. Makine Öğrenimi, yapay zekanın bir alt dalıdır ve akıllı sensörlerin oluşturduğu verilerdeki kalıpları ve anormallikleri tespit etme konusunda büyük bir potansiyele sahiptir. Şu anda, makine öğrenimi tabanlı analitik, bir dizi yapay zeka yeteneği sunan büyük nesnelerin interneti ve bulut servisi sağlayıcılar tarafından sağlanmaktadır (Rabah, 2018).

Son yıllarda hayatımızın çoğu alanında dijital dönüşüme tanık olduk ve bu dönüşüm toplumlarımızda da değişiklikler yarattı. Veri veya işlemleri takip etme, işbirliği yapma ve iletişim kurma şeklimiz dijital teknoloji ile değiştirildi ve sürekli geliştirilmektedir. Bilindiği gibi yapay zeka, makinelerin deneyimden öğrenmesini, yeni girdilere uyum sağlamasını ve insan benzeri görevleri gerçekleştirmesini mümkün kılmaktadır. Bugün satranç oynayan bilgisayarlardan kendi kendine giden arabalara kadar pek çok yapay zeka örneği derin öğrenme ve doğal dil işlemeye dayanmaktadır. Yapay zeka, bulut bilişim ve büyük veri kavramlarını bir çatı altında toplayan dijital ikiz teknolojisi, aslında fiziksel dünyadaki bir cihazın, kişinin veya tesisin dijital dünyadaki simülasyonu olarak da ifade edilmektedir. Dijital ikiz, gelişmiş veri analitiği ve nesnelerin interneti bağlantısı aracılığıyla kolaylaştırılan Endüstri 4.0 devriminin ön saflarında yer almaktadır. Dünyada dijital ikizleri kullanıp geliştiren ve endüstri 4.0 adı verilen bu 4. Sanayi Devrimine yön veren şirketler ise IBM, Microsoft,

Tesla ve General Electric gibi endüstri devi şirketlerdir.

4.2 Dijital İkiz

Alan yazın incelendiğinde Dijital İkizin ilk olarak Michigan Üniversitesi'nde 2003 yılında Michael Grieves tarafından Ürün Yaşam Döngüsü Yönetimi (Product Lifecycle Management-PLM) merkezinin oluşturulması için yapılan ilk yönetici sunumunda gündeme geldiği anlaşılmaktadır. Adı 'PLM için Kavramsal İdeal' olmasına rağmen gerçek alan, sanal alan, gerçek alandan sanal alana veri akışı için bağlantı, sanal alandan gerçek uzaya bilgi akışı için bağlantı ve sanal alt uzaylar olmak üzere bir dijital ikizin tüm unsurlarına sahip bir model olarak sunulmuştur (Grieves, 2006). Modeli oluştururken varsayılan önerme, her sistemin iki sistemden oluşması temeline dayalı olan, her zaman var olan fiziksel sistem ve fiziksel sistem hakkındaki tüm bilgileri içeren yeni bir sanal sistem olarak yapılmıştır. Bu, gerçek uzayda var olanlar ile sanal uzayda var olanlar arasında bir aynalama veya ikizleme sistemleri olduğu anlamına gelmektedir (Grieves, 2016). Bu sanal mükemmellik uzun yıllar üzerinde yapılan yenilikçi çalışmalar sonucunda Bilgi Yansıtma Modeli (Information Mirroring Model) adını almıştır. Grieves ile bu çalışmaları birlikte yürüten John Vickers tarafından modelin Dijital İkiz' olarak tanımlanmasıyla model artık bu isimle anılmaya başlamıştır (Grieves 2011). Grieves (2015) daha sonra dijital ikizi; fiziksel bir ürün, bu ürünün sanal bir temsili ve fizikselden sanala veri ve sanaldan fiziksele bilgi ve süreçler besleyen (Bakınız Şekil 4.1) çift yönlü bir döngü olarak veri bağlantılarından oluşan üç bileşenli olarak tanımlayarak önceki tanımı genişletmiştir. Burada söz konusu olan fiziksel ikiz, gerçek dünyadaki fiziksel sistemlerin sistemini veya sanal bir dijital ikiz tarafından çoğaltılan ürünü tanımlayan göreceli bir terimdir (Jones ve arkadaşları, 2020). Başka bir şekilde ifade etmek gerekirse dijital ikiz, fiziksel bir nesne hakkındaki bilgilerin nesnenin kendisinden ayrılabileceği ve daha sonra o nesneyi aynalayabileceği bir kavramdır.



Şekil 4.1: Dijital ikiz modeli

Dijital ikiz, çeşitli nedenlerle fiziksel ikizin ayrılmaz bir parçasıdır. İlk olarak, dikkate almamız gereken iki şey vardır.

- Özünde sistemi kontrol etmek isteyen insanlarla Dijital İkiz aracılığıyla iletişim kuran bir fiziksel ikiz ve
- Genellikle makineden makineye (M2M) iletişim olarak adlandırılan, diğer sistemlerle iletişim kuran bir fiziksel ikiz.

Bu bilgilerle Grieves (2016) dijital ikizi, mikro atom seviyesinden makro geometrik seviyeye kadar potansiyel veya gerçek fiziksel olarak üretilmiş bir ürünü tam olarak tanımlayan bir dizi sanal bilgi yapısı olarak tanımlamıştır.

Haag ve Reiner. (2018). dijital ikizi, konuşlandırıldığı ortamın gerçek davranışını simüle edebilen bir dizi gerçekçi model olarak tanımlamaktadır. Dijital ikiz fiziksel ikizindeki sensörlerden gelen veriler aracılığıyla gerçek hayattaki nesnenin özelliklerini, durumunu ve davranışını içermektedir.

Puri (2017) ise dijital ikizi öngörücü ikiz olarak da adlandırmaktadır. Bu ikiz, fiziksel sistemdeki arızaları önceden tahmin ederek ve geçmiş verilerle birlikte yorumlayarak sistemlerin karar almalarına yarar sağlamaktadır (Aynacı, 2020).

Terminoloji zamanla değişse de, Dijital İkiz modelinin temel konsepti, 2003'deki başlangıcından bu yana oldukça sabit kalmıştır. Fiziksel bir sistem hakkında dijital bir bilgi yapısının kendi başına bir varlık olarak oluşturulabileceği fikrine dayanmaktadır. Bu dijital bilgi, fiziksel sistemin kendisinde gömülü olan bilginin bir 'ikizi' ve sistemin tüm yaşam döngüsü boyunca bu fiziksel sistemle bağlantılı olmaktadır. Aynacı (2020)'nın aktardığı gibi dijital ikiz teknolojisinin temelinde olan simülasyon teknolojisi 1970 yılında NASA'daki çalışmalarda görülmüştür. NASA'nın Apollo 13 uzay aracını Ay'a göndermesinden kısa bir süre sonra ortaya çıkan teknik arızalar sebebiyle aracı ve içindeki görevlileri dünyaya geri getirmek amacıyla kurtarma operasyonu başlatma zorunluluğu doğmuştur. Bu operasyonunun en önemli noktası Apollo 13'ün bileşenleriyle aynı verilere sahip olan bir ayna model sayesinde mühendislerin geri dönüş için olası çözümleri modellemeleri ve test etmeleri mümkün olmuştur. Gerçekleştirilen simülasyonlar sonrasında uzay aracı ve astronotların geri dönüşü sağlanmıştır. NASA, bu olaydan sonra sistemlerini sürekli izleyebileceği ve daha hassas veri güncellemesi yapabileceği dijital modeller kullanmaya başlamıştır (Aynacı, 2020). NASA'nın uyguladığı bu teknik ile dijital ikiz teknolojisinin temelinde yatan teknik aynı anlaşılmaktadır. Fiziksel bir nesnenin dijital modeli oluşturulmuş ve fiziksel nesnenin durumunu izlemek, sorunları teşhis etmek ve olası çözümlerini test etmeyi sağlayan bir 'ikiz' yaratılmıştır (Houten, 2018). Bu sayede NASA uzay araştırmalarında, bugünün dijital ikizinin öncüsü olan eşleştirme teknolojisini ilk kullanan olmuştur. Bugün ise yeni modeller, yol haritaları, yeni araçlar ve yeni sistemler geliştirmek için dijital ikizler kullanılmaktadır (Marr, 2017).

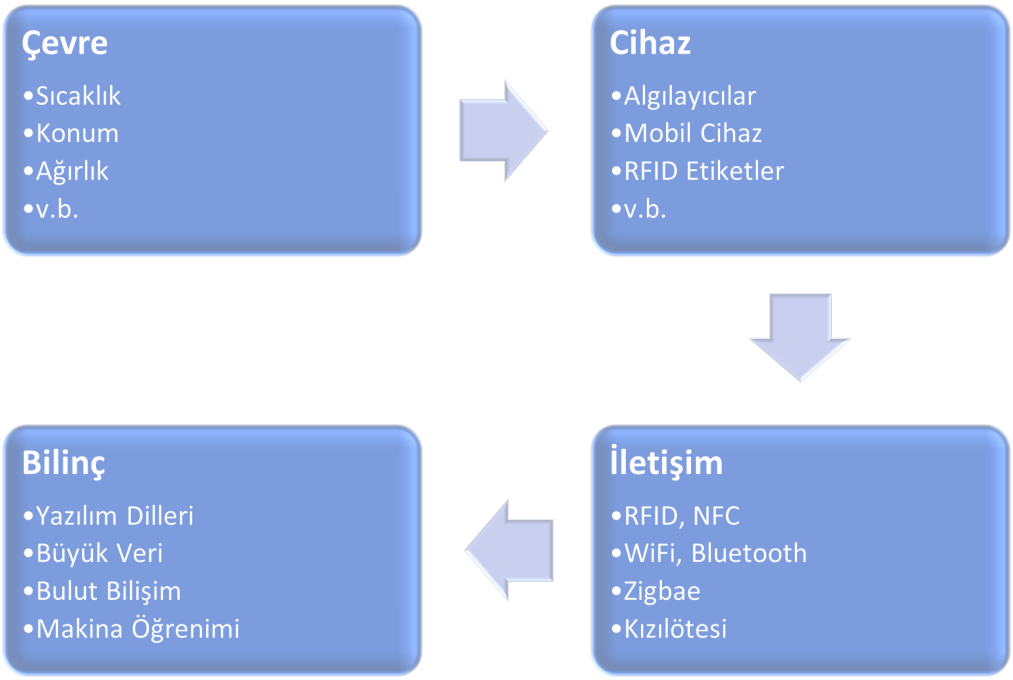
4.3 Nesnelerin İnterneti ve Dijital İkiz İlişkisi

Lee ve Lee (2015) nesnelerin internetini 'Her Şeyin İnterneti veya Endüstriyel İnternet olarak da adlandırılan Nesnelerin İnterneti (IoT), birbirleriyle etkileşime girebilen küresel bir makine ve cihaz ağı olarak tasarlanan yeni bir teknoloji paradigması' olarak tanımlanmıştır. IoT, geleceğin teknolojisinin en önemli alanlarından biri olarak kabul ediliyor ve çok çeşitli endüstrilerden büyük ilgi görüyor. İşletmeler için IoT'nin gerçek değeri, bağlı cihazlar birbirleriyle iletişim kurabildiğinde ve satıcı tarafından yönetilen envanter sistemleri, müşteri destek sistemleri, iş zekası uygulamaları ve iş analitiği ile entegre olabildiğinde tam olarak gerçekleştirilebilir.

Gubbi ve ark. (2013) tarafından, Kablosuz Sensör Ağı (Wireless Sensor Network-WSN) teknolojilerinin sağladığı her yerde bulunan algılama araçlarının, modern günlük yaşamın birçok alanıyla keşismesi sayesinde hassas ekolojilerden ve doğal kaynaklardan kentsel ortamlara kadar çevresel göstergeleri ölçme ve anlama yeteneği sunduğunu ve iletişim amaçlı hareketli bir ağda bu cihazların yaygınlaşmasıyla bilgilerin platformlar arasında paylaşılmasının Nesnelerin İnterneti'ni (IoT) yarattığı ifade edilmiştir.

Gökrem ve Bozuklu (2016) nesnelerin internetini, çevremizdeki fiziksel olayları kontrol etmemizi ve takip ederek analiz etmemizi sağlayan cihaz, yazılım ve erişim hizmetlerini kapsayan bir iletişim ağı olarak tanımlamıştır. Akıllı endüstri uygulamalarının nesnelerin internetinin algılama yetenekleri ile çeşitli endüstriyel işlemleri otomatikleştirmek amacıyla sanayi altyapısını birleştirdiği ifade edilmiştir. IEEE P2423 standardının henüz tamamlanmamış olması ve literatürde çok sayıda farklı nesnelerin interneti mimari çerçevesi olması sebebiyle kendi tanımladıkları nesnelerin interneti organizmasının katmanları Şekil 4.2'de görülmektedir.

Nesnelerin internetinin çekirdek katmanı çevredir. Çevre; her türlü doğal çevre fiziksel büyüklüklerin bulunduğu ortamı ifade eder, bu katmanda her türlü ölçülebilir büyüklük ham haldedir. Cihaz katmanında bu ham veriler algılanarak analog veya sayısal sinyallere dönüştürülür, bu verilerin işlenmesi için iletilmesi gerekmektedir. İnsan-makine, makine-makine iletişimi için gerekli olan



Şekil 4.2: Nesnelerin İnterneti Katmanları (Kaynak: Gökrem ve Bozuklu, 2016)

taşıyıcılar ile kablolu ve kablosuz iletişim altyapısı ve iletişim protokolleri iletişim katmanında yer alır. Daha sonra bu veriler ilgili iletişim protokolleriyle bilinç olarak adlandırılan veri işleme merkezine gönderilirler. Burada küçük çaptaki veri işleme işlemleri gömülü sistemler ile gerçekleşirken büyük uygulamalarda ise bu veriler depolanmak üzere bulut bilişim sistemlerine iletilir. Burada depolanan veriler artan yığınlar halinde büyük veriyi oluştururlar. Verimliliğin arttırılabilmesi için bu büyük miktardaki verinin analiz edilmesi gerekmektedir ve bu da makine öğrenimi yöntemleri kullanılarak gerçekleştirilir (Gökrem & Bozuklu, 2016).

Ercan ve Kutay (2016) nesnelerin internetinin temellerinin 1990'ların başında Weiser tarafından önerilen 'Her Zaman Her Yerde Hesaplama (Ubiquitious Computing)' kavramına dayandığını ifade etmektedir. Nesnelerin interneti terimini ilk defa kullanan kişinin ise MIT RFID araştırma grubunda yer alan Ashton olduğunu belirtmektedir. Üretim alanlarında RFID etiketi taşıyan malzemelerden RFID okuyucularla otomatik olarak alınan bilgilerin veri ağı ortamında saklanırken başka bir akıllı sistemi harekete geçirecek farklı bir süreci başlatabileceklerini de belirtmektedir (Ercan ve Kutay, 2016).

Dijital ikizler, nesnelerin internetiyle fiziksel ikizin operasyonel ortamda nasıl davrandığını ve performans gösterdiğini anlamak için gereken verileri getirir. Ayrıca, nesnelerin internetiyle dijital ikizlerin birleşimi, fiziksel sistem ve operasyonel süreçlerin önleyici bakım ve analitik/AI (yapay zeka) tabanlı optimizasyonunu iyileştirebilir.

4.4 Makina Öğrenmesi ve Dijital İkiz İlişkisi

Basit uygulamalar için dijital ikiz teknolojisi, makine öğrenmesini kullanmak zorunda kalmadan değer sunmaktadır. Basit uygulamalar, sınırlı sayıda değişken ve girdiler ile çıktılar arasında kolayca keşfedilebilir doğrusal bir ilişki ile karakterize edilir. Bununla birlikte, birden fazla veri

akışıyla mücadele eden çoğu gerçek dünya sistemi, verileri anlamlandırmak için makine öğrenimi ve analitikten faydalanmaktadır. Bu bağlamda makine öğrenmesi, daha sonra çeşitli şekillerde yararlanılabilecek kalıpları ortaya çıkarmak için veri akışına uygulanan herhangi bir algoritmayı ifade eder. Örneğin, makine öğrenmesi karmaşık analitik görevleri otomatikleştirip, verileri gerçek zamanlı olarak değerlendirebilmekte, minimum denetim ihtiyacıyla davranışı ayarlayıp, istenen sonuçların olasılığını artırabilmektedir. Makine öğrenmesi ile sistem maliyetten tasarrufa katkı verecek eyleme geçirilebilir, öngörüler üretmeye katkıda bulunabilir. Akıllı binalar, dijital ikizdeki makine öğrenimi yeteneklerinden yararlanacak mükemmel bir uygulama örneğidir (Cityzenith, 2018; Scholten,2017). Dijital ikiz için de makine öğrenimi kullanımları: simülasyon tabanlı, kontrollü deney test ortamında operatör/kullanıcı tercihlerinin ve sinir ağı kullanılarak önceliklerinin denetimli öğrenmesi, sanal ve gerçek dünya ortamlarında kümeleme teknikleri kullanılarak nesnelerin ve kalıpların denetimsiz öğrenmesi (Madni ve ark., 2018; Madni ve ark., 2019) belirsiz, kısmen gözlemlenebilir operasyonel ortamlarda sistem ve çevre durumlarının pekiştirici öğrenmesini içermektedir (Madni, 2018; Madni, Madni & Sievers, 2018).

4.5 Dijital İkiz Nasıl Yapılır?

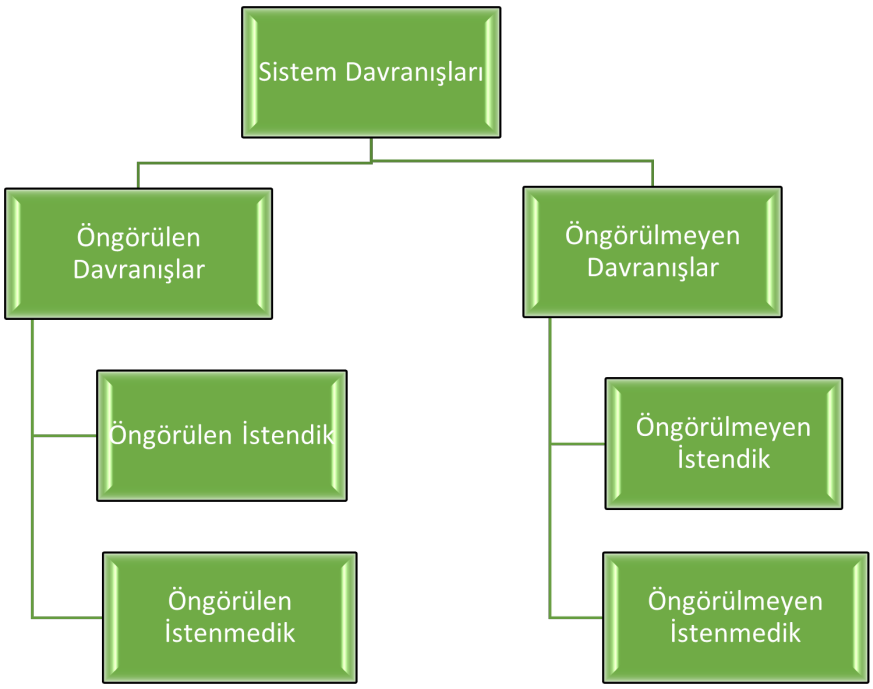
Fiziksel bir nesneyle çalışabilmekle ilgili sorunlardan biri, davranışına yönelik araştırma aralığının hem pahalı hem de zaman alıcı olmasıdır. Bunun için öncelikle üretilmek istenen nesnenin fiziksel olarak en az bir tane üretilmesi gereklidir. Daha sonra bu nesnenin maruz kalacağı, gerçek güçlerden etkilendiği fiziksel bir ortam oluşturmak gereklidir. Bu, nesneye uygulanacak güçleri ve nesnenin bunlara verdiği tepkileri, aralarındaki ilişkili seviyelerini araştırmakta sınırlı olduğumuz anlamına gelmektedir. Çoğu zaman, uygulanan kuvvetler nesnenin tahrip olmasına neden olur ve maliyeti önemli ölçüde artırır. Öte yandan öngörülemediği için fiziksel bir testin kapsamadığı herhangi bir koşulu ilk gördüğümüz zaman fiziksel nesnenin fiili kullanımda olduğu zaman olabilir. Bu durumlar kullanıcılarına yaralanmalarına veya ölümüne neden olabilecek arızalarla sonuçlanan birçok öngörülemeyen koşul veya acil davranış olacağı anlamına gelmektedir.

Grieves (2017) ancak yirminci yüzyılın son yarısında, bilgiyi fiziksel bir nesneden ayırabilmek ve dijital ikiz oluşturmanın mümkün olduğunu ifade etmiştir. Bu dijital ikiz, bir Bilgisayar Destekli Tasarım (Computer Aided Design-CAD) olarak nispeten nadir kullanılmış, yıllar içinde giderek daha zengin ve sağlam temele oturmuştur. Başlangıçta dijital ikiz sadece tanımlayıcıyken, son yıllarda eyleme geçirilebilir hale gelmiştir. Eyleme geçirilebilir olan şey, CAD nesnesinin artık zamandan bağımsız, boş uzayda asılı duran üç boyutlu bir nesne olmadığıdır. Artık davranışını belirlemek için bu nesne üzerindeki fiziksel kuvvetleri simüle edilebilmektedir. CAD modelleri formun statik temsilleriyken, simülasyonlar sadece formun değil aynı zamanda davranışın da dinamik temsilleridir.

Grieves ve Vickers (2016), Sistem Davranışı modelinin Grieves ve Vickers Kategorilerini (Bakınız Şekil 4.3) kullanarak fiziksel ikizlerin, yani Akıllı, Bağlantılı Ürün Sistemleri (SCPS-Smart, Connected Product Systems)'nin sistem karmaşıklığını artırdığını, sistemlerin bilgi işlem ve iletişim yeteneği olduğunda çok daha fazla davranışsal seçenek sonuçları olduğunu söylemektedir. Bu davranışlar, öngörülen istenen işlevlerdir. Ancak bu durumda diğer tüm davranış kategorilerinde de artış olacaktır. Öngörülen istenmeyen ve öngörülemeyen istenmeyen davranış değerlerinde orantılı bir artış olması beklenir. Ayrıca öngörülemeyen istenmeyen davranışlar da oluşabilir ki bu durum, sistem hakkında düşündüğümüz kadar bilgili olmadığımız gerçeğini göstermektedir.

Dijital ikiz, sistem karmaşıklığını azaltma yeteneğine sahiptir. Ürün yaşam döngüsünün oluşturma aşamasında, sistem performansını modelleme ve simüle etme yeteneği, öngörülemeyen davranışları tanımlayarak ve bunları öngörülen davranışlara taşıyarak azaltmayı sağlayabilir. Ön-

görülen istenmeyen davranışlar daha sonra ele alınabilir. Yaşam döngüsünün operasyonel aşamasında, dijital ikiz simülasyonları ürün performansından elde edilen gerçek verilerle yürütebilir. Bu, öngörülemeyen istenmeyen davranışları gerçekten oluşmadan önce tanımlamaya ve ele almaya izin vermektedir.



Şekil 4.3: Sistem Davranışı modelinin Grieves ve Vickers Kategorileri (Kaynak: Grieves & Vickers, 2016)

Dijital ikiz kavramının gerçeğe dönüşmesi, bağlamsal ve operasyonel verileri yakalamak ve nesneyi dijital ikizinden kontrol etmek için, bir varlığın içindeki bir parça veya varlığın tamamı gibi fiziksel bir nesneye sensörler ve aktüatörler eklemek Nesnelerin İnterneti (Internet of Things-IoT) teknolojisindeki gelişmelerle mümkün hale gelmiştir.

Dijital ikiz uygulamalarının çoğu, bir varlık içindeki tek bir parçanın performansını izlemek gibi küçük kapsamla başlar zamanla genişler. Bu iki şekilde olur. İlk olarak, organizasyon tüm bir makinenin, varlığın veya iş sürecinin tam bir resmini vermek için bir dizi daha küçük dijital ikizi bir araya getirir. İkincisi, kuruluşlar mevcut bir dijital ikize simülasyonlar gibi daha karmaşık yetenekler ekler. Slevin (2018) de bu genişlemenin iki şekilde olduğunu söylemektedir. Birincisinde; organizasyon tüm bir makinenin, varlığın veya iş sürecinin tam bir resmini vermek için bir dizi daha küçük dijital ikizi bir araya getirmektedir. İkincisinde ise kuruluşların mevcut bir dijital ikize simülasyonlar gibi daha karmaşık yetenekleri eklemesiyle gerçekleşir. Her iki durumda da, artan ihtiyaçları karşılamak için dijital ikizin içindeki işlevsellik toplanması ve yönetilmesi gereken ekstra verileri elde edebilmek için performansı korurken, ölçeklemek için güvenli bir şekilde işlevsellik ekleyebilmek gerekir (Grieves, 2017; Slevin, 2018).

Dijital ikiz, basit bir şekilde fiziksel den dijitale ve dijitalden fiziksele iki yönlü bir veri akışı olarak karakterize edilmektedir. Varlık, varlığın dijital ikizi ve sahip olduğu verileri görüntülemek, yönetmek veya işlemek için dijital ikize erişmesi gereken herkes arasında veri akışı olur. Dijital ikiz aracılığıyla ne kadar ileri analizler ve simülasyonlar gerçekleştirilirse sürece o kadar fazla insan dahil

olacaktır. Bununla birlikte her dijital ikiz nesnenin veya varlığın tam bir resmini oluşturmak için başlangıç modelini oluşturmaya ve gerçek zamanlı veya neredeyse gerçek zamanlı olarak destekleyici verileri sağlamaya yardımcı olan birden fazla bilgi akışı içermektedir. Bu bilgi akışı sadece fiziksel nesne ile ikizi arasında değil aynı zamanda dijital ikiz ile Bilgisayar Destekli Tasarım (Computer Aided Design - CAD), Kurumsal Kaynak Planlama (Enterprise Resource Planning – ERP), Ürün Yaşam Döngüsü Yönetimi (Product ieves'in Lifecycle Management – PLM) ve İmalat Yürütme Sistemleri (Manufacturing Execution Systems – MES) gibi kurumsal sistemler arasındadır (Grieves, 2017; Slevin, 2018). Jones ve arkadaşları (2020), Grieves'in ürün yaşam döngüsü bağlamında Tablo 1 ve Şekil 4'deki terimleri kullanarak dijital ikiz oluşturma sürecini aşağıdaki aşamalarla aktarmıştır.

1. İlk olarak dijital ikizin tasarım aşamasıyla dijital ikiz Prototip olarak hayata başlar,
2. Gerçekleştirme aşamasında üretilen her bir ürün için dijital ikiz Örneği oluşturulur,
3. Dijital İkiz Örneklerinin toplamı Dijital İkiz Toplamı'nı oluşturur. Hem Örnekler hem de Toplamı; simülasyon, modelleme ve değerlendirme gibi sanal teknikleri mümkün kılan, fiziksel ürünün bulunduğu ortamın sanal temsili olan Dijital İkiz Çevresinde bulunur.
4. Dijital İkiz Örnekleri /Toplamı ve Çevresi, kullanımdan Ayrılma /İmha etme aşamasıyla sonlanan fiziksel ürünün gerçek ömrünün de ötesinde varlığını sürdürür.

Tablo 4.1: Digital Twin'i çevreleyen temel kavramların listesi ve açıklamaları (Kaynak: Jones ve arkadaşları, 2020)

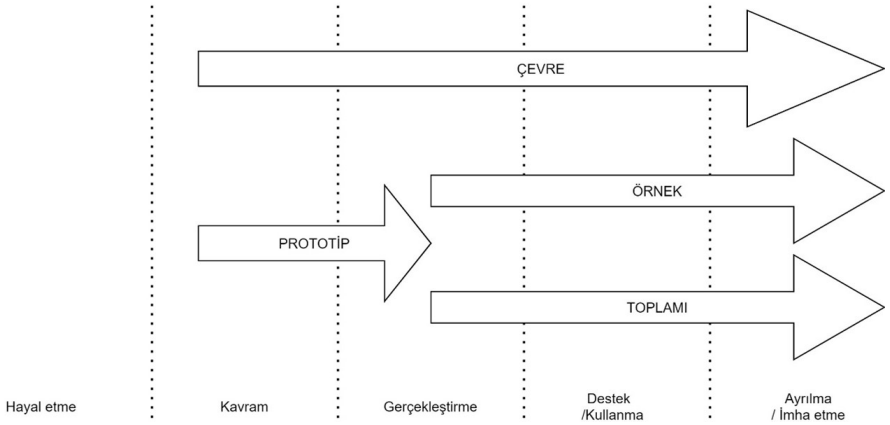
Dijital İkiz	Hem mikro hem de makro düzeyde doğru olan fiziksel bir ürünün eksiksiz bir sanal açıklaması.
Dijital İkiz Prototip	Fiziksel ikizi oluşturmak için gerekli tüm bilgileri içeren bir prototip ürünün sanal açıklaması.
Dijital İkiz Örnek	Ürün ömrü boyunca tek bir ürüne bağlı kalan belirli bir fiziksel ürün örneği.
Dijital İkiz Toplamı	Tüm Dijital İkiz Eşgörünüm kombinasyonu.
Dijital İkiz Ortam	Dijital ikizler üzerinde çalışmak için çok alanlı bir fizik uygulama alanı. Bu işlemler, performans tahmini ve bilgi sorgulamayı içerir.

Slevin (2018) Dijital ikiz oluşturma sürecini ; 1. Tasarım, 2. İşlem, 3. Çoğaltma aşamaları olmak üzere üç adımla tanımlamıştır.

Tasarım Aşaması:

Dijital ikizin tasarlanması aşamasında iki ana unsur vardır.

1. Nesnelerin interneti (IoT) cihazlarından gerçek zamanlı veri akışını ve diğer kurumsal sistemlerden operasyonel ve işlemsel bilgilerle entegrasyonu sağlamak için fiziksel varlığı dijital ikizine entegre etmek için ihtiyaç duyduğunuz etkinleştirme teknolojisini seçmek gerekir. İhtiyaç duyulan cihaz türü, varlığın 3B (3 boyutlu) temsilini oluşturmak için gereken modelleme yazılımı ve dijital ikiz içindeki bilgilere kimin erişeceği veya fiziksel varlığın kontrolünü bu sayede ele geçireceği konusunda net olmak gerekir. Güvenli IoT cihaz yönetimi, bulunulan ağdaki cihazları tanımlamayla ilgili risklerin üstesinden gelmek için çok önemlidir. Bu; her aygıtın kimliğini doğrulamak, hazırlamak, yapılandırmak, izlemek ve yönetmek için gerekli yetenekleri sağlar. Kimlik odaklı bir IoT platformu, bunu geniş ölçekte hızlı ve güvenli bir şekilde yapmayı sağlar.



Şekil 4.4: Dijital ikiz öğeleri ile fiziksel ürün arasındaki kapsam ve ilişkiler. (Kaynak: Jones ve arkadaşları, 2020)

2. Varlığın yaşam döngüsü boyunca gereken bilgi türünü, bu bilgilerin nerede saklandığını ve bunlara nasıl erişilebileceğini ve kullanılabileceğini anlamak gerekir. Bilgilerin, sistemler arasında hızlı ve etkili bir şekilde değiş tokuş edilebilecek, yeniden kullanılabilir bir şekilde yapılandırılması önemlidir. Kimlik odaklı bir IoT platformu dijital ikizde yer alan her öğenin kimliğini yönetebilir. Bu sayede insanlar, sistemler ve şeyler arasındaki güvenli iletişimi otomatikleştirmek için mesajlaşma hizmetleri sağlayabilir.

İşlem Aşaması:

Öncelikle dijital ikizin hangi işlevi yerine getirmek için oluşturulacağına karar vermek gereklidir. Buna karar vermek için aşağıdaki sorular sorulmaktadır.

(a) Dijital ikizle sadece fiziksel varlığı izlemek için mi?
(b) Dijital ikizin fiziksel varlığı kontrol etmesi ve değiştirmesi amacıyla mı?
(c) Tahmine dayalı bakıma yardımcı olmak için fiziksel varlıktan verileri ileri analizler için kullanılabilir hale getirmek istiyor musunuz?
(d) Operasyonel performansa ve ürün geliştirmeye yardımcı olacak simülasyonlar gerçekleştirmek için dijital ikiz içindeki verileri ve modelleri kullanmak istiyor musunuz?

Yukarıdaki soruların cevabı, fiziksel varlığa eklenecek cihaz türlerini ve bilgi işlemenin uç noktalara taşınmasına izin veren daha karmaşık cihazlar kullanıp kullanılmayacağını ayrıca entegrasyon ve veri hazırlığını belirleyecek ve yönetim gereksinimlerini belirleyecektir. Dijital ikiz için uygulama ne kadar karmaşıksa, bu yetenekler o kadar kapsamlı olmaktadır. Örneğin, çoğu dijital ikiz, operasyonel performansı ve karar vermeyi iyileştirmek için analitikten yararlanmaya çalışacaktır. Verilerin nasıl alındığını, saklandığını, hazırlandığını ve sunulduğunu kontrol etmek, ileri analiz uygulamasını sağlamak için çok önemlidir. Yüksek kaliteli sonuçlar elde etmek için IoT cihazlarınızdan gelen verilerin kalitesini garanti etmek gerekir. Veri aktarma ve kabul etme hakları da dahil olmak üzere her IoT cihazı doğrulanır. Tasarıma göre kimlik yaklaşımı benimsemek, bu yetenekleri en başından itibaren dijital ikize entegre etmektedir.

Çoğaltma Aşaması:

Dijital ikiz uygulamalarının çoğu, bir varlık içindeki tek bir parçanın performansını izlemek gibi küçük başlar, ancak zamanla genişler. Bu iki şekilde olur. İlk olarak, organizasyon tüm bir makinenin, varlığın veya iş sürecinin tam bir resmini vermek için bir dizi daha küçük dijital ikizi bir araya getirir. İkincisi, kuruluşlar mevcut bir dijital ikize simülasyonlar gibi daha karmaşık yetenekler

ekler. Her iki durumda da, bu artan ihtiyaçları karşılamak için dijital ikizin içindeki işlevselliği yok saymak veya değiştirmek uygun değildir. Toplanması ve yönetilmesi gereken ekstra verileri karşılamak için performansı korurken, ölçeklemek için güvenli bir şekilde işlevsellik ekleyebilmek gerekir.

Kimlik odaklı bir IoT platformu, yeni cihazların ve uygulamaların ikiz ile bağlantı kurmasına ve etkileşime girmesine olanak tanıyan kapsamlı entegrasyon ve açık API'ler (Application Programming Interface : Uygulama Programlama Arayüzü) aracılığıyla dijital ikizin yeteneklerini hızlı ve güvenli bir şekilde genişletmeyi sağlamaktadır (Slevin, 2018).

Dijital İkiz Teknolojisi; Küresel endüstriler, ağır varlıklar, karmaşık üretim hatları ve alana özgü büyük miktarda veri ile tasarlanmıştır. Digital ikiz teknolojisi, bir süreç boyunca malzemelerin akışını gerçek zamanlı olarak görselleştirmek için kullanılır. Teknoloji, üreticilerin sanal olarak simüle edilmiş görüntüler ve görseller yardımıyla malzeme ve ekipmanların fiziksel yönünü dijital bir ekranda temsil etmelerini sağlar. Bu sayede üretim birimlerinde süreç akışını izlemek için gereken çaba, maliyet ve zamanın çoğunu azaltmanın iyi bir yoludur.

4.6 Dijital İkizin Kullanım Alanları

Dijital ikiz kullanarak ulaşım araçlarının motorları, trenler, açık deniz platformları ve türbinler gibi nesneler üretilmeden önce sanal olarak tasarlanıp, test edilebildiği gibi dijital ikiz bakım-onarım işlerine yardımcı olmak amacıyla da kullanılabilirler. Örneğin, mühendisler, gerçek hayattaki ürünü uygulamaya geçirmeden önce parçalar için gereken bir düzeltmenin uygulanabilirliğini test etmek için de dijital bir ikiz kullanabilirler. Dijital ikiz akıllı fabrikaların kurulmasında ve tüm üretim süreçlerinin simüle edilmesinde de kullanılmaktadır.

4.6.1 Akıllı Şehirler

Dijital İkizlerin akıllı bir şehirde önemli ölçüde etkili olma potansiyeli ve kullanımı, IoT aracılığıyla bağlantıdaki hızlı gelişmeler nedeniyle yıldan yıla artmaktadır. Akıllı şehir sayısı arttıkça bağlantıda daha fazla akıllı şehir var ve bununla birlikte daha fazla dijital ikiz kullanımı gerçekleşecek demektir. Bunun yanı sıra bir şehirdeki temel hizmetlere gömülü IoT sensörlerinden ne kadar fazla veri toplanırsa bu da gelişmiş AI algoritmalarının oluşturulmasını amaçlayan araştırmaların önünü açacaktır (Fuller ve ark., 2020; Mohammadi & Taylor, 2017; Sivalingam ve ark., 2018; Pargmann, ve ark., 2018).

4.6.2 İmalat Sektörü

Dijital ikiz için bir sonraki tanımlanan kullanım alanı bir üretim hattı içindir. Üreticilerin her zaman, herhangi bir üretici için önemli bir itici güç ve motivasyon olan zamandan ve paradan tasarruf etmek amacıyla ürünlerin izlenebileceği ve izlenebileceği bir yol araması bunun en büyük nedenidir. Aynı şekilde, akıllı bir şehrin gelişmesiyle birlikte bağlantı, üretimin dijital ikizleri kullanmasındaki en büyük itici güçlerinden biridir. Mevcut büyüme, 4. sanayi devrimini oluşturan Endüstri 4.0 konsepti ile uyumludur. Dijital ikiz, üretim hattı geri bildiriminin yanı sıra makine performansı hakkında gerçek zamanlı durum verme potansiyeline sahiptir. Üreticiye sorunları daha erken tahmin etme yeteneği verir. Digital Twin kullanımı, cihazlar arasındaki bağlantıyı ve geri bildirimi artırarak güvenilirliği ve performansı artırır. Dijital ikizlerle birleştirilmiş yapay zeka algoritmaları, makine performans ve tahmin analizi için gerekli olan büyük miktarda veriyi tutabildiğinden daha fazla doğruluk potansiyeline sahiptir (Fuller ve ark., 2020; Longo ve ark., 2019).

Dijital ikizin uygulandığı başka bir alan olan otomotiv sektörü, en yeni araçları oluşturmak için en son teknolojileri hızla benimseyen bir endüstridir. Birçok otomobil şirketi, verimliliği artırmak ve kullanıcı deneyimini tatmin etmek için sürekli olarak yeni otomobil markaları üzerinde yenilikler ve deneyler yapmaktadır. Otomobil endüstrisinde dijital ikizler uygulamak, güvenliği sağlamak için deneysel test sürüşlerine ve sensör işlemlerine yardımcı olabilir. Ayrıca bu teknik, üretim maliyetini düşürür ve üretim hattı performansını genişletir. Günümüzde bazı otomobil üreticileri dijital ikizi şimdiden benimsemiştir. Tesla ve Rolls Royce gibi büyük şirketler kullanıcı deneyimini artırmak ve güvenliği sağlamak adına araçlarının sanal ortamda dijital ikizlerini oluşturur (Özgür, 2020; Fuller ve ark., 2020).

İnşaat sektörü, dijital ikiz kullanım için bir dizi uygulamaya ev sahipliği yapan başka bir sektördür. Bir binanın veya yapının geliştirme aşaması, dijital ikiz için potansiyel bir uygulamadır. Teknoloji, yalnızca akıllı şehir binalarının veya yapılarının geliştirilmesinde uygulanamaz, aynı zamanda devam eden gerçek zamanlı bir tahmin ve izleme aracı olarak da kullanılabilir. Dijital ikizin ve veri analitiğinin kullanılması, sanal olarak yapılan ve daha sonra fiziksel olarak uygulanan herhangi bir değişiklikle binaları ve yapıları tahmin ederken ve bakımını yaparken potansiyel olarak daha fazla doğruluk sağlayacaktır. Algoritmalar, fiziksel binadan önce dijital ikiz içinde gerçek zamanlı olarak uygulanabildiğinden, dijital ikiz, simülasyonları gerçekleştirirken inşaat ekiplerine sağladığı veri daha fazla doğruluk değeri taşımaktadır (Longo ve ark., 2019).

4.6.3 Sağlık

Sağlık sektörü, dijital ikiz teknolojisinin uygulandığı başka bir alandır. Bir zamanlar imkansız olan şeyler mümkün hale geldiğinden, teknolojinin sağlık hizmetlerine katkısı nedeniyle alandaki büyüme ve gelişmeler emsalsizdir. Nesnelerin interneti açısından, cihazlar daha ucuz ve uygulanması daha kolay olduğu için bağlantılarda artış olmaktadır.[37], [38]. Artan bağlanabilirlik, sağlık sektöründe sadece dijital ikiz kullanımının potansiyelini büyütmektedir. Gelecekteki uygulama, vücudun gerçek zamanlı analizini veren bir insanın dijital ikizi olacaktır. Daha gerçekçi olan güncel bir uygulama, belirli ilaçların etkilerini simüle etmek için kullanılan bir dijital ikizdir. Başka bir uygulama, cerrahi prosedürleri planlamak ve gerçekleştirmek için dijital ikiz kullanılmaktadır (Gahlot ve ark., 2019).

4.7 Endüstride Dijital İkiz

General Electric (GE), dijital ikiz kullanımını ilk olarak 2016 yılında bir patent başvurusunda belgelemiştir. Patentte belirtilen tasarımla dijital ikizler oluşturmak için bir araç olan 'Predix' platformu adlı bir uygulama geliştirdiler. Predix, veri analitiği ve izleme süreci için kullanılır. Son yıllarda GE, bir yazılım şirketi yerine endüstriyel çok uluslu bir şirket olarak kaynaklarına odaklanmayı planlayarak dijital ikiz planlarını küçültmüştür. Ancak Siemens, Makineleri ve fiziksel altyapıyı bir dijital ikize bağlayan bulut tabanlı bir sistemle Endüstriyel 4.0 konseptini benimseyen 'MindSphere' adlı bir platform geliştirmiştir. İşletmeleri dönüştürme ve dijital ikiz çözümleri sağlama umuduyla tüm bağlı cihazları ve milyarlarca veri akışını kullanmaktadır (Magargle ve ark., 2017; Petrik & G. Herzwurm, 2019).

Dijital ikiz ve yapay zeka teknolojisini geliştirmek için alternatif bir platform da PTC tarafından oluşturulan 'ThingWorx'dir. Platformun ana odak noktası Endüstriyel Nesnelerin İnterneti veya nesnelerin interneti (IIoT/IoT) verilerini toplamak ve kullanıcılara değerli bilgiler sağlamaktır. Sezgisel, rol tabanlı bir kullanıcı arabirimi aracılığıyla sunan bu Endüstriyel İnovasyon Platformu, dijital ikiz çözümü için bir ortam geliştirirken veri analitiğinin sorunsuz gelişimini kolaylaştırmaktadır (Chen ve ark., 2018).

IBM, milyonlarca IoT cihazından toplanan veriler aracılığıyla büyük ölçekli sistemleri gerçek zamanlı olarak yönetmek için kullanılabilen çok yönlü bir nesnelerin interneti veri aracı olarak pazarlanan 'Watson IoT Platformu' adlı bir platform geliştirmiştir. Platformun bulut tabanlı hizmetler, veri analitiği, uç yetenekler ve blok zincir gibi çeşitli ek özellikleri vardır. Tüm bu özellikler, uygulamayı dijital ikiz sistem için olası bir platform haline getirir (Kumar ve Jasuja, 2017).

Endüstride açık kaynaklı olarak öne çıkarılabilecek iki büyük proje vardır. Birincisi, bir dijital ikizin durumlarını yönetebilen, fiziksel ve dijital ikizlere erişim ve kontrol sağlayan, kullanıma hazır bir platform olan Eclipse'in 'Ditto' projesidir. Platform, halihazırda bağlı cihazlar için destek sağlayan ve dijital ikizlerin bağlantısını ve yönetimini basitleştiren bir arka uç rolündedir (Damjanovic-Behrendt, 2018). Bentley Systems tarafından geliştirilen 'imodel.js' adlı bir başka açık kaynaklı proje, dijital ikizler oluşturmak ve bunlara erişmek için ve oluşturulmuş bir platformdur (Bentley Systems, 2018).

4.8 Sonuç

Eski bir tarihe sahip olan dijital ikiz teknolojisi nesnelerin internetinin günümüzdeki yükselişi ve endüstri 4.0 ile son yıllarda sürekli olarak stratejik bir teknoloji trendi olarak adlandırılmaktadır. Dijital ikiz uygulamalarındaki artış büyük ölçüde nesnelerin interneti, büyük veri, çoklu fiziksel simülasyon ve Endüstri 4.0, gerçek zamanlı sensörler ve sensör ağları, veri yönetimi, veri işleme gibi veri odaklı ve dijital bir üretim geleceğiyle ilgili teknolojilerdeki ve girişimlerdeki gelişmelerden kaynaklanmaktadır.

Dijital İkiz, son yıllarda uzay ve havacılık alanında kavramsal bir temel olarak benimsenmiştir. NASA, teknoloji yol haritalarında ve sürdürülebilir uzay araştırmaları tekliflerinde kullanmıştır (Akt: Grieve, 2016; Caruso ve ark., 2010; Piascik ve ark., 2010).

Dijital İkiz fikri, sistemlerin sanal versiyonunu tasarlayabilmek, test edebilmek, üretebilmek ve kullanabilmektir. Mevcut ürünle birlikte gelecekteki ürünlere uygulanan öğrenme ile ürün bilgileri yakalanır, depolanır ve değerlendirilir. Tasarımların gerçekten üretilebilir olup olmadığını anlamak, sistem kullanımdayken arıza modlarını belirlemek gerekir. Fiziksel sistem fiilen üretilmeden önce tüm bu bilgilere ihtiyacımız var. Bu, fiziksel sistemin konuşlandırıldığında ve kullanımdayken arızalarını azaltacak, maliyet, zaman ve kullanıcılarına verilen zararı azaltacaktır (Grieves & Vickers, 2017). Bu süreç özünde bir ürünün yaşam döngüsü boyunca izlenmesi, yönetimi ve geliştirilmesine yönelik bilgili, veriye dayalı bir yaklaşımın uygulanmasını sağlar.

Küresel endüstriler, ağır varlıklar, karmaşık üretim hatları ve alana özgü büyük miktarda veri ile tasarlanmıştır. Digital ikiz teknolojisi, bir süreç boyunca malzemelerin akışını gerçek zamanlı olarak görselleştirmek için kullanır. Teknoloji, üreticilerin sanal olarak simüle edilmiş görüntüler ve görseller yardımıyla malzeme ve ekipmanların fiziksel yönünü dijital bir ekranda temsil etmelerini sağlar. Bu sayede üretim birimlerinde süreç akışını izlemek için gereken çaba, maliyet ve zamanın çoğunu azaltmanın iyi bir yoludur.

Sonuç olarak hem akademi hem de endüstri, dijital ikizleri veya temsil ettiği ilkeleri araştırmakta, geliştirmekte ve uygulamaya çalışmaktadır. Dijital ikiz uygulamalarındaki artış büyük ölçüde Nesnelerin İnterneti, büyük veri, çoklu fiziksel simülasyon ve Endüstri 4.0, gerçek zamanlı sensörler ve sensör ağları, veri yönetimi, veri işleme gibi veri odaklı ve dijital bir üretim geleceğiyle ilgili teknolojilerdeki ve girişimlerdeki gelişmelerden kaynaklanmaktadır. Nesnelerin internetinin bir alt dalı olan dijital ikiz, sağlayacağı kolaylık ile ilerleyen dönemlerde hayatımızın çoğu alanında kullanılacak gibi görünmektedir. Ancak dijital ikiz geliştirmenin tüm biçimlerinde bu tür sistemlerin modellenmesiyle ilgili standartlaştırılmış bir yaklaşım olmadığı için zorluk vardır. İlk tasarımdan

dijital ikiz simülasyonuna, ister fizik tabanlı ister tasarım tabanlı olsun, standart bir yaklaşım olması gerekir. Standartlaştırılmış yaklaşımlar, bir dijital ikizin geliştirilmesi ve uygulanmasının her aşaması sırasında bilgi akışını sağlarken bir yandan da alan ve kullanıcı anlayışını sağlar. Bu nedenle alanda söz sahibi kurumların standartlaşma çalışmalarını tamamlanması büyük önem taşımaktadır.

4.9 Kaynaklar

Aynacı, İ. (2020). Dijital İkiz Ve Sağlık Uygulamaları. İzmir Kâtip Çelebi Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, 3(1):70- 82. https://dergipark.org.tr/en/download/article-file/1095385 (Erişim Tarihi, 11 Nisan 2021).

Bentley Systems. (Oct. 2018). Releases Imodel.Js Open-Source Library.

Caruso, P., D. Dumbacher & M. Grieves (2010). Product Lifecycle Management and the Quest for Sustainable Space Explorations. AIAA SPACE 2010 Conference & Exposition. Anaheim, CA.

Chen, X., Kang, E., Shiraishi, S., Preciado, V. M. & Jiang, Z. (2018). Digital behavioral twins for safe connected cars, Proceedings. 21th ACM/IEEE Int. Conf. Model Driven Eng. Lang. Syst. MODELS, pp. 144-153.

Cityzenith. (9 October 2018). What are Digital Twins? What the Building and Real Estate Industries Need to Know. Cityzenith,

Damjanovic-Behrendt, V. (2018). A digital twin-based privacy enhancement mechanism for the automotive industry. Proceedings. Int. Conf. Intell. Syst. (IS), pp. 272-279, Sep. 2018.

Doğan, Ö. (Ağustos 16, 2020). Dijital İkiz Teknolojisi Nedir?. Teknoloji.org https://teknoloji.org/dijital-ikiz-teknolojisi-nedir/ (Erişim Tarihi, 11 Ağustos 2021)

Ercan, T., Kutay, M. (2016). Endüstride Nesnelerin Interneti (IoT) Uygulamaları. Afyon Kocatepe Üniversitesi Fen ve Mühendislik Bilimleri Dergisi. 16, 035102, 599-607. DOI: 10.5578/fmbd.43411 https://dergipark.org.tr/en/download/article-file/657943 (Erişim Tarihi, 11 Ekim 2021).

Fuller, A., Fan, Z., Day C. & Barlow, C. (2020). Digital Twin: Enabling Technologies, Challenges and Open Research, in IEEE Access, vol. 8, pp. 108952-108971, https://ieeexplore.ieee.org/abstract/document/9103025/references#references

Gahlot, S., Reddy, S. R. N. & Kumar, D. (2019). Review of smart health monitoring approaches with survey analysis and proposed framework. IEEE Internet Things J., vol. 6, no. 2, pp. 2116-2127.

Gökrem, L., Bozuklu, M. (2016). Nesnelerin İnterneti: Yapılan Çalışmalar ve Ülkemizdeki Mevcut Durum. Gaziosmanpaşa Bilimsel Araştırma Dergisi. 13, 47-68

Grieves, M. (2006). Product Lifecycle Management: Driving the Next Generation of Lean Thinking. New York, McGraw-Hill.

Grieves, M. (2011). Virtually perfect : Driving Innovative and Lean Products through Product Lifecycle

Management. Cocoa Beach, FL, Space Coast Press.

Grieves, M. (2011). Virtually perfect : Driving Innovative and Lean Products through Product Lifecycle

Management. Cocoa Beach, FL, Space Coast Press.

Grieves, M. (2011). Virtually perfect : Driving Innovative and Lean Products through Product Lifecycle Management. Cocoa Beach, FL, Space Coast Press.

Grieves, M. (2006). Product Lifecycle Management: Driving the Next Generation of Lean Thinking. New York, McGraw-Hill.

Grieves, Michael. (2015). Digital Twin: Manufacturing Excellence through Virtual Factory Replication. https://www.researchgate.net/publication/275211047_Digital_Twin_Manufacturing_Excellence_through_Virtual_Factory_Replication (Erişim Tarihi, 13 Ekim 2021).

Grieves, M. (2016). Origins of the Digital Twin Concept. 10.13140/RG.2.2.26367.61609. https://www.researchgate.net/publication/307509727_Origins_of_the_Digital_Twin_Concept (Erişim Tarihi, 13 Ekim 2021).

Grieves, M., & Vickers, J. (2017). Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems. In Transdisciplinary perspectives on complex systems(pp. 85- 113). Springer, Cham. https://www.researchgate.net/profile/Michael-Grieves/publication/306223791_Digital_Twin_Mitigating_Unpredictable_Undesirable_Emergent_Behavior_in_Complex_Systems/links/5aa54e1ea6fdccd544bc386f/Digital-Twin-Mitigating-Unpredictable-Undesirable-Emergent-Behavior-in-Complex-Systems.pdf

Gubbi, J., Buyya, R., Marusic, S. & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. Future Generation Computer Systems. Volume 29, Issue 7, 164 https://doi.org/10.1016/j.future.2013.01.010. https://www.sciencedirect.com/science/article/pii/S0167739X13000241) 5-1660, (Erişim Tarihi, 13 Ekim 2021)

Haag, S. & Anderl, R. (2018). Digital Twin – Proof of Concept. Manufacturing Letters. 15. 10.1016/j.mfglet.2018.02.006.

Houten, H.V. (2018). The rise of the digital twin: how healthcare can benefit. Philips Haber Merkezi. https://www.philips.com/content/corporate/en_AA/about/news/archive/blogs/innovationmatters/20180830-the-rise-of-the-digital-twin-how-healthcare-can-benefit.html/ (Erişim Tarihi, 20 Nisan 2019).

Jones, D., Snider, C., Nassehi, A., Yon, J. & Hicks, B. (2020). Characterising the Digital Twin: A systematic literature review. CIRP Journal of Manufacturing Science and Technology. Volume 29, Part A, 36-52, ISSN 1755-5817. https://doi.org/10.1016/j.cirpj.2020.02.002. (https://www.sciencedirect.com/science/article/pii/S1755581720300110). (Erişim Tarihi, 13 Nisan 2021).

Kumar, S. & Jasuja, A. (2017). Air quality monitoring system based on IoT using raspberry pi, Proceedings Int. Conf. Comput. Commun. Autom. (ICCCA), pp. 1341-1346, May 2017.

Lee, I., Lee, K. (2015). The Internet of Things (IoT): Applications, investments, and challenges for enterprises. Business Horizons. 58 (4), 431-440. doi.org/10.1016/j.bushor.2015.03.008, https://www.sciencedirect.com/science/article/pii/S0007681315000373 (Erişim Tarihi, 13 Ekim 2021)

Longo, L., Nicoletti, F. & Padovano, A. (2019). Ubiquitous knowledge empowers the smart factory: The impacts of a service-oriented digital twin on enterprises' performance. Annual. Review. Control. vol. 47, pp. 221-236

Madni, A.M. (2018). Next Generation Adaptive Cyber-Physical Systems. In Proceedings of the 21st Annual Systems Engineering Conference, Tampa, FL, USA, 22–24 October 2018.

Madni, A.M., Madni, C.C. & Sievers, M. (2018). Adaptive Cyber-Physical-Human Systems. In Proceedings of the 2018 INCOSE International Symposium, Washington, DC, USA, 7–12 July 2018

Madni, A.M., Sievers, M., Ordoukhanian, E., Pouya, P. & Madni, A. (2018). Extending Formal Modeling for Resilient Systems. In Proceedings of the 2018 INCOSE International Symposium, Washington, DC, USA, 7–12 July 2018.

Madni, A.M., Sievers, M., Erwin, D., Madni, A., Ordoukhanian, E. & Pouya, P. (2019). Formal Modeling of Complex Resilient Networked Systems. In Proceedings of the AIAA Science and

Technology Forum, San Diego, CA, USA, 7–11 January 2019.

Magargle, R., Johnson, L., Mandloi, P., Davoudabadi, P., Kesarkar, O., Krishnaswamy, S. et al. (2017). A simulation-based digital twin for model-driven health monitoring and predictive maintenance of an automotive braking system. Proceedings. 12th Int. Modelica Conf., pp. 35-46, Jul. 2017.

Marr, B. (2017). What Is Digital Twin Technology – And Why Is It So Important? https://www.forbes.com/sites/bernardmarr/2017/03/06/what-is-digital-twin-technology-andwhy-is-it-so-important/#4d67dd832e2a (Erişim Tarihi, 13 Nisan 2021).

Mohammadi N., Taylor, J. E. (2017). Smart city digital twins, Proceedings. IEEE Symp. Ser. Comput. Intell. (SSCI), pp. 1-5. Nov. 2017

Pargmann, H., Euhausen, D. & Faber, R. (2018). Intelligent big data processing for wind farm monitoring and analysis based on cloud-technologies and digital twins: A quantitative approach, Proceedings. IEEE 3rd Int. Conf. Cloud Comput. Big Data Anal. (ICCCBDA), pp. 233-237, Apr. 2018.

Petrik, D. & Herzwurm, G. (2019). IIoT ecosystem development through boundary resources: A siemens MindSphere case study, Proceedings. 2nd ACM SIGSOFT Int. Workshop Software-Intensive Bus. Start-ups Platforms Ecosyst. IWSiB, pp. 1-6.

Piascik, R., J. Vickers, D. Lowry, S. Scotti, J. Stewart. & A. Calomino (2010). Technology Area 12: Materials, Structures, Mechanical Systems, and Manufacturing Road Map, NASA Office of Chief Technologist

Puri, D. (2017). Oracle's digital twins implifies design process for complex IoT systems. https://www.networkworld.com/article/3235962/oracles-digital-twin-simplifies-designprocess-for-complex-iot-systems.html (Erişim Tarihi, 11 Nisan 2021).

Rabah, K. (2018). Convergence of AI, IoT, Big Data and Blockchain: A Review. The Lake Institute Journal. Vol. 1, No. 1, 1 – 18. https://fardapaper.ir/mohavaha/uploads/2018/06/Fardapaper-Convergence-of-AI-IoT-Big-Data-and-Blockchain-A-Review.pdf (Erişim Tarihi, 11 Ekim 2021).

Scholten, A. (2017). Smart Buildings and Their Digital Twins. Realcomm20 Advis. Newslett. 17. https://www.realcomm.com/advisory/827/2/smart-buildings-and-their-digital-twins (Erişim Tarihi, 21 Ekim 2021)

Sivalingam, K., Sepulveda, M., Spring, M. & Davies, P. (2018). A review and methodology development for remaining useful life prediction of offshore fixed and floating wind turbine power converter with digital twin technology perspective, Procedings. 2nd Int. Conf. Green Energy Appl. (ICGEA), pp. 197-204, Mar. 2018.

Slevin, B. (Dec 12, 2018). How do you create a digital twin? Opentext Blogs. https://blogs.opentext.com/how-do-you-create-a-digital-twin/ (Erişim Tarihi, 11 Nisan 2021)

5. Veriyi Anlama: Python ile İstatistiğe Giriş

Veriyi Anlama: Python ile İstatistiğe Giriş

Selçuk KIRAN*, İlkim Ecem EMRE*
***Marmara Üniversitesi, İşletme Fakültesi, Yönetim Bilişim Sistemleri Bölümü**

5.1 Giriş

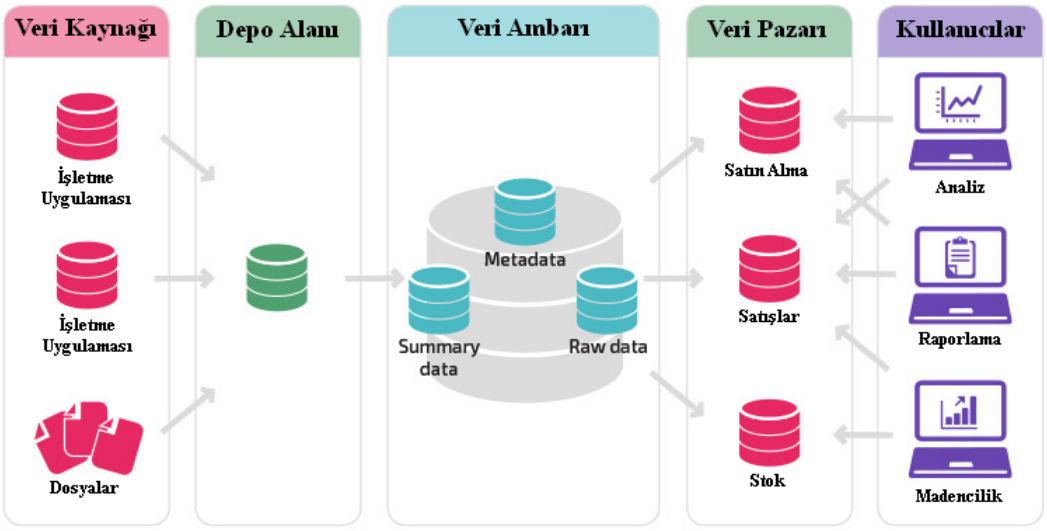
Bu bölüm kapsamında; yapay zekâ, makine öğrenmesi gibi günümüzün güncel çalışma alanlarında önemli bir aşama olan veriyi anlama sürecinden bahsedilecektir. Veriyi anlama birçok farklı alanda yapılan veri analizinin temelini oluşturmaktadır. Yazıda, bu analiz sürecinde temel istatistik yöntemlerinin nasıl kullanılabileceğinden bahsedilip bu işlemlerde kullanılabilecek temel Python kodları paylaşılmış ve açıklanmıştır.

5.2 Veriden Bilgiye

Veriden anlamlı bilgi elde etme yolunda en temel adımlardan biri veriyi anlama aşamasıdır. Yapay zekâ, makine öğrenmesi, veri madenciliği gibi veriden bilgi elde edebilme amacı ile yapılan çalışmalarda, birbirine benzer süreç adımları farklı şekillerde isimlendirilebilse de odaklanılan noktanın mevcut veri ile mümkün olan en iyi veya en yüksek performanslı sonuçların elde edilmesi olduğu söylenebilir. Bu amaç ile de yapılması gereken şey, araştırmacının elindeki veri setini iyi bir şekilde anlayabilmesi ve bu veri seti ile neler yapabileceğini kavrayabilmesidir.

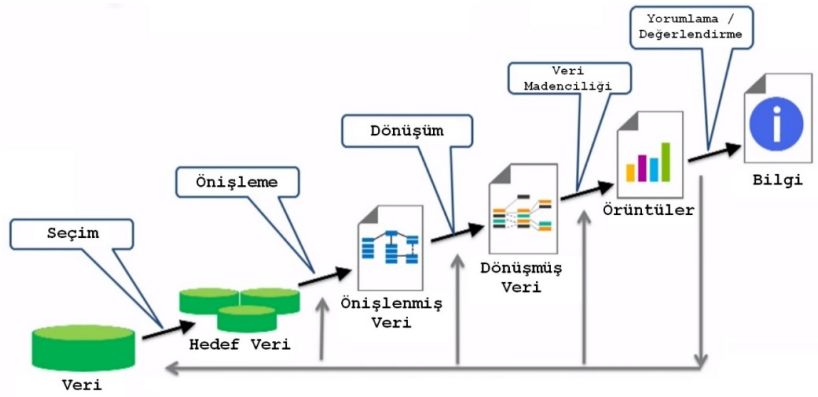
Veri tabanlarında toplanan gigabytelarca veri öncelikli olarak sanal depo alanlarında biriktirilmektedir. Bu biriktirilen veri, veri ambarına aktarılır. Verinin üst verisi (meta data) mümkün olduğunca korunurken, ham veri (raw data) doğrudan kopyalanır. Veri ambarında veriler ile ilgili oluşturulan özet veri (summary data) sayesinde de analistler verinin içeriği ile ilgili bir genel bakışa sahip olurlar. Veri ambarındaki veri, içeriğine uygun bir şekilde parçalara ayrılarak veri marketlerinde kullanıma sunulur. Veri marketlerinde oluşturulan yapı, analistler ile de paylaşılarak kullanıcıların analiz, raporlama, madencilik gibi görevleri yerine getirmesi sağlanır. Burada dikkat edilmesi gereken önemli nokta, veri yapısının, veri ambarına kadar olan aşamalarda (veri ambarı

aşaması dahil) güvenlik sebebiyle gizli olmasıdır. Veri marketleri aşamasıyla birlikte kullanılan alanlardan bazılarının yapısı analistlere açılır. Bu esnada kimlik bilgileri gibi kişisel veriler gizlilik kuralı uyarınca saklanmaya devam eder. Bu paragrafta anlatılan yapı Şekil 1'de görülebilir.



Şekil 5.1: Veri toplama süreci

Verilerin toplanarak hedef veri haline getirilmesi, veriyi analize hazırlamanın sadece başlangıç kısmıdır. Sonrasında veri önişleme ve dönüşüm aşamalarından geçerek sıra veri madenciliğine gelir. Burada yapılan analizler yorumlanıp değerlendirilerek yapılan çalışmanın sonuçları ortaya konur. Veritabanlarında bilgi keşfi (Knowledge Discovery in Databases – KDD (Fayyad vd., 1996)) olarak adlandırılan bu süreçle ilgili olarak Şekil 2'ye bakılabilir.



Şekil 5.2: Veri toplama süreci

Veri analizi ve modellemesi sırasında, veri hazırlamaya önemli miktarda zaman harcanır. Veri yükleme, temizleme, dönüştürme ve yeniden düzenleme aşamalarının genellikle bir analizin süre olarak %80 ve daha fazlasını aldığı söylenebilir (McKinney, 2018). Bazen verilerin dosyalarda veya veritabanlarında saklanma şekli, belirli görevler için doğru biçimde olmayabilir. Dolayısıyla araştırmacının çalışacağı veri setini çalışmadan önce analize hazır hale getirmesi gerekmektedir (McKinney, 2018).

2016'da yapılan bir araştırmada (GilPress, 2016) veri bilimcilere zamanlarının çoğunun hangi aşamada geçtiği sorulmuştur. Bu veri bilimcilerin büyük çoğunluğu (%60) zamanlarının büyük kısmını veri setini temizleme ve hazırlama ile geçirdiklerini belirtmişlerdir. Aynı araştırmaya göre veri bilimciler, zamanlarının %19'unu veri toplamaya, %9'unu örüntüleri bulmaya, % 4ünü algoritma iyileştirmeye ve %8'ini diğer işlemlere ayırmaktadırlar. İlk iki sırada çıkan işlerin ikisinin de veri hazırlama kısmına girdikleri göz önüne alınırsa veri bilimcilerin zamanlarının yaklaşık üçte birini veri hazırlamada geçtiğini belirttikleri görülebilir. Uzmanların görüşüne göre veriler analize hazırlandıktan sonra kalan veri madenciliği daha az zaman maliyetine sahiptir. Aynı çalışmada veri bilimcilere veri biliminin en keyifsiz kısmı da sorulmuştur. Veri bilimcilerin %57'si veri setini temizleme ve hazırlama, %21'i de veri toplama olmak üzere, toplamda %80'e yakın kısmı verinin analize hazırlanmasını en sıkıcı kısım olarak nitelendirmişlerdir. %10'u eğitim veri setini hazırlama cevabını verirken, %4'ü algoritmaları iyileştirme demiş, kalan %8 de diğer seçenekleri tercih etmişlerdir. Buradan da anlaşılan, verilerin analize hazırlanma kısmının en uzun süren kısım olmasının yanı sıra aynı zamanda en sıkıcı kısım olduğudur. Sıkıcılık veri hazırlığı esnasında dikkat dağılmasından dolayı daha çok hatanın oluşmasına da sebep olmaktadır.

Veri hazırlama kısmındaki bu yavaşlık ve sıkıcılık katsayılarını azaltmak için verinin toplanması kısmında kullanılan uygulamalarda biraz daha özenli olmakta fayda vardır. Örneğin bir formda kişilerden adres alınırken şehir bilgisi metin kutusu ile alındığı takdirde, İstanbul şehrini "İst.", "Kadıköy", "Acıbadem" gibi değişik kısaltmalar, ilçe ve mahalle adlarıyla ifade edenler çıkabileceği gibi "İsanbul", "İsstanbul" gibi yazım hataları ile yazanlar da çıkacaktır. Bu kelimelerin her biri veri tabanında farklı bir şehir ismiymiş gibi muamele gördüğünden verinin hazırlanması uzun vakit alacaktır. Bu gibi durumlarda uygulamada metin kutusu yerine açılır kutu (DropDownList) kullanılması tüm bu hataların önüne geçecektir ve haliyle verinin hazırlanmasında büyük bir zaman tasarrufu sağlayacaktır.

Bir başka önemli sorun da eksik veriden kaynaklanmaktadır. Eksik veri az ise bu satırların silinmesi iyi bir çözüm olup eksik verinin fazla olduğu durumlarda eldeki toplam veriyi çok azalttığından bu çözüm tercih edilmemelidir. Bu gibi durumlarda verinin ortalama değerle doldurulması tercih edilebildiği gibi aykırı değerlerin çok olduğu durumlarda medyan değeri de kullanılabilmektedir. Eğer bir uygulamadan geliyorsa, alanı zorunlu hale getirmek de iyi bir tercih olabilir, bu şekilde eksik verinin önüne geçilebilecektir.

Veri setinin temiz olması bu veri setinden elde edilecek modellerin kalitesini de etkileyecektir. Bu sebeple veri setinin mümkün olduğunca temiz bir şekilde analizlere hazır hale getirilmesi önem taşımaktadır. Bu noktada "garbage in, garbage out" (A Dictionary of Computing, 2008) yani "çöp girdi, çöp çıktı" prensibini hatırlamak gerekir. Verinin kalitesizliği çıktıları da etkilemekte yani kalitesiz, kötü girdi kalitesiz ve kötü çıktı elde edilmesine sebep olmaktadır.

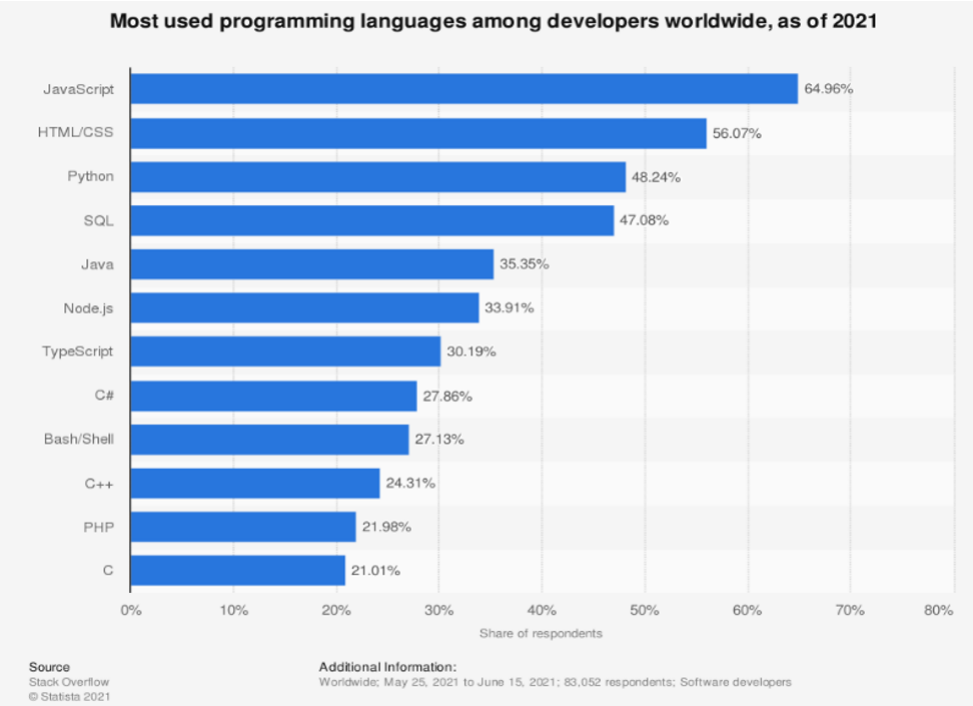
Makine öğrenmesi, veri madenciliği, yapay zekâ gibi uygulamalara başlamadan önce takip edilmesi gereken birtakım adımlar mevcuttur ancak hepsinin başında veriyi anlamak gelir. Verinin doğru bir şekilde anlaşılması sonraki adımlar için kolaylık sağlar. Bunun için hem grafiklerden hem de betimleyici temel istatistik yöntemlerinden yararlanılabilir. İstatistiksel analizler SPSS gibi araçlar kullanılarak yapılabileceği gibi farklı programlama dilleri de bu işin yapılmasına için imkân sağlamaktadır. Bu bölüm kapsamında yapılan uygulamalar Python dili kullanılarak gerçekleştirilmiştir.

5.3 Programlama Dilleri

Bir algoritmanın programlama dillerine uyarlanmış hali program olarak tanımlanır dolayısıyla bilgisayarın ne yapması gerektiğini söyleyen komutlar kümesine program denir (Çobanoğlu, 2019). Bilgisayarın temelde 0 ve 1 olarak algıladığı komutların yerine getirilmesini programlama dili sağlar. Yani programlama dili programcı ile bilgisayar arasında bir köprü kurararak bilgisayarın istenilen görevleri yerine getirilmesini sağlar. Python dili, 1991 yılında Hollandalı Guido van Rossum tarafından geliştirilmeye başlanmıştır günümüzde ise Python Yazılım Vafkı gönüllüleri tarafından geliştirilmektedir. En son sürümü 3.10.0'dır. (`https://www.python.org/downloads/windows/`). Üçüncü sürümden önceki sürüm ikinci sürümdür ve bu iki versiyon birbiri ile uyumlu değildir.

Python dili; üst seviye dillerden biri olması, nesne tabanlı olması, zengin kütüphane desteği olması, uygulama geliştirme, kullanıcı ara yüzü geliştirme, veri analizi gibi farklı alanlarda kullanılabilmesi gibi açılardan avantaj sağlamaktadır (Reddy vd., 2018). Python dilinin aynı zamanda, kolay okunabilir olması, farklı işletim sistemlerinde çalışabilmesi ve geniş bir topluluk desteğine (`https://www.python.org/community/`) sahip olması gibi sebeplerle ilgi çektiği söylenebilir.

Özellikle kurumlarda; araştırma, prototip hazırlama ve yeni fikirlerin uygulanması için veri analizinde SAS veya R gibi hazır program veya spesifik programlama dilleri kullanılırken bu fikirlerin daha büyük sistemlerin bir parçası olması amacıyla Java, *C#* veya C++ gibi dillerin kullanıldığı görülmekte ancak Python dilinin bu durumun aşılması için kullanılabildiği belirtilmiştir (McKinney, 2018). Yani Python hem araştırma hem de sistem geliştirilmesi aşamalarında kullanılabildiği için araştırmacılara avantaj sağlamaktadır. Özellikle hem akademi hem de sektör için veri bilimi, makine öğrenmesi ve yazılım geliştirme alanlarında en önemli dillerden biri haline gelmiştir (McKinney, 2018). Statista (2021) tarafından yayımlanan araştırmada 2021 yılında, programcılar tarafından en çok kullanılan diller arasında üçüncü sırada Python yer almaktadır (Şekil 3).



Şekil 5.3: En çok kullanılan programlama dilleri (Statista, 2021)

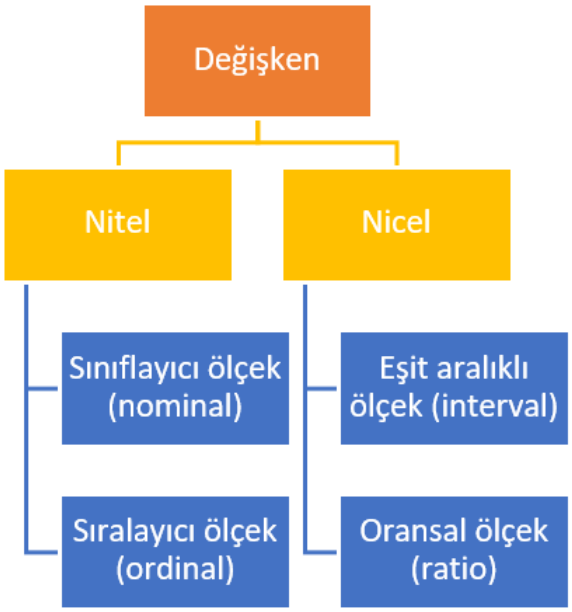
Python dili ile çalışabilmek için kullanılabilecek farklı tümleşik geliştirme ortamları (Integrated Development Environment – IDE) bulunmaktadır. Python resmi editörü, Integrated Development and Learning Environment (IDLE) olmakla beraber PyCharm, Visual Studio, Atom, JuPyter, Spyder, Eclipse gibi farklı editörler de mevcuttur. Bir diğer alternatif ise Google tarafından sağlanan Colaboratory (colab) (https://colab.research.google.com) uygulamasıdır. Colab web tabanlı olarak çalışan ve bir Gmail hesabına sahip herkesin kullanabileceği bir Python editörüdür. Hem kod yazılmasını hem de bu kodların web tarayıcısında çalıştırılmasına imkan sağlar. Colab; özellikle programlama derslerinde kullanılabilecek, ortak olarak yürütülen çevrimiçi çalışmalarda, interaktif çalışmaya ve kod paylaşmaya imkan sağlayan faydalı bir uygulamadır. Aynı zamanda Google Drive üzerinde klasör oluşturulmasını ve kod dosyalarının bulut depolama alanında tutulmasını sağlar.

Günümüzde özellikle veri analizi diyince akla gelen en popüler kavramlardan biri olan "veri bilimi" sözcüğünün "istatistik" sözcüğü ile bir süre daha yan yana kullanacağı ve bu iki kavramın birbirine bağımlı olduğunu belirtilmiştir (Gürsakal, 2016). Yapay zekâ, makine öğrenmesi, veri madenciliği gibi birçok yöntem farklı alanlarda kullanılırken bu yöntemlerin temelinde aslında istatistiksel yöntemler yatmaktadır. Veri analizinde kullanılacak en ileri yöntemlerin kullanımından önce temel istatistiksel yöntemler verinin anlaşılmasını sağlayabilir. Temel istatistiksel analizler veriyi anlama aşamasında araştırmacıların temel olarak başvurması gereken hesaplamaları ve görselleştirmeleri sunar. İstatistik kavramı; veri toplama, analiz, yorumlama bilimi olarak tanımlanmakta, değişkenler arasında ilişkilerin anlaşılmasını, açıklanmasını, mevcut ilişkilerden yola çıkarak tahmin ve kontrol etme amaçlarının yerine getirilmesini sağlamak olarak açıklanmaktadır (Gürsakal, 2016). Betimsel istatistik sayesinde mevcut ortalama, yüzde, görselleştirmeler gibi farklı temel yöntemlerle veri setinin genel yapısı ortaya konurken bu amaç doğrultusunda merkezi eğilim ölçüleri ve/veya dağılma ölçülerinden yararlanılır (Weiss, 2017).

Veriyi anlayabilmek için araştırmacının çalıştığı veri türlerine hâkim olması gerekir. Hangi algoritmanın hangi tür veri ile çalıştığını bilmek analiz aşamasında önem teşkil etmektedir. Herhangi bir veri seti ile yapılan her tür çalışmada araştırmacının hangi analizleri veya test yöntemlerini yürütebileceğine karar vermesi, çalıştığı veri setinin yapısını kavradıktan sonra mümkün olur. Yani veri setini oluşturan değişkenlerin genel yapısının anlaşılması doğru analizlerin seçilmesi konusunda araştırmacıya kolaylık sağlar.

Araştırmada incelenen birimlerin çeşitli özellikleri olarak tanımlanan değişken (Demir, 2020) farklı durumlarda değişik değerler alabilmektedir (Gürsakal, 2016). Değişkenlerin aldığı değerlere ise veri denilir (Weiss, 2017). Temel olarak bakıldığında bir araştırmada kullanılacak değişkenler sayısal (nicel) veya kategorik (nitel) özellikte olabilirler. Değişkenlere dair yapılan tanımlamalar Şekil 4'teki görselde verilmiştir (Demir, 2020; Gürsakal, 2016; Weiss, 2017):

- Nitel değişken (Qualitative variable): Kategorik, sözel olarak ifade edilebilen, sayısal olmayan değişkenlerdir. Bu değişkenlerde sıfır noktasının bir anlamı yoktur.
  - Sınıflayıcı ölçek (nominal): Sayılar arasındaki sıranın veya sayılar arasındaki uzaklığın bir anlamı yoktur; belli bir özelliğe göre sınıflandırma veya gruplandırma için yapılan ölçme işlemlerinde bu ölçekten yararlanılır. (cinsiyet, göz rengi)
  - Sıralayıcı ölçek (ordinal): Sayı sırasının bir anlamının olduğu, sayılar arasındaki uzaklığın bir anlamının olmadığı, küçükten büyüğe veya büyükten küçüğe sıralanabilen değerlerin ölçümünde bu ölçek kullanılır. (Eğitim durumu, sınıf, gelir düzeyi)
- Nicel değişken (Quantitative variable): Sayısal olarak ifade edilen değişkenlerdir. Tam sayı değerler (sayılabiliyorlarsa) kesikli (discrete), ondalıklı değerler (ölçülebiliyorlarsa) sürekli (continuous) nicel değişken olarak adlandırılır. (yıl (sayılabilir), kilo (ölçülebilir))
  - Eşit aralıklı ölçek (interval): Sayı sırasının ve sayılar arasındaki uzaklığın bir anlamının



Şekil 5.4: Değişken türleri

olduğu, mutlak sıfır noktasının olmadığı ölçektir. Her bir gruptaki aralık miktarı eşittir. (sıcaklık)

– Oransal ölçek (ratio): Sayı sırasının ve sayılar arasındaki uzaklığın bir anlamının olduğu, mutlak sıfır noktasının olduğu, değişkenin gerçek miktarını yansıtan ölçektir. (aylık gelir, yaş)

Tablo 5.1'deki Python uygulamasında değişkenlerin nicel veya nitel olma durumları incelenmiştir. Nicel veriler tam sayı (integer) ve ondalıklı sayı (float) şeklinde nitel veriler ise söz dizisi (string) olarak ifade edilirler. Veri türlerinin anlaşılabilmesi için herhangi bir kütüphaneye ihtiyaç yoktur. Python içerisindeki print ve type fonksiyonları ile bir değişken ekrana yazdırılabilir ve bunun türü incelenebilir. Burada, Amerikan sisteminde olduğu gibi, ondalık ayracının "." işareti olduğuna dikkat edilmelidir.

Tablo 5.1: Python Uygulama 1

Komut	**Ekran Çıktısı**
print(19) print(10.5) print('A') print("Python")	19 10.5 A Python
type(10) type(10.5) type('A') type("Python")	int float str str

Verilerin sunulmasında görselleştirme araçları kullanılabilir. Görselleştirme araçları verilerin sunulmasını, anlaşılmasını ve özetlenmesini sağlar. Bu araçlar için matplotlib veya seaborn kütüphanelerinden yararlanılabilir. Bu çalışmada matplotlib ile yapılmış örneklere yer verilmiştir. Bu kütüphaneler ile ilgili dokümantasyon ve detaylı bilgilere `https://matplotlib.org/` ve `https://seaborn.pydata.org/` linklerinden ulaşılabilir. Belli başlı grafik tiplerinin tanımları aşağıda verilmiştir (Weiss, 2017):

- Sütun/çubuk grafik (bar chart): Çubuk grafik, nitel verilerin farklı değerlerini bir eksende ve bu değerlerin frekanslarını veya yüzdelerini diğer eksende görüntüler. Bu grafik tipinde çubuklar birbirleriyle kesişmeyecek şekilde yerleştirilirler.
- Çizgi grafik (line chart): Çizgi grafik, nicel verilerin bir eksende nicel veya nitel değerlerin de diğer eksende verildiği ve her bir değerin bir çizgi yardımıyla birleştirildiği grafik türüdür (McKinney, 2018).
- Pasta grafiği (pie chart): Pasta grafiği, disk biçimde olup nitel verilerin frekanslarıyla orantılı olarak bölünmüş alanlardan oluşan bir grafik türüdür.
- Histogram: Histogram, nicel verilerin sınıflarını bir eksende ve bu sınıfların frekanslarını da diğer eksende görüntüler. Bu grafikteki çubuklar birbirine değecek şekilde yerleştirilir.

Tablo 5.2'de kullanılan import komutu ile matplotlib kütüphanesindeki pyplot modülü plt kısaltmasıyla programlama ortamına dahil edilir. Bu adımdan sonra ilgili modüldeki kullanılmak istenen fonksiyonlar plt. kısaltması yazılarak çağrılabilecektir. Böylelikle her seferinde kütüphaneyi çağırmak yerine kısaltma kullanılmış olur (kütüphane çağrılmadığında komutlar çalışmaz). Tabloda adı geçen show fonksiyonu grafiğin ekranda görüntülenmesini sağlar; title grafiğin başlığını belirlemek için, xlabel yatay eksendeki değişkenin ismini, ylabel ise dikey eksendeki değişkenin ismini belirlemek için kullanılır.

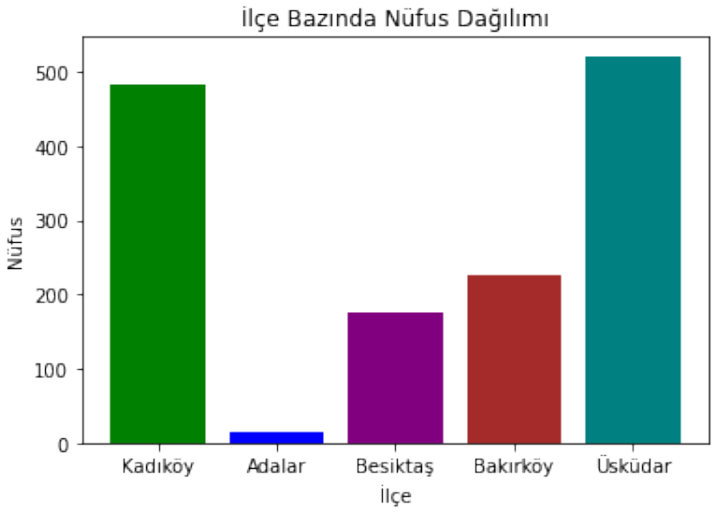
Sütun/çubuk grafik oluşturmak için matplotlib kütüphanesindeki bar fonksiyonundan yararlanılır. color parametresi ise burada sütunların renklerini değiştirmek amacıyla kullanılmıştır. renkler isimli değişkende istenen renk isimleri tutulmaktadır. Tablo 5.2'deki örnekte dikey eksende nüfus miktarı, yatay eksende ise ilçe isimlerinin yer alması istenmektedir.

Tablo 5.2: Python Uygulaması 2

Komut	**Ekran Çıktısı**
import matplotlib.pyplot as plt	n/a

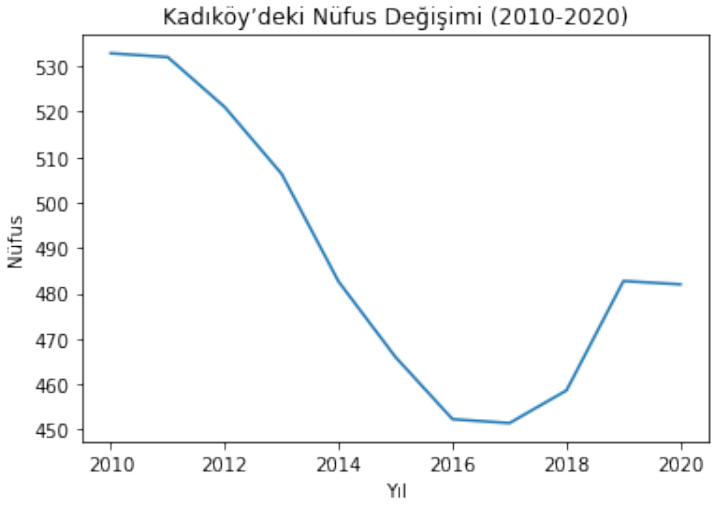
#sütun/çubuk grafik
ilce = ['Kadıköy','Adalar','Besiktaş','Bakırköy','Üsküdar']
nufus = [481.983,16.033,176.513,226.229,520.771]
renkler = ['green','blue','purple','brown','teal']
 	
plt.bar(ilce, nufus, color = renkler)
plt.title('İlçe Bazında Nüfus Dağılımı')
plt.xlabel('İlçe')
plt.ylabel('Nüfus')
plt.show()
	Şekil 5

Çizgi grafik oluşturmak için matplotlib kütüphanesindeki plot fonksiyonundan yararlanılır. Tablo 5.3'teki örnekte dikey eksende nüfus miktarı, yatay eksen ise yılların yer aldığı görülmektedir. Yıl değerleri sayısal değer olarak verilirse ilk grafik, kategorik değer olarak verilirse de ikinci grafik elde



Şekil 5.5: Çubuk grafik

edilecektir.



Şekil 5.6: Çizgi grafik – 1

Pasta grafiği oluşturmak için matplotlib kütüphanesindeki pie fonksiyonundan yararlanılır. Tablo 5.4'teki örnekte vurgu isimli değişkende öne çıkarılmak istenen değişkenlerin indis sıralarına göre birer değer verilmiştir. Kadıköy örneği pasta grafikte vurgulanmak istendiğinden ilk sıradaki değer 0,2 olarak belirlenmiş diğerlerine ise 0 değeri verilmiştir. labels değişkeni istenen ilçe isimlerinin pasta grafikte görüntülemesini sağlarken ilk parametre olan y bu değerlere ait sayısal değerleri tutmaktadır. autopct parametresi ise yüzde değerlerinin grafik üzerinde görüntülenmesini sağlarken kaç basamağın yüzdelik dilimde gösterileceğinin belirtilmesine imkân verir.

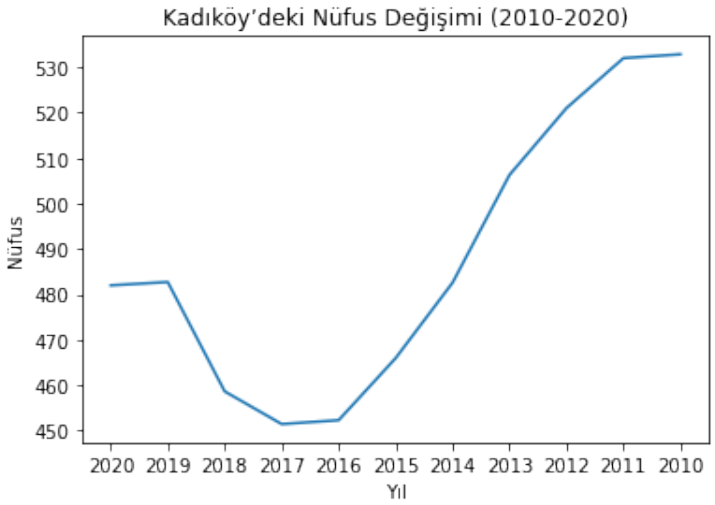
Histogram oluşturmak için matplotlib kütüphanesindeki hist fonksiyonundan yararlanılır. Tablo 5.5'teki örnekte dikey eksende frekans değerleri, yatay eksende ise yaş aralıkları yer almaktadır.

Tablo 5.3: Python Uygulaması 3

Komut	**Ekran Çıktısı**

#çizgi grafik
nufus = [481.983,482.713,458.638,451.453,452.302,
465.954,482.571,506.293,521.005,531.997,532.835]
yil = [2020,2019,2018,2017,2016,
2015,2014,2013,2012,2011,2010]
plt.plot(yil, nufus)
plt.xlabel('Yıl')
plt.ylabel('Nüfus')
plt.title("Kadıköy'deki Nüfus Değişimi (2010-2020)")
plt.show()
	Şekil 5.6

import matplotlib.pyplot as plt nufus = [481.983,482.713,458.638,451.453,452.302, 465.954,482.571,506.293,521.005,531.997,532.835] yil = ["2020","2019","2018","2017","2016", "2015","2014","2013","2012","2011","2010"] plt.plot(yil, nufus) plt.xlabel('Yıl') plt.ylabel('Nüfus') plt.title("Kadıköy'deki Nüfus Değişimi (2010-2020)") plt.show()	Şekil 5.7



Şekil 5.7: Çizgi grafik – 2

Tablo 5.4: Python Uygulaması 4

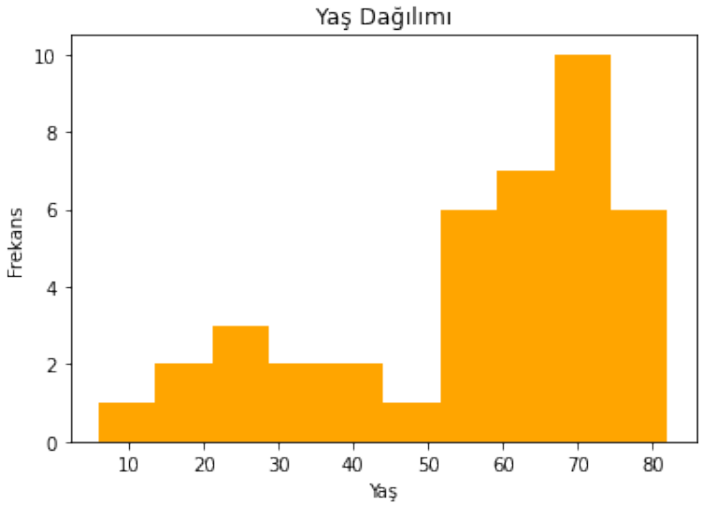
Komut	**Ekran Çıktısı**

#pasta grafiği
vurgu = [0.2, 0, 0, 0, 0]
ilce = ['Kadıköy','Adalar','Besiktaş','Bakırköy','Üsküdar']
nufus = [481.983,16.033,176.513,226.229,520.771]
plt.pie(nufus, labels = ilce,
explode = vurgu,
autopct='%2.2f%%')
plt.title('İlçe Bazında Nüfus Dağılımı')
plt.show()
	Şekil 5.8

Şekil 5.8: Pasta grafiği

Tablo 5.5: Python Uygulaması 5

Komut	**Ekran Çıktısı**
#histogram import matplotlib.pyplot as plt yas = [70, 40, 65, 63, 82, 63, 18, 57, 39, 63, 16, 28, 22, 76, 67, 73, 72, 73, 71, 62, 76, 57, 32, 22, 77, 35, 65, 59, 58, 70, 73, 69, 59, 75, 73, 63, 6, 81, 46, 59] plt.hist(yas, color='orange') plt.title("Yaş Dağılımı") plt.xlabel("Yaş") plt.ylabel("Frekans") plt.show()	Şekil 5.9



Şekil 5.9: Histogram

Grafiklerin dışında verilerin özetlenmesinde merkezi eğilim ölçüleri kullanılabilir. Bunlar; aritmetik ortalama, mod ve medyandır. Merkezi eğilim ölçülerinin hesaplanması için statistics kütüphanesinden yararlanılabilir. Bu kütüphane ile ilgili dokümantasyon ve detaylı bilgilere `https://docs.python.org/3/library/statistics.html` linkinden ulaşılabilir. Merkezi eğilim ölçülerinin tanımları aşağıda verilmiştir (Weiss, 2017):

- Aritmetik ortalama (mean): Bir veri setindeki gözlem değerlerinin toplamının gözlem değerlerinin sayısına bölünmesi ile elde edilir. Ortalama ile aynı şeyi ifade eder.
- Ortanca değer/medyan (median): Küçükten büyüğe sıralı bir dizide yer alan dizide ortadaki değerdir. Tek sayıda elemana sahip, sıralanmış bir dizide ortanca değer ortadaki değer iken çift sayıda elemana sahip bir dizide ortadaki iki değer ortanca değer olarak kabul edilir.
- Mod (mode): Veri setinde en sık tekrar eden değeri ifade eder.

Tablo 5.6'daki kodda görülen import komutu ile statistics kütüphanesi Colab ortamına dahil edilir. Bu adımdan sonra gerekli fonksiyonlar fonksiyon isminden önce st. kısaltması yazılarak çağrılırlar. Böylelikle her seferinde kütüphaneyi çağırmak yerine kısaltma kullanılmış olur. Bir sayı dizisinin aritmetik ortalamasını hesaplamak için statistics kütüphanesindeki mean fonksiyonundan yararlanılır. Örnekteki sayı dizisinin aritmetik ortalaması 5'tir.

Tablo 5.6: Python Uygulaması 6

Komut	**Ekran Çıktısı**
import statistics as st	n/a

#aritmetik ortalama
sayilar = [1,2,3,5,7,9,8,2,7,6]
st.mean(sayilar)
	5

Bir sayı dizisinin ortanca değerini (medyanını) hesaplamak için statistics kütüphanesindeki median fonksiyonundan yararlanılır. Tablo 5.7'deki örnekte, tek sayıda eleman içeren sayı dizisinin ortanca değeri 4'tür. Ayrıca print fonksiyonu ile dizinin küçükten büyüğe sıralanması sağlan-

mıştır. Bu şekilde 4 rakamının dizinin sıralandığı durumda en ortadaki sayı olduğu kolayca tespit edilebilmektedir. Çift sayıda eleman içeren sayı dizisinin ise iki adet ortanca değeri vardır. Örnekte görüldüğü üzere bu iki değer, dizinin küçükten büyüğe sıralandığında görülebilir. Bu durumda 5 ve 6 değerleri indis değerlerini ifade ederler. 5. indisteki 4 sayısı ve 6. indisteki 6 sayısı bu dizinin ortanca değerleridir.

Tablo 5.7: Python Uygulaması 7

Komut	**Ekran Çıktısı**

#ortanca değer
sayilar = [1,2,3,8,4,9,6,2,10] #tek sayıda eleman
sirali = sorted(sayilar)
print(sorted(sayilar))
st.median(sayilar)
	[1, 2, 2, 3, 4, 6, 8, 9, 10]
	4

#ortanca değer
sayilar = [1,2,3,8,4,9,6,2,10,12] #çift sayıda eleman
sirali = sorted(sayilar)
print(len(sirali))
med1 = (len(sirali)/2)
med2 = (len(sirali)/2)+1
print(med1)
print(med2)
	[1, 2, 2, 3, 4, 6, 8, 9, 10, 12]
	5.0
	6.0

Bir sayı dizisinin modunu hesaplamak için statistics kütüphanesindeki mode fonksiyonundan yararlanılır. Tablo 5.8'deki sayı dizisinde en sık tekrar eden rakam 7'dir. Rakam dizisi yerine harflerden oluşan bir dizide ise en sık tekrar eden harf geri döndürülmektedir. Tablo 5.8'deki harf dizisinde frekansı en fazla olan yani en sık tekrar eden harf 'B' harfidir.

Tablo 5.8: Python Uygulaması 8

Komut	**Ekran Çıktısı**

#mod
sayilar = [1,2,3,5,7,9,7,2,7,6]
st.mode(sayilar)
	7

harfler = ['A','B','b','B','A','B'] st.mode(harfler)	'B'

Verilerin özetlenmesinde dağılım ölçüleri de kullanılabilir. Bunlar; yayılma bandı, standart sapma ve varyans hesaplamalarıdır. Merkezi eğilim ölçülerinin hesaplanması için statistics veya math kütüphanelerinden yararlanılabilir. Bu kütüphaneler ile ilgili dokümantasyon ve detaylı bilgilere https://docs.python.org/3/library/statistics.html ve https://docs.python.org/3/library/math.html linklerinden ulaşılabilir. Dağılma ölçülerinin tanımları aşağıda verilmiştir (Weiss, 2017):

- Yayılma bandı (range): Veri setindeki gözlem değerlerinden en küçük ve en büyük eleman arasındaki farktır.
- Standart sapma (standard deviation): Veri setindeki gözlemlerin, aritmetik ortalamadan, ortalama olarak ne kadar uzakta olduğunu gösterir.
- Varyans (variance): Standart sapmanın karesidir.

Yayılma bandının hesaplanması için özel bir fonksiyon kullanılmamaktadır. min ve max fonksiyonları Python ile birlikte gelen temel fonksiyonlardandır. Bunlar kullanılarak iki değer arasındaki fark hesaplanabilir. Tablo 5.9'da sırasıyla en küçük değer, en büyük değer ve aralarındaki fark ekrana yazdırılmıştır. Bu şekilde yayılma bandı, yani dizideki sayıların hangi aralıkta yer aldıkları görülebilir.

Tablo 5.9: Python Uygulaması 9

Komut	**Ekran Çıktısı**

#yayılma bandı
sayilar = [1,2,3,5,7,9,7,2,7,6]
print(min(sayilar))
print(max(sayilar))
print(max(sayilar)-min(sayilar))
	1
	9
	8

Standart sapma ve varyansın hesaplanması için import komutu ile statistics kütüphanesi Colab ortamına dahil edilir. Bu adımdan sonra ilgili fonksiyonlar fonksiyon isminden önce st. kısaltması yazılarak çağrılır. Böylelikle her seferinde kütüphaneyi çağırmak yerine kısaltma kullanılmış olur. Bir sayı dizisinin standart sapmasını ve varyansını hesaplamak için statistics kütüphanesindeki stdev fonksiyonundan yararlanılır. Tablo 5.10'daki sayı dizisinin standart sapması 2,72, varyansı ise 7,43'tür.

Tablo 5.10: Python Uygulaması 10

Komut	**Ekran Çıktısı**
import statistics as st	n/a

#standart sapma
sayilar = [1,2,3,5,7,9,7,2,7,6]
st.stdev(sayilar)
	2.7264140062238043

#varyans
sayilar = [1,2,3,5,7,9,7,2,7,6]
st.variance(sayilar)
	7.433333333333334

math kütüphanesindeki fonksiyonların kullanıldığı durumlar için ise bu kütüphanenin, çalışma ortamına dahil edilmesi gerekir. math kütüphanesindeki pow (üs alma) ve sqrt (karekök) fonksiyonlarından standart sapma ve varyans hesaplamalarının sağlamasının yapılmasında yararlanılabilir. Tablo 5.11'deki dizinin varyansının karekökü standart sapmasını vermektedir. Sonuç olarak yine sırasıyla 2,72 ve 7,43 değerleri elde edilmiştir.

Tablo 5.11: Python Uygulaması 11

Komut	**Ekran Çıktısı**
import math	n/a

#standart sapma
sayilar = [1,2,3,5,7,9,7,2,7,6]
math.sqrt(st.variance(sayilar))
	2.7264140062238043

#varyans
sayilar = [1,2,3,5,7,9,7,2,7,6]
math.pow(st.stdev(sayilar),2)
	7.4333333333333345

5.4 Kaynaklar

A Dictionary of Computing. (2008). A Dictionary of Computing (A. Butterfield, G. E. Ngondi, & A. Kerr (ed.)). A Dictionary of Computing; Oxford University Press. https://doi.org/10.1093/acref/9780199234004.001.0001

Çobanoğlu, B. (2019). Herkes İçin Python (2. baskı). Pusula Yayıncılık. http://files/37010/478372.html

Demir, İ. (2020). SPSS ile İstatistik Rehberi. Efe Akademi Yayınevi.

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). The KDD Process for Extracting Useful Knowledge from Volumes of Data. Communications of the ACM, 39(11), 27–34.

GilPress. (2016). Data Scientists Spend Most of Their Time Cleaning Data. https://whatsthebigdata.com/2016/05/01/data-scientists-spend-most-of-their-time-cleaning-data/

Gürsakal, N. (2016). R ile Betimsel İstatistik (2. baskı). Dora Yayıncılık.

McKinney, W. (2018). Python for Data Analysis: Data Wrangling with Pandas, Numpy, and IPython (2. baskı). O'Reilly Media, Inc.

Reddy, K. P. N., Y, G., D, S., & S M, R. (2018). Comparison of Programming Languages: Review. International Journal of Computer Science & Communication, 9(2), 113–122.

Statista. (2021). Most used programming languages among developers worldwide, as of 2021. https://www.statista.com/statistics/793628/worldwide-developer-survey-most-used-languages/

Weiss, N. A. (2017). Introductory Statistics (10th Globa). Pearson Education Limited.

Eğitsel Veri Madenciliğinde Sınıflandırma Algoritmaları ile Tahmin Modeli Geliştirme

Kadir Burak OLGUN*, Sevinç GÜLSEÇEN†

***Öğr. Gör. Dr., Bursa Uludağ Üniversitesi, Orhangazi Yeniköy Asil Çelik MYO, Bilgisayar Programcılığı, Bursa, Türkiye, E-mail: kburakolgun@uludag.edu.tr**

†Prof. Dr., İstanbul Üniversitesi, Fen Bilimleri Enstitüsü, Enformatik Anabilim Dalı, İstanbul, Türkiye, E-mail: gulsecen@istanbul.edu.tr

6.1 Giriş

Verinin öneminin artışındaki etkenlerden biri de insanların tıpkı doğada bıraktıkları ekolojik ayak izleri gibi dijital ortamda bıraktıkları ayak izlerinin keşfedilmesidir. Bu keşif ile dijital ortamda birikmiş olan veriler kullanılarak nasıl bir yarar sağlanabilir sorusunun üzerine gidilmiş, veri madenciliği kavramının yükselişi ivmelenmiştir. Elde edilen veriler birçok alanda verimliliğin artırılması, gelecek hakkında tahminde bulunabilme, ihtiyaçları saptayabilme gibi farklı amaçlar için kullanılırken, eğitim alanında da bir takım verilerin elde edildiği ve bu veriler ile ne gibi analizlerin gerçekleştirilerek ne tür faydalar sağlayabileceği merak konusu olmuştur. Eğitim ortamlarından, öğrenci ve eğiticilerden elde edilen veriler ile araştırmacılar artık öğrenme süreçlerinin ve ortamlarının iyileştirilmesi, süreç içerisindeki kaynakların doğru yönetimi ve öğrenci ya da öğretici tarafında ileriye dönük tahminlerin yapılması konusunda ilerleme göstermektedir. Özellikle öğrencilerin okulu bırakma ve akademik başarı tahminleri üzerine yapılan çalışmaların çokluğu eğitsel veri madenciliğinin gelişiminde büyük rol oynamıştır. Öğrencilerin akademik başarılarının tahmin edilebilmesi ile geliştirilebilecek erken uyarı sistemlerinin sunacağı yorumlar ile öğrenme sürecinin iyileştirilmesi bu alandaki en büyük hedeflerden biridir.

6.2 Eğitsel Veri Madenciliği Nedir?

Öğrenme ortamlarının dijitalleşmesi ile çoğunlukla öğrencilere ait bir takım verilerin ortaya çıkması veri madenciliği yöntem ve tekniklerinin eğitim alanında da uygulanmaya başlamasını tetiklemiştir. Veri madenciliğinin eğitimde kullanılmaya başlanması diğer alanlardaki kullanımına göre nispeten yeni sayılabilir. Kurumların elindeki öğrenciler hakkındaki veriler artmaya başladıkça büyük veri havuzları oluşmuş ayrıca eğitimin İnternet ortamına taşınmaya başlaması ile dijital ortamlardan öğrenme-öğretme etkileşim verileri de elde edinmeye başlanmıştır. Tüm bunların bir etkisi olarak da araştırmacıların ve eğitimcilerin arasında önemini günden güne artıran ve eğitim kalitesinin iyileştirilmesine katkı sağlayacak bulguların elde edilmesini hedefleyen eğitsel veri madenciliği kavramı kullanılmaya başlanmıştır (Güldal & Çakıcı, 2017; Ktona, ve diğ., 2014; Romero & Ventura, 2007).

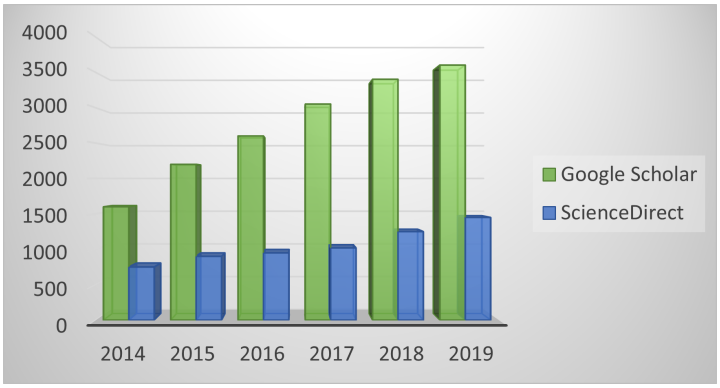
Eğitsel veri madenciliği, eğitim sürecindeki her ortamdan toplanan öğrenci, eğitici ya da sürecin kendisinden elde edilen eşsiz veriler kullanılarak pedagojik bir yaklaşım ile veri madenciliği tekniklerinin kullanılmasıyla eğitim problemlerine çözüm arayan yeni tekniklerin geliştirilmesini hedefleyen çok disiplinli bir araştırma alanı olarak tanımlanabilir. Eğitsel veri madenciliğinde kullanılan verilerin benzersizliği elde edilen sonuçlara da yansıyacağından sorunsuz ve verimli bir öğrenme ortamı için bu sonuçların pedagojik yaklaşımlarla yorumlanması ve iyileştirmelerin gerçekleştirilmesi uygun olacaktır. Veri madenciliğinin bir çok alana uygulandığı gibi eğitime de uygulanması çok farklı olarak düşünülmese de eğitsel veri madenciliğini farklı kılan Romero ve Ventura (2007)'nın aşağıda açıkladığı amaç, kullanılan veriler ve uygulanan tekniklerdir.

- Amaç: Eğitsel veri madenciliğinin amacı diğer alanlardaki uygulamalara göre daha özneldir ve hassas ölçüm tekniklerine ihtiyaç duymaktadır.
- Veriler: Eğitim ortamlarından elde edilen verilerin türlerinde farklılıklar vardır ve bu verilerin eğitim alanına özgü veriler olması içsel anlamsal bilgilere, çok sayıda anlamlı hiyerarşik düzene sahip olduklarını gösterir. Dolayısıyla öğrencinin ve sistemin pedagojik yönlerinin de analize dahil edilmesi gerekmektedir.
- Teknikler: Eğitim verileri ve problemlerinin farklı özel niteliklere sahip oluşu geleneksel veri madenciliği tekniklerinin tümünün eğitim problemine uyarlanmasını engeller ve belirli eğitim sorunları için belirli tekniklerin kullanılması önerilmektedir.

Eğitsel veri madenciliği sadece farklı amaçlar güden, vizyon ve misyonları yönünden farklı açılardan bakan öğretmenlerden değil, kendi öğrenme süreçlerini anlamaya çalışan ya da geri bildirimler ile durum hakkındaki düşüncelerini desteklemeye çalışan öğrenciler gibi birçok gruptan oluşmaktadır. Ayrıca sınıf ortamı, İnternet tabanlı eğitim sistemleri, öğrenme yönetim sistemleri, dijital anketler, sınavlar, görsel-yazılı içerikler gibi ve artık eğitim ortamı olarak kullanılabilen sosyal ağlar, forumlar, sanal ortamlar, eğitici oyun ortamları gibi çeşitli eğitim sistemleri ya da ortamları birbirlerinden tamamen farklı veriler oluşturur. Bu veriler eğitsel veri madenciliğinde amaca yönelik çok farklı tekniklerin kullanılarak probleme çözüm aranmasını gerektirmektedir (Merceron & Yacef, 2005; Hanna, 2004; Romero & Ventura, 2010).

Araştırmacıların büyük bir çoğunluğu tarafından henüz yeni bir alan olarak görülen eğitsel veri madenciliği buna karşın giderek artan bir ilgi ile karşılaşıldığı Şekil 1 yorumlanarak söylenebilir.

Tüm bu gelişmeler ve yaklaşımlar ele alındığında veri madenciliği yöntemlerinin eğitimde kullanılması eğitsel veri madenciliği alanının ortaya çıkmasına sebep olmuştur.



Şekil 6.1: Yıllara göre Eğitsel Veri Madenciliği alanındaki çalışma sayıları.

6.3 Verilerin Elde Edilmesi

Eğitsel veri madenciliği çalışmalarında karşılaşılan en büyük problem verilerin elde edilmesidir denebilir. Özellikle öğrencilere ait demografik veriler dışında kalan tüm veriler verimli ve doğru sonuçlar elde etmek için oldukça uzun süreli ve detaylı bir biçimde toplanmalıdır. Verilerin araştırmacı tarafından elde edilmesi dışında var olan dijital öğrenme ortamlarından daha önce kayıt altına alınmış verilerin çekilmesi de mümkündür. Bu alanda kullanılabilecek veriler daha önceki başlıkta belirtildiği gibi bir çok farklı biçimde ve farklı ortamlardan elde edilebilir fakat çalışmalarda kullanılan veriler aşağıdaki özelliklerine göre gruplanabilir (Bousbia & Belamri, 2014):

1. Veri Kullanılabilirliği:
   (a) Kurum veritabanları ya da öğrenme ortamları yazılımlarında kaydedilen veriler mevcuttur.
   (b) Araştırmalar sırasında üretilen veriler kullanılabilir.
   (c) Karşılaştırmalı veri havuzlarında veriler mevcuttur.
2. Kaynaklar:
   (a) Öğrencilerin sınıf içi faaliyetleri hakkında notlar alan gözlemcilerin elde ettiği veriler.
   (b) Öğrencilerin faaliyetlerini dijital ortamda kaydeden yazılımların elde ettiği veriler.
   (c) Hem dijital ortamdan elde edilen hem de alınan notlardan elde edilen veriler.
3. Öğrenme Ortamı:
   (a) Geleneksel eğitim.
   (b) Dijital ortamlarda saklanabilir veri üreten bilgisayar tabanlı eğitim.
4. Veri Türleri:
   (a) Nitel ya da nicel veriler.
   (b) Kişisel ve / veya demografik veriler.
   (c) Uygulanan anketlere verilen cevaplar.
   (d) Eğitim ortamı içerisinde bilgisayar desteği ile elde edilen bireysel etkileşimler.
   (e) Sosyal etkileşimler.
   (f) Yüz ve vücut reaksiyonları.

Tahmine dayalı modeller oluşturmak için öğrenme yönetim sistemlerinden gelen veriler ve klasik öğrenci yönetim sistemlerinden gelen demografik ve bilişsel bazı veriler kullanılır fakat kurumlardaki bazı öğrenci bilgilerinin gizliliği ve etik kaygılar başarılı tahminler elde etmek için ihtiyaç duyulabilecek bir takım verilerin elde edilmesini imkânsızlaştırmaktadır (Yu, et al., 2020). Eğitsel veri madenciliğinde verimli ve daha doğru tahminler yapabilmek için ihtiyaç duyulan tüm

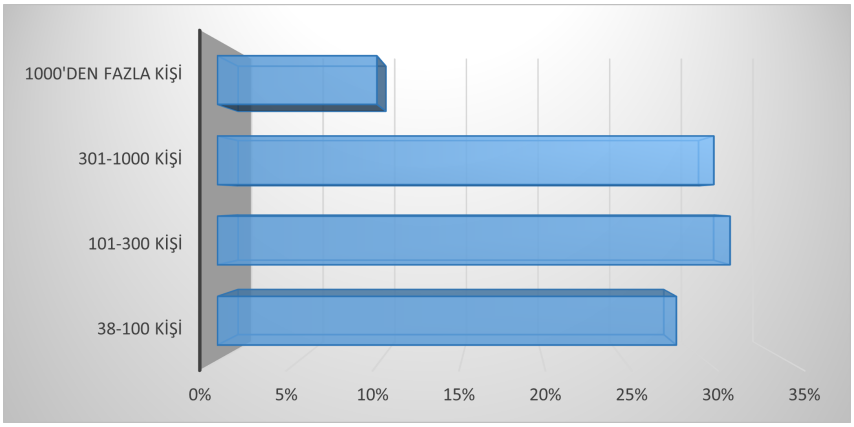
verilere ulaşmak etik kaygılar nedeniyle çoğu zaman mümkün olmasa da araştırmacılar tarafından toplanan demografik bilgiler ile bu eksiklikler giderilmeye çalışılmaktadır.

Bu alanda veri toplarken ya da verilere erişim sağlarken karşılaşılabilecek zorluklar aşağıdaki gibi sıralanabilir:

- Kurum politikaları
- Etik değerler
- Kişisel verilerin korunması
- Duyuşsal veri sorunsalı
- Dengesiz veri setleri
- Düzensiz-süreksiz veriler
- Verilerin depolanması

Kurumların belirledikleri politikalar gereği ellerindeki verilere erişim izni vermemesi, etik kaygılar gereği öğrenciler hakkındaki anlık verilerin elde edilme sürecinin imkansızlığı(Çin'de tüm sınıflara ve koridorlara yerleştirilen kameralar ile duygu analizi çalışması etik değerler nedeniyle bir çok ülke tarafından kınanmıştır.), kişisel verilerin korunmasının bir çok ülkede kanunlaşması ile birlikte veri erişim izinlerinde yaşanan sorunlar, insanların duyguları vasıtasıyla elde edilen verinin anlık duygu değişimleri sebebiyle uzun vadede bozulması, özellikle tahmin analizlerinde herhangi bir gruba yığılımın fazla olması durumunda algoritmaların eğilimini o gruptan yana yapması, uzun süreli veri toplarken devam etmeyen, süreç içerisinde konumu ya da durumu değişen öğrencilerden düzenli ve sürekli olarak veri elde edilememesi ve son olarak bu alanda elde edilen verilerin farklı formatlarda olması sebebiyle büyük boyutlara ulaşarak depolama sorunlarının yaşanması eğitsel veri madenciliği alanında karşılaşılan zorlukları açıklamaktadır.

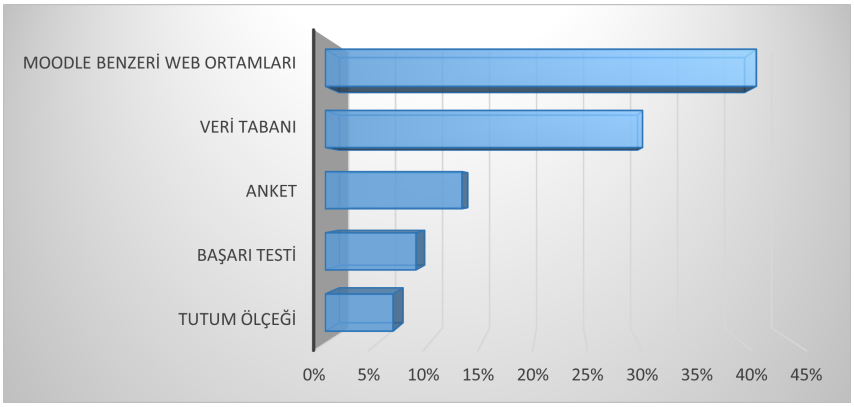
Tüm bu zorluklara rağmen araştırmacılar çalışmalarını gerçekleştirebilmek için veri toplama sürelerini kısaltmaya ve verilerin elde edildiği grupları küçültme eğilimine yönelmişlerdir. Şekil 6.2'de alandaki verilerin çoğunlukla kaçar kişilik gruplardan elde edildiği görülmektedir.



Şekil 6.2: Çalışma verilerinin satır sayısına göre yüzdelik dağılımı.

Şekil 6.2'de görüldüğü üzere alanda çoğunlukla 100-1000 kişi aralığında bir gruptan veri edilmektedir. Bu verilerin analiz aşaması düşünüldüğünde veri seti binlerce satırdan oluşmayacaktır fakat veri madenciliğinde genel olarak yapılan çalışmalarda binlerce hatta on binlerce satır veri setlerinin kullanılması olağan bir durumdur. Bu durumun sebebi daha önce bahsedilen zorluklarken, analiz aşamasında da eğitsel veri madenciliğinin daha küçük veri setleri ile doğru çıkarımlara erişmesinin zorunluluk olduğu anlamı çıkarılabilir.

Verilerin elde edilme şekli incelendiğinde ise yine karşılaşılan zorluklar ve analiz sonucu yapılan çıkarımların performansının artırılması amacıyla dijital ortamlardan elde edilen verilerin kullanıldığı söylenebilir. Şekil 6.3'te alandaki çalışmalardan verilerin nerelerden elde edildiği gösterilmiştir.



Şekil 6.3: Verilerin elde edildiği ortamlar.

Veri tabanları ve öğrenme yönetim sistemleri üzerinden elde edilen verilerin anketler ile toplanan demografik veriler, başarı testleri ya da tutum ölçekleri ile elde edilen verilere oranla daha yüksek performansta çıkarımlar gerçekleştirmeyi sağladığı görüldüğü için çoğunlukla bu iki ortamdan alınan veriler alanda tercih edilmektedir.

6.4 Eğitsel Veri Madenciliğinde Kullanılan Modeller, Algoritmalar ve Araştırma Konuları

Veri madenciliğinde kullanılan modeller tahmin edici ve tanımlayıcı modeller olarak iki ana başlık altında incelenebilir. Bu modeller bazı tekniklerden ve bu teknikler ise beraber de kullandıkları yöntemlerden oluşmaktadır. Tablo 6.1'de bu modellere ait bir takım teknikler ve bu tekniklerde kullanılan yöntemler gösterilmiştir.

Tablo 6.1: Veri madenciliği modelleri, yöntemleri ve teknikleri.

Tahmin Edici Modeller		Tanımlayıcı Modeller	
Sınıflandırma	Regresyon	Kümeleme	Birliktelik Kuralları
Karar Ağaçları		Bölme Yöntemleri	Apriori Algoritması
Yapay Sinir Ağları		Hiyerarşik Yöntemler	Carma Algoritması
Genetik Algoritmalar		Yoğunluk Tabanlı Yöntemler	Sequence Algoritması
K-En Yakın Komşu		Izgara Tabanlı Yöntemler	Eclat Algoritması
Naive-Bayes		Model Tabanlı Yöntemler	GRI Algoritması
Bellek Temelli Nedenleme			FP-Growth

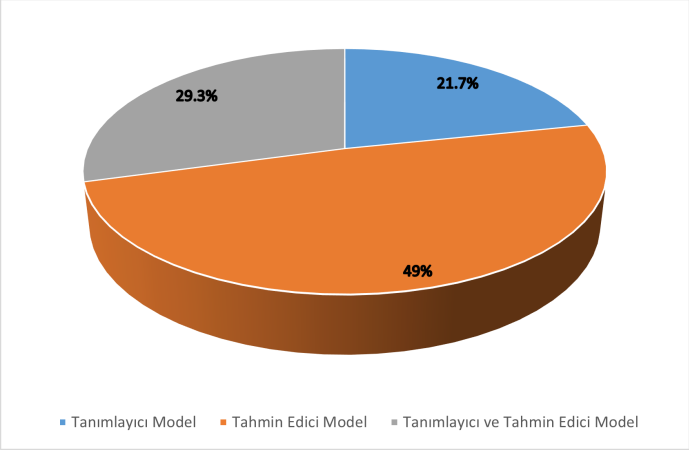
Tahmin edici teknikler olan sınıflandırma ve regresyon veri madenciliğinde elde olan verilerden yola çıkarak sonraki gelecek olan veriler hakkında öngörüde bulunma amacıyla kullanılır denebilir.

Bu şekilde tahminde bulunurken sınıflandırma kategorik, regresyon ise süreklilik gösteren veriler için kullanılmaktadır. Bu modeller için birçok teknik ya da algoritma bulunması kullanılan veri setlerine göre seçim yapılabilmesine imkan sağlarken aralarından en çok kullanılan yöntem kolay kullanımı sayesinde karar ağaçlarıdır. Karar ağaçları iki aşamalı olarak çalışmaktadır. Birinci aşamada var olan veri seti analiz edilerek algoritmanın öğrenme adımı gerçekleştirilir. İkinci aşamada ise algoritma bu öğrenme ile model geliştirir ve gelecek veri seti hakkında çıkarımda bulunur. İkinci aşamada modelin oluşturulması için 1. Adımda ortaya çıkan kurallar kullanılır ve bu kuralların çerçevesinde tahminler gerçekleştirilir. Örneğin bir veri setindeki bilgilere göre eğitilmiş modelin kasko poliçesi risk tahminleri incelendiğinde;

EĞER yas<25 VEYA yas>50 VE ehliyetYili<2 İSE riskDurumu = Yüksek

Şeklinde kurallar oluşturulur. Karar ağaçları bunlar gibi birçok kuralın artarda sıralanması ile dallanarak büyür ve her bir dallanma mutlaka bir sonuca çıkar. Sonuçlar bu modelde risk durumunun seviyesini belirtmektedir.

Şekil 6.4 ve Şekil 6.5'te alanda kullanılan modellerin ve tekniklerin yüzdelik dağılımları görülmektedir (Tekin & Öztekin, 2018).

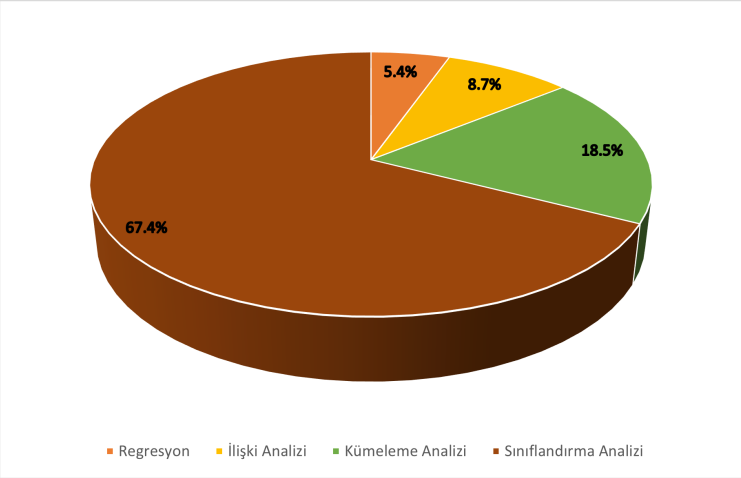


Şekil 6.4: Modellerin kullanım yüzdeleri.

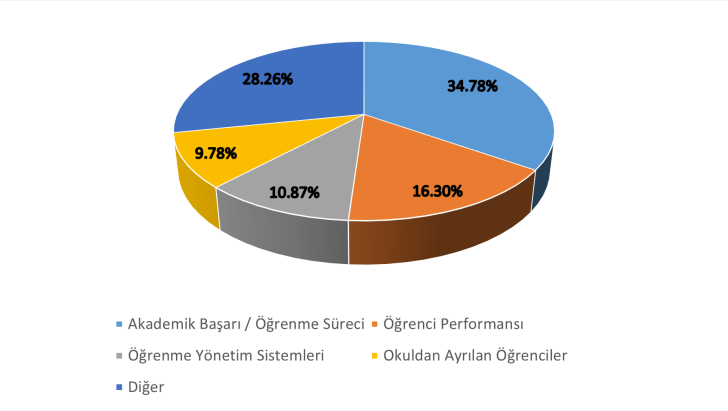
Alandaki çalışmaların neredeyse yarısında sadece tahmin edici model üzerinde, yaklaşık %80'nin ise tahmin edici modelin kullanıldığı araştırmaların gerçekleştirildiği görülmektedir.

Grafikler incelendiğinde eğitsel veri madenciliği çalışmalarının büyük bir kısmının tahmin edici modelin sınıflandırma tekniği kullanılarak gerçekleştirildiği söylenebilir. Yapılan araştırmaların konuları incelendiğinde ise Şekil 6.6'da görüldüğü üzere eğitsel veri madenciliği çalışmalarının yaklaşık yarısının öğrencilerin akademik performans ve başarı durumlarının tahmin edilmesi üzerine olduğu görülmektedir.

Öğrenci performansını tahmin etmeye yönelik gerçekleştirilen çalışmalarda yaygın olarak kullanılan sınıflandırma yöntemlerinden yüksek performans elde edilenler bir çok araştırmacı tarafından Lojistik Regresyon, Karar Ağaçları, K En Yakın Komşu, Destek Vektör Makineleri ve Rastgele Orman algoritmaları olarak gösterilirken son yıllarda Naive Bayes algoritmasının da çalışmalarda tercih edildiği görülmektedir. Burada listelenen algoritmalardan en iyisi herhangi biri olamaz, kullanılan verinin özelliklerine göre her algoritma farklı sonuçlar verebilmektedir. Bu sebeple araştırmalarda birden fazla algoritma denenerek daha iyi performans veren tercih edilir ve model geliştirilir.



Şekil 6.5: Tekniklerin kullanım yüzdeleri.



Şekil 6.6: Araştırma konularının yüzdelik dağılımları.

6.5 Sınıflandırma Nedir ve Modeller Nasıl Değerlendirilir?

Yu ve Diğerleri (2020), yükseköğrenimde yaptıkları çalışma ile tahmin analizlerinin yöneticiler, eğitmenler ve öğrenciler gibi eğitim alanındaki farklı paydaşlar için uygulanabilir bir takım iç görüler sağlayabileceğini ve bunun için öğrenme yönetim sistemlerinden elde edilen verilerin yeterli tahmin gücüne erişebileceğini belirtmişlerdir.

Eğitsel veri madenciliği öğrencilerin davranışlarını ve başarı performanslarını, alan bilgisi içerikleri, değerlendirme yöntemleri, eğitimsel fonksiyonlar ve uygulamalar gibi özellikleri karakterize etmek için kullanılan kalıplar oluşturmak ve tahminlerde bulunmak amacıyla kullanılmaktadır (Hicham, ve diğ., 2020). Eğitim alanında özellikle performansın tahmin edilebilir olması oldukça önemlidir, son yıllarda araştırmacıların öğrencilerin öğrenmesini analiz etmek, eğitim sürecinin kalitesini iyileştirmek ve geniş bir bakış açısı elde etmek gibi amaçlar için veri madenciliği tekniklerini kullanmaları kaçınılmaz olmuştur (Razak, ve diğ., 2018). Alandaki çalışmaların büyük bir kısmının tahmin etmeye yönelik olduğu düşünüldüğünde ve veri tipleri ele alındığında sınıflandırma algoritmalarının kullanıldığı görülmektedir.

Sınıflandırma anlamsız verileri sınıf etiketleri oluşturarak bir model oluşturmayı amaçlayan denetimli bir makine öğrenmesi tekniğidir. Sınıflandırma tahmin edici modelin bir tekniği olarak veri madenciliği ile elde edilen verilerden yola çıkarak sonraki gelecek olan veriler hakkında öngörüde bulunma amacıyla kullanılır. Sınıflandırmada kullanılacak verilerin kategorik olması gerekmektedir. Kategorik olmayan veriler için regresyon kullanılabilir.

Sınıflandırma yöntemlerinden karar ağaçları kullanım kolaylığı sebebiyle oldukça fazla tercih edilir ve ilk aşama olan algoritmanın eğitilmesinde eğitim veri seti kullanılır. Eğitim veri setinin dağılımının dengesiz olması yani tahmin edilecek herhangi bir sınıftan verinin diğerine göre daha çok olması algoritmanın o sınıfa yönelmesine neden olabilir. Bu sebeple kullanılan eğitim veri setinin dengeli bir sınıf dağılımında olması önem arz etmektedir. Eğitilen algoritma eğitim veri setinin içerisindeki bilgilerden yola çıkarak kurallar belirler ve model oluşturur. Daha sonra model test veri seti olarak sunulan yeni verileri önceki bilgilerden yola çıkarak oluşturulan kurallara uygun olarak sınıflandırmaya çalışır. Doğru sınıflandırma sayıları genel itibari ile modelin başarı performansını temsil eder denebilir.

Geliştirilen modellerin değerlendirilmesi için dikkate alınması gerek bir takım değerler vardır. Bu değerlerin bir kısmı kullanılan algoritmalar tarafından hesaplanıp sunulurken bir kısmının ise araştırmacı tarafından hesaplanması gerekmektedir. Sınıflandırma algoritmaları ile oluşturulan bir modelin değerlendirilmesinde aşağıdaki değerler göz önünde bulundurulmalıdır.

- Doğruluk oranı
- Duyarlılık
- Kesinlik
- Kappa
- MCC(Matthew Korelasyon Katsayısı)

Modelin çalıştırılması sonucu elde edilen hata matrisi üzerinden bu değerlerin Kappa hariç tümü hesaplanabilmektedir. Kappa değeri de hata matrisinin altında ayrıca hesaplanmış olarak verilmektedir. Hata matrisleri sınıflandırma algoritmalarının performansının doğrudan görülebildiği ve gerçek değerler ile tahmin edilen değerlerin karşılaştırıldığı tablolardır. Tablo üzerindeki tahmin sütunun altında gerçek satırı ile kesişen sınıfların sayıları verilmektedir. Örnek bir hata matrisi Tablo 6.2'de gösterilmiştir.

Örnek hata matrisi 2'li bir sınıflandırma yapan model için tasarlanmış buradaki DP: Doğru pozitif sayısı, YP: Yanlış pozitif sayısı, YN: Yanlış negatif sayısı, DN: Doğru negatif sayısı olarak

Tablo 6.2: Örnek hata matrisi görünümü.

Hata Matrisi		**Tahmin**	
		1	0
Gerçek	1	DP	YP
	0	YN	DN

temsil edilmiştir. Aşağıda maddeler halinde her bir durumun açıklamaları verilmiştir:

- DP: Veri setinde pozitif olan değerlerin, model tarafından doğru tahmin edildiği toplam sayı
- YP: Veri setinde pozitif olan değerlerin, model tarafından yanlış tahmin edildiği toplam sayı
- YN: Veri setinde negatif olan değerlerin, model tarafından yanlış tahmin edildiği toplam sayı
- DN: Veri setinde negatif olan değerlerin, model tarafından doğru tahmin edildiği toplam sayı

$$Doğruluk = \frac{DP + DN}{DP + YP + YN + DN} \tag{6.1}$$

Doğruluk oranı 0 ile 1 arasında bir değer alır ve yukarıdaki formüle göre doğru tahmin edilmiş örneklerin sayısının, toplam örnek sayısına bölünmesi ile hesaplanmaktadır. Çıkan değer yüzdelik olarak düşünülerek(örneğin; 0,75 = %75) yorumlanabilir.

$$Duyarlılık = \frac{DP}{DP + YP} \tag{6.2}$$

Duyarlılık değeri doğru tahmin edilmiş pozitif örneklerin sayısının, gerçekte pozitif olan toplam örnek sayısına bölünmesi ile hesaplanır ve bu değer de yüzdelik olarak düşünülmelidir.

$$Kesinlik = \frac{DP}{DP + YN} \tag{6.3}$$

Kesinlik değeri ise doğru tahmin edilmiş pozitif örneklerin sayısının, pozitif olarak tahmin edilmiş toplam örneklerin sayısına bölünmesi ile hesaplanırken yine çıkan değer yüzdelik olarak alınabilir.

Bu değerlerden duyarlılık ve kesinlik hesaplanırken tahmin edilen örneklerde önemli olan negatif değerler ise benzer formüller bu kez negatif değerler için dönüştürülerek uygulanabilir.

Kappa değeri bir tutarlılık ölçütü olarak tutarsız olan veriler için kalite ölçüm değeri olarak kullanılır. Fakat sınıflandırma performansının açıklanması için tek başına yeterlidir denemez ve duyarlılık, doğruluk gibi metrikler ile beraber değerlendirilmelidir. Kappa değeri 0 ile 1 arasında olur. 0.00 ile 0.20 aralığındaki Kappa değeri hafif, 0.21 ile 0.40 aralığındaki Kappa değeri makul, 0.41 ile 0.60 aralığındaki Kappa değeri orta, 0.61 ile 0.80 aralığındaki Kappa değeri sağlam, 0,81 ile 1.00 aralığındaki Kappa değeri ise neredeyse mükemmel olarak ifade edilmektedir (Landis & Koch, 1977).

Tüm bunların yanında modelin performansının ölçülmesi için ek olarak MCC değeri de hesaplanabilir.

$$MCC = \frac{DP * DN - YP * YN}{\sqrt{(DP + YP) * (DP + YN) * (DN + YP) * (DN + YN)}} \tag{6.4}$$

MCC değeri -1 ile 1 arasında hesaplanır ve değerin 1'e yakın olması modelin başarılı bir performans gösterdiğini, -1'e yakın olması modelin yanlış yönde bir öngörüde bulunduğunu, 0'a yakın olması da modelin tesadüfi öngörüler geliştirdiğini temsil eder.

6.6 R ile Sınıflandırma Algoritmaları Kullanarak Bir Tahmin Modeli Geliştirme

Bu uygulamada aşağıdaki veri madenciliği aşamaları adım adım izlenerek `https://archive.ics.uci.edu/ml/datasets/Student+Performance` adresinden erişime açık olan Student Performance isimli veri seti kullanılacaktır. Veri setinin her bir niteliğine ait açıklamalar ilgili web sayfasında mevcuttur.

Veri madenciliği aşamaları:

1. Veri Seçimi: Veri seçimi aşamasında veri setini oluşturacak farklı ortamlardan elde edilmiş olan tüm veriler aynı ortama her bir veri satırının doğru bir biçimde eşleştirilerek taşınması ve veri bütünleştirme işleminin tamamlanmasıdır. Kullanılacak olan veri setinde bu işlem gerçekleştirilmiş olduğundan bu uygulamada herhangi bir işlem yapılmayacaktır.
2. Veri Önişleme: Bu aşamada veri seti incelenerek sınıflandırma için kategorik olmayan veriler kategorize edilerek hazırlanır ve kayıp veriler için var olan yöntemlerden biri uygulanır. Bu yöntemlerden bazıları kayıp veriye sahip olan satırı veri setinden çıkarmak, kayıp verinin olduğu hücreye aynı nitelikteki diğer değerlerin ortalamasını yazmak ya da bu değeri çevresindeki diğer değerler göz önünde bulundurarak sentetik veri üreterek hesaplamak. Ayrıca veri önişleme aşamasında gürültülü veri olarak adlandırılan toplanan verinin türüne göre tespit edilebilecek aykırı veriler veri setinden çıkarılabilir. Yine kullanılacak veri seti bu aşamadan da geçirildiği için uygulama kısmında veri önişleme yapılmayacaktır.
3. Veri İndirgeme: Eldeki veri setinin nitelikleri için ilişki analizi yapılarak tahmin edilecek hedef nitelik ile herhangi bir ilişki tespit edilmeyen nitelikler veri setinden çıkarılabilir. Eğitsel veri madenciliğinde veri setlerinin nitelik ve satır sayılarının genellikle az olması sebebiyle geliştirilen modellerin daha yüksek performans gösterip göstermedikleri bu aşama gerçekleştirilmeden ve gerçekleştirilerek karşılaştırılmalıdır. Son olarak hedef niteliğin mümkünse ikili bir biçimde kategorize edilmesi önerilir. Bu sayede model her bir satırdaki kayıt için pozitif ya da negatif olarak tahminde bulunabilmektedir. Elimizdeki veri setinde hedef nitelik(G3) 0-20 arasında bir değer olarak girilmiştir. Bu noktada modelin işini kolaylaştırmak ve sonucu daha kolay yorumlayabilmek için Excel üzerinde hedef nitelik 0-9 -> 0(Başarısız) ve 10-20->1(Başarılı) şeklinde kategorize edilmiştir.
4. Veri Madenciliği: Bu aşamada veri madenciliği modellerinden hedefe uygun olan seçilere uygun yöntemin teknikleri denenir ve en iyi performansı gösteren algoritmanın geliştirdiği model tercih edilir. Ayrıca veri setinde kategorik hale getirilen hedef nitelik dengesiz bir dağılım gösteriyor ise over-sampling teknikleri kullanılarak dağılım dengelenmeye çalışılır. Bu noktada eldeki veri seti için sınıflandırma algoritmalarından Rastgele Orman ve Naive Bayes algoritmaları uygulama için seçilmiştir ve veri seti incelendiğinde başarılı öğrenci sayısının 265, başarısız öğrenci sayısının ise 130 olduğu görülmüştür. Veri setine hedef nitelikteki dengesizliği gidermek için en güvenilir ve en çok tercih edilen over-sampling tekniklerinden SMOTE(Syntetic Minority Over-sampling Technique) uygulanmıştır.
5. Değerlendirme: Modelin değerlendirilmesi için daha önceki başlıkta belirtilen değerler yorumlanır ve benzer çalışmalar ile karşılaştırma yapılarak modelin istenileni elde edip etmediği kontrol edilir.

RStudio üzerinde yeni bir R script dosyası açarak ilk olarak projede kullanılacak bazı paketlerin

yüklenmesi için ve yüklü olan paketlerdeki kütüphanelerin kullanılabilmesi için aşağıdaki kod satırları çalıştırılmalıdır.

```
install.packages("readxl");
library(readxl);
library(dplyr);
library(caret);
library(UBL)
```

Kullanılacak olan kütüphaneler seçilen yöntem ve algoritmalara göre değişiklik gösterebilir. Ayrıca bazı kütüphaneler yüklenirken zorunlu olarak yüklenen başka kütüphaneler de mevcuttur. Örneğin sentetik veri üretme amacıyla UBL kütüphanesi dahil edilirken bununla birlikte Rastgele Orman algoritması için kullanılan randomForest kütüphanesi de script içerisine dahil edilir. Kullanılan kütüphaneler ve ne için kullanıldıkları Tablo 6.3'te verilmiştir.

Tablo 6.3: Kullanılan kütüphaneler.

readxl	Bir excel dosyasından veri setinin çekilmesi için
dplyr	Veri üzerinde gruplama yapma ve verinin özetini çıkarma gibi temel işlevler için
caret	Veriyi test ve eğitim seti olarak istenilen oranda ayırabilmek için
UBL	Sentetik veriler oluşturan SmoteClassif() fonksiyonu için

Aşağıdaki kod satırı ile excel formatındaki veri seti dat içerisine aktarılır.

```
dat <- read_excel("***/student.xlsx");
```

Daha sonra veri indirgeme aşamasında 0-1 olarak kodlanan performance isimli nitelik hedef nitelik olarak kullanılmak için factor() fonksiyonu ile veri seti içerisinde aynı sütuna tanımlanır.

```
dat$performance=factor(dat$performance);
```

dplyr kütüphanesi sayesinde performance niteliğine göre gruplandırılan verilerin kaçar adet olduğunu özetlemek için aşağıdaki kod satırı çalıştırılır.

```
dat %>% group_by(performance) %>% summarise(number = n());
```

Kodun ekran çıktısı Şekil 6.7'deki gibidir.

	performance <fct>	number <int>
1	0	130
2	1	265

Şekil 6.7: Başarı durumuna göre gruplanmış veri seti.

Buna göre başarısız(0) öğrenci sayısı 130 iken başarılı(1) öğrenci sayısı 265'tir. Başarılı ve başarısız öğrencilerin arasındaki bu farkı azaltmak için SmoteClassif() fonksiyonu uygulamadan önce vektör tipinde olan veri setimizi dataframe tipine çeviriyoruz.

```
dat<- as.data.frame(dat);
```

Artık veri seti sentetik veriler oluşturularak dengelenebilir.

```
dat = SmoteClassif(performance~., dat, C.perc =
list("0" = 2), k = 5);
```

SmoteClassif() fonksiyonu belirlenen k değeri(varsayılan olarak 5 tercih edilmektedir) sayısı kadar komşudan sentetik veri oluşturacaktır. list() içerisindeki "0"=2 başarısız öğrencilerin sayısını 2 kat artırma anlamına gelmektedir. İşlem sonrası veri setindeki dağılım tekrar incelendiğinde Şekil 6.8'deki sayılara ulaşılmıştır.

performance	number
<fct>	<int>
0	260
1	265

Şekil 6.8: Veri dengelendikten sonra başarı durumuna göre gruplanmış veri seti.

Burada dikkat edilmesi gereken hususlardan biri SmoteClassif() fonksiyonu k en yakın komşulardan sentetik veriler üretirken sadece nümerik veriler için veri üretebilmektedir. Bu sebeple veri setindeki tüm niteliklerin ön işleme aşamasında nümerik olarak kodlanması gerekmektedir. Kullanılan veri setinde nümerik olmayan nitelikler dönüştürülerek analiz tamamlandığında son olarak model üzerindeki nitelik öncelikleri test edilmiş ve nümerik olmayan niteliklerin neredeyse hiçbir önemi olmadığı görülmüştür. Önemsiz olan bu niteliklerin tümü excel dosyasından silinerek çıkarılıp bu şekilde analizler gerçekleştirilmektedir.

```
dat = SmoteClassif(performance~., dat, C.perc =
list("0" = 2), k = 5);
```

set.seed(123);

Bu fonksiyon ile veri test ve eğitim seti olarak ikiye ayrılmadan önce rastgelelik sağlanması amacıyla kullanılmaktadır. Ardından veri caret kütüphanesi yardımıyla %70 eğitim veri seti, %30 test veri seti olarak ayrılıp Train ve Test içerisine atılmıştır.

```
trainIndex <- createDataPartition(dat$performance,
p = .7, list = FALSE, times = 1);
Train <- dat[trainIndex,];
Test  <- dat[-trainIndex,];
```

Eğitim veri seti ve test veri setindeki başarılı(1) ve başarısız(0) öğrencilerin dağılım oranı dengeli bir biçimdedir.

```
model=randomForest(Train$performance~.,data=Train,
ntrees=500, importance=TRUE);
```

Yukarıdaki kod satırı randomForest kütüphanesinin aynı isimli fonksiyonu ile Rastgele Orman algoritması kullanılarak modeli eğitmektedir.

```
pred <- predict(model, Test[,-33]);
confusionMatrix(table(Test$performance, pred));
```

Eğitilen model test veri setinin içerisinden 33. sütun olan hedef nitelik çıkarılarak bu niteliğin tahmin edilmesi amacıyla çalıştırılır ve Şekil 6.9'teki hata matrisi elde edilir.

Hata matrisine göre model test veri setindeki 78 başarısız(0) öğrenciden 76'sını ve 79 başarılı(1) öğrenciden 73'ünü doğru bir şekilde tahmin etmiştir. Modele ait değerlendirme yapabilmek için kullanılabilen diğer değerler de bu matrisin alt kısmında oluşturulur ve Şekil 6.10'teki gibi görüntülenir.

```
Confusion Matrix and Statistics

   pred
     0  1
0 76  2
1  6 73
```

Şekil 6.9: Analiz sonucu olarak hata matrisi görünümü.

Görüntüde modelin genel başarı oranının %94.9, başarılı öğrencileri tahmin etme oranının %92.4 ve başarısız öğrencileri tahmin etme oranının ise %97,4 olduğu görülmektedir.

```
               Accuracy : 0.949
                 95% CI : (0.9021, 0.9777)
    No Information Rate : 0.5223
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.8981

 Mcnemar's Test P-Value : 0.2888

            Sensitivity : 0.9268
            Specificity : 0.9733
         Pos Pred Value : 0.9744
         Neg Pred Value : 0.9241
             Prevalence : 0.5223
         Detection Rate : 0.4841
   Detection Prevalence : 0.4968
      Balanced Accuracy : 0.9501

       'Positive' Class : 0
```

Şekil 6.10: Model değerlendirme değerleri görünümü..

Kappa değeri incelendiğinde 1'e yakın olan değerin model performansını mükemmele yakın olarak tanımladığı söylenebilir. Ayrıca MCC değeri elde edilen veriler ile hesaplanabilir. Tüm bunların yanında model içerisinde önemli dallanmaların hangi niteliklerde gerçekleştiğini görebilmek için aşağıdaki kod satırı kullanılabilir.

```
importance(model);
```

Çıktısı (Şekil 6.11):

	0	1	MeanDecreaseAccuracy	MeanDecreaseGini
age	5.517109	2.16086639	5.590990	6.049311
Medu	4.139007	0.93195787	3.878476	5.852933
Fedu	5.299699	1.26425587	5.017102	6.460723
traveltime	3.974523	1.20205741	3.546245	3.450937
studytime	8.484220	-2.42242688	5.125810	4.267104
failures	8.546592	14.29627088	16.515263	12.085938
famrel	2.194620	2.46730469	3.266984	3.995831
freetime	5.432404	-0.03798337	4.201903	5.826983
goout	5.505783	7.00619233	8.732651	7.713693
Dalc	6.039696	2.06371586	5.838923	3.247001
Walc	5.118096	4.58229085	7.190983	4.915188
health	4.225254	0.67095580	3.475049	5.737645
absences	7.433377	4.17131174	8.158998	7.456892
G1	24.323836	24.98116314	29.420330	42.916402
G2	36.532313	33.54038393	39.235777	60.829833

Şekil 6.11: Modelin niteliklere verdiği önem değerleri.

Değerler incelendiğinde ilk(G1) ve ikinci(G2) periyod notlarının büyük oranda önemli dallanmalar yaptığı yüksek değerlerden görülürken ayrıca daha önceki sınıf başarısızlıklarının sayısının da dallanmalarda önemli olurken özellikle başarılı öğrencilerin tahmin edilmesinde rol oynadığı görülmektedir.

Analizler sonrası modelin önemli gördüğü nitelikler ile tekrar yeni bir model oluşturmak az nitelik sayısı sebebiyle daha kötü performans gösteren bir model ortaya çıkmasına sebep olabilir.

Fakat özellikle ilk(G1) ve ikinci(G2) periyod notlarının etkisinin büyüklüğü modelin yüksek tahmin performansı göstermesine neden olurken, öğrencilerin notlarının bulunduğu bir veri seti ile gerekli süre içerisinde bir tahmin üretilemeyip öğrenme süreci ya da ortamına müdahale etmenin verimsiz olacağı düşünüldüğünde bu niteliklerin veri setinden çıkarılarak yeni bir model geliştirmek eğitsel veri madenciliği açısından daha doğru olacaktır. Öğrenci notları dışındaki diğer nitelikler dönem boyunca dersin son haftalarından önce toplanabilir ve öğrenme süreci sonlanmadan önce gerekli görülen düzenlemeler modelin çıkardığı başarısız öğrenciler üzerinde ve önemli nitelikler açısından düzenlemeler yapılarak öğrenci başarı performansı artırılabilir.

6.7 Kaynaklar

Bousbia, N., & Belamri, I. (2014). Which Contribution Does EDM Provide to Computer-Based Learning Environments? A. Pena-Ayala (Dü.) içinde, Educational Data Mining (s. 3-28). Springer. doi:10.1007/978-3-319-02738-8_1

Güldal, H., & Çakıcı, Y. (2017). Eğitsel Veri Madenciliği. Balkan Education Studies 2017 (s. 135-143). Edirne: Trakya Üniversitesi Yayınları.

Hanna, M. (2004). Data Mining in the E-learning Domain. Campus-Wide Information Systems, 21(1), 29-34.

Hicham, A., Jeghal, A., Sabri, A., & Tairi, H. (2020). A Survey on Educational Data Mining [2014-2019]. Proceedings of The International Conference on Intelligent Systems and Computer Vision (s. 1-6). Fez,Morocco: IEEE.

Ktona, A., Xhaja, D., & Ninka, I. (2014). Extracting Relationships between Students' Academic Performance and Their Area of Interest Using Data Mining Techniques. Proceedings of 2014 Sixth International Conference on Computational Intelligence (s. 6-11). Bhopal-India: Communication Systems and Networks.

Landis, R., & Koch, G. (1977). The Measurement of Observer Agreement for Categorical Data. Biometrics, 33(1), 159-174.

Merceron, A., & Yacef, K. (2005). Educational Data Mining: a Case Study. (s. 1-8). Amsterdam: International Conference on Artificial Intelligence in Education.

Razak, R. A., Omar, M., & Ahmad, M. (2018). A Student Performance Prediction Model Using Data Mining Technique. International Journal of Engineering & Technology, 7(2.15), 61-63. doi:10.14419/ijet.v7i2.15.11214

Romero, C., & Ventura, S. (2007). Educational Data Mining: A Survey from 1995 to 2005. Expert Systems with Applications, 33(1), 135-146. doi:10.1016/j.eswa.2006.04.005

Romero, C., & Ventura, S. (2010). Educational Data Mining: A Review of the State of the Art. IEEE Transactions on Systems, Man, and Cybernetics, 40(6), 601-618.

Tekin, A., & Öztekin, Z. (2018). Eğitsel Veri Madenciliği ile İlgili 2006-2016 Yılları Arasında Yapılan Çalışmaların İncelenmesi. Eğitim Teknolojisi Kuram ve Uygulama, 8(2), 67-89. doi:10.17943/etku.351473

Yu, R., Li, Q., Fischer, C., Doroudi, S., & Xu, D. (2020). Towards Accurate and Fair Prediction of College Success: Evaluating Different Sources of Student Data. Proceedings of The 13th International Conference on Educational Data Mining (s. 292-301). Sanal Konferans: International Educational Data Mining Society.

7. Ürün Yaşam Döngüsü Yönetimi (PLM)

"Dijitalleşen Dünyada Şirketlerin Sürdürülebilirliğini Sağlamak" (PLM)

Ramise KOÇAK*

*Otokar Otomotiv ve Savunma Sanayi A.Ş.

7.1 Giriş

Sürdürülebilirlik kavramının önemini vurgulayan, hayatta kalabilmek için değişen koşullara uyum sağlamanın nedenli önemli olduğunu anlatan, Darwin bir sözü ile makaleme başlamak istiyorum;

"Ne en zeki olan hayatta kalır, ne en güçlü olan; hayatta kalan, değişime adapte olabilen ve içinde bulunduğu çevredeki değişime en iyi uyum sağlayabilendir. " Charles Darwin

Dijitalleşme, değişen dünya koşulları ile birlikte hayatımızda vazgeçilmez hale geldi ve dijitalleşme süreçleri her alanda kullanılmaya başlandı. Eğitimden, sağlığa, üretimden, satışa aklınıza gelebilecek pek çok alanda yerini aldı.

Teknoloji her gün hızla değişime uğruyor ve bu hıza uyum sağlayan dijitalleşen, değişimi iş süreçlerine uygulayabilen kurumlar/kuruluşlar ayakta kala biliyor.

Dijitalleşme kavramının kısaca tanımını yapalım.

Dijitalleşme nedir?

Elimizdeki verinin (Bilgi, belge, analiz sonuçları yada araştırmaların) bilgisayar destekli olarak dijital ortama aktarılmasıdır.

Dijitalleşme, verilerden yararlanarak sonuçlar elde edilmesini sağlamaktadır.

Ekonomi, finans, eğitim, Sağlık, savunma, güvenlik ve iş dünyasında etkin olarak kullanılmaktadır.

Dijital dönüşüm nedir?

dijital dönüşüm, kavram olarak dijitalleşmeden daha geniş bir kapsama sahiptir.

Bir ürün ya da hizmet kullanıcı beklentilerini karşılayacak şekilde, önceki versiyonundan daha verimli bir kullanıma sahip olarak dönüştürüldüğünde, dijital dönüşüm ortaya çıkar.

Dijital teknolojiler sayesinde, kurumlar / kuruluşlar yeni metotları benimseyerek verimlilik, kişiselleştirme ve güvenlik anlamında müşterilerine daha kaliteli hizmetler sunmaya başlamıştır. Dijitalleşme hızla artarken rekabetçi sektörlerin, iyi piyasa araştırması yapmaları, trendleri takip etmeleri ve müşterinin eğilimlerine yönelik yatırımlar yapmaları, oldukça önem arz eder. Yıkıcı yenilikler karşısında teknolojinin hızına ayak uyduramayan şirketler sürdürülebilirliğini sağlayamıyor.

Bir örnek ile pekiştirelim, akıllı telefonların profesyonel bir kamera gibi kullanılması ve dijital fotoğrafların önem kazanmasıyla, fotoğraf makinesi konusunda dünya devlerinden biri olan Kodak, değişimin hızına ayak uyduramayarak, ciddi bir yıkım yaşadı ve 2012 de iflasını açıkladı. Benzer şekilde akıllı telefon teknolojisine uyum sağlayamayan bir dönem Pazar lideri olan Nokia firması , iphone ve Samsung telefonları piyasaya çıkınca rekabet edemedi, 2020 verilerine göre Şirket akıllı telefon segmentinde % 0.7'lik pazar payı ile 15'inci sırada yer aldı.

Bu örnekleri daha da çoğalta biliriz, start up olarak başlayan piyasa verilerini doğru analiz edip eğilimler doğrultusunda yeni yatırım yapan şirketlerin, hızla büyüyüp Pazar lideri oldukları (facebook, uber, Tesla vs..) diğer taraftan büyük, hantal, değişimi iş süreçlerine hızla adapte edemeyen dünyanın en büyük şirketlerinin bile piyasadaki gücünü kaybetmeleri, yıkıcı inovasyon karşısında ne kadar güçsüz olduklarını gösteriyor. Compacq, 1982 yılında kuruldu, Intel ile olan ortaklığından dolayı her Intel işlemcinin gelecek neslini içeren bilgisayarlarla çıkan ilk şirket olduğu için pazarda teknolojik bir lider konumundaydı 1990'larda 2000'lerin ortalarına kadar dünyanın bir çok yerinde önde gelen bilgisayar markalarından birisi olmayı başaran Compaq, 2013 yılında tamamen ortadan kaldırıldı.

Sony, telefon üretiminde kendi prensiplerinden vaz geçmedi ve değişime ayak uyduramadı, bunun sonucunda bir çok ülkeden çekilmek zorunda kaldı.

Ekonomiler ve toplumlar dijital dönüşümde hızlı aksiyon alma hedefinde iken, pandemi bu süreci daha da hızlandırdı. Covid 19 süreci ile birlikte, teknolojinin hayatımızdaki yeri daha da güçlendi ve hatta hayatımızın ayrılmaz bir parçası haline geldi.

Bu nedenle dijital dönüşüm, şirketler için bir tercih meselesi olmaktan çıktı, hayatta kalmanın gerekliliği haline dönüştü, teknoloji ile fırsatları yakalayamayan kurumların önümüzdeki yıllarda yok olma ihtimalinin oldukça yüksek olduğunu birkez daha hatırlatmak isterim.

Dijitalleşme şirketlerde hangi alanlarda kullanılabilir?

Büyümek ve gelişmek isteyen işletmeler için dijital verilerin analizi oldukça önemlidir.

Özelikle arge – üretim – pazarlama – satış - insan kaynakları için kullanımı oldukça işlevseldir. Pazarlama ve insan kaynakları alanında veriye dayalı stratejiler, raporlar, planlar ve ölçümler hazırlanarak müşteri ihtiyacına yönelik yatırımlar yapılır ve insan kaynağı yönetimi ona göre planlanır.

Şirketlerde dijitalleşmeyi sağlayan araçlar ve tekniklere örnek vermek gerekirse; en sık kullanılan; yapay zeka, yapay zeka yazılımları, nesnelerin interneti, sanal gerçeklik, arttırılmış gerçeklik, büyük veri, algoritma, endüstri 4.0 gibi teknikleri sayabiliriz.

Büyümek , rakipleri ile yarışabilmek ve hatta bir adım önde olabilmek isteyen şirketlerin teknolojik gelişmeleri yakından takip eden, veri analizi yapabilen yazılımlarla çalışıp inovasyonu sağlayacak dönüşümleri iş süreçlerinde uygulamaları gerekmektedir.

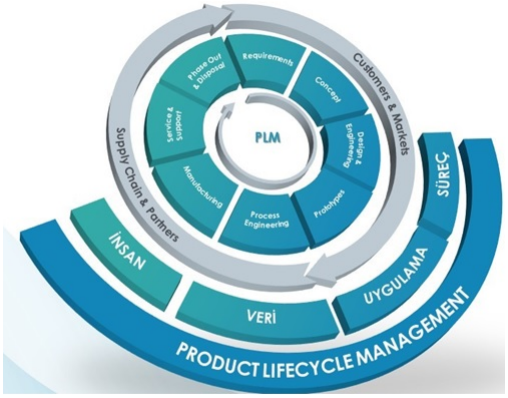
Yeni dünya normları, kurumları iş süreçlerini dijital ortamda ve uçtan uca bütünleşik bir platform üzerinde yönetmeye yönlendiriyor, çünkü; işbirimleri arasında veri transferinin hızlı olması çalışanın veriye kolay ulaşabilmesi, verimliliği arttırıyor. Teknolojinin gelişimi ve her alanda yer alması ile birlikte, ürün geliştiricileri, tarihte bu eşi benzeri görülmemiş değişime ayak uydurmak için adeta birbiri ile yarışıyor . Günümüzde tüm ürün ve hizmetler mobil ya da kablolu Internet bağlantılarıyla iletişim kuran sistemlere dahil ediliyor. Ürün ekipleri için en önemli KPI, ürün

geliştirme sürecinde hızlanma, ürünü piyasaya hızlı sunma ve müşteri eğilimleri doğrultusunda üretim planlarını güncelleyip yatırımlarını ona göre yapmaları gerekiyor.

İşte bu noktada devreye PLM yani ürün yaşam döngüsü yönetimi giriyor.

7.2 PLM (Product Life Cycle Management)

PLM (Product Lifecycle Management) kelimesinin baş harflerinden oluşur (Şekil 7.1). Bir ürünün fikir aşamasından başlayıp kullanıcının eline ulaşana kadar geçirdiği süreçlerin tamamını kapsar. Bu süreçler PLM yazılımları yardımıyla dijital bir ortamda yönetilir. PLM ile bir ürüne ait veriler merkezileştirilip, bilginin yeniden kullanımı optimize edilebilir. En önemlisi pahalı ve hatalara yol açan gereksiz bilgi yığınını ortadan kaldırılır.



Şekil 7.1: PLM

PLM Sistemi; Ürün odaklı mühendislik faaliyetlerinde, tüm süreç ve bilgi akışının, etkileşimli ve işbirlikçi bir ortam içerisinde, kurum ve bireyler arasında yönetimi ve paylaşımını sağlayan, işletmenin en kıymetli değeri olan Mühendislik Datalarının güvenli bir şekilde, sürece katılan tüm birim ve bireyler arasında, etkin ve etkileşimli bir şekilde yönetilip, denetlenmesi (PDM) yanı sıra, "dijital" ürün sürecinde; Doküman, Proje, İş Akışı, Değişiklik, Konfigürasyon, Varyant, Kalite, Maliyet ve Tedarikçi Yönetimi gibi bir çok aşamayı içerisine alan, ürünün fikirsel doğumundan, üretimine ve akabinde servis ve teknik destek süreçleri dahil olmak üzere, ürünün piyasadan çekilmesine kadar tüm süreçlerde verimlilik sağlayan bir teknoloji uygulamasıdır

PLM çözümleri, avantajları ;

- Tüm süreçlerin ve birimlerin tek bir platformdan izlenebilirliği ve takibinin sağlanması
- Bilginin tekrar kullanılabilirliğinin sağlanması, ürünün pazara daha kısa sürede, daha hızlı ve daha az maliyetle çıkması
- Mevcut sürecinize uygun iş akışı ve onay mekanizması ile hatalı tasarım, üretim sürecinin önüne geçilmesi
- Ürün, revizyon tarihçelerinin takibinin ve bilgilerin yeniden kullanılabilirliğinin sağlanması
- Bölümler arası doğru bilgi akışının sağlanması
- CAD Dataları ve dokümanların tek bir platformdan yönetilmesi
- İş birlikteliğinin ve verimliliğin artırılması
- Kurumsal hafızanın güçlendirilmesi ve bilgi güvenliğine katkısı

Gibi avantajlar sunarak bir kuruluşun müşterilerin ihtiyaçlarını daha etkin bir sekilde karşılamasına ve rakipleri karşısında güçlü olmasına yardımcı olur.

Akıllı fabrikalar da uygulanan Dijital uygulama çözümleri
"Devletler milli teknolojiler üretip satabildikleri ölçüde rekabet gücüne sahip oluyor."
Bunun temelinde uzman mühendislik ve arge ekipleri ile kullanılan gelişmiş teknolojiler yatıyor. Gelişmiş teknolojiye sahip ülkeler her alanda daha fazla söz sahibi oluyor.
Pek çok alanda, eğitim den, sağlığa, üretim den, satışa dijitalleşme örnekleri sayılabilir.
Kapsamı daraltarak, üretim sektöründe akıllı fabrika bazında örnekleme yapalım.
Akıllı fabrikalar da uygulanan yöntemler (Şekil 7.2):

- Bilgisayar destekli tasarım
- Bulut Bilişim
- Nesnelerin İnterneti
- Üç boyutlu baskı ile üretim
- Robot teknolojilerinin kullanılması
- Veri analizi
- Yapay zeka ve Makine öğrenmesi
- Arttırılmış gerçeklik/Sanal Gerçeklik ile tasarım sürecini kısaltma
- PLM/PDM ile bütünleşik veri yönetimi
- ERP çözümleri
- RPA Robotik süreç otomasyonu
- İnsansız Fabrika otomasyonu

Şekil 7.2: Akıllı fabrikalar da uygulanan yöntemler

7.3 Kaynaklar

https://nokiamob.net/2021/01/29/nokia-mobile-shipped-15-5-million-phones-in-q4-2020/?utm_campaign=DonanimHaber&utm_medium=referral&utm_source=DonanimHaber

https://www.theverge.com/2012/5/23/3039627/hp-compaq-low-end-branding
https://www.teknolojioku.com/mobil/sony-telefonlarinin-satilmamasi-icin-ugrasiyor-60f2abbf9f7584588e544a3f

8. Robotik Süreç Otomasyonu (RPA)

Robotik Süreç Otomasyonu (RPA)

Hülya ÇİVAK*, Safiye TURGAY†
***Karabük ve Sakarya Üniversitesi Ortak Programı, Endüstri Mühendisliği**
†Sakarya Üniversitesi, Endüstri Mühendisliği

8.1 Giriş

RPA hayatımıza gireli kısa bir süre olmasına rağmen bu kadar popüler olmasının en önemli sebepleri, tekrarlı işlerin otomotize edilerek, daha hızlı, daha verimli, daha az hata ile uzun ve sıkıcı süreçleri başarı ile tamamlamasıdır. Ayrıca günün her saati yorulmadan, mola vermeden ve tanımlanmış kuralların dışında hareket etmeyen bu sanal çalışanlar, verimliliği de arttırmaktadır.

Pandemi ile birlikte teknolojiye olan adaptasyonun artması neticesinde, artık teknolojiye olan olumsuz bakış açısı yerini teknolojiyi etkin kullanmaya bırakmıştır. Yine pandemi ile hayatımıza giren evden çalışma şekli ile de RPA süreçleri, çalışma alanlarındaki sınırları ortadan kaldırmıştır.

Dünya Ekonomi Forum'un raporuna göre 2025 yılı için öngörülen çalışma şekilleri ve iş gereksinimler şu şekilde olacaktır:

- İş gücünün %50'si evden çalışacak
- 26 ülkede 85 milyon iş ortadan kalkacak
- 97 milyon yeni iş ortaya çıkacak
- İş gücünün %50'sine yeni beceriler kazandırılması gerekecek
- İş dünyasında çalışan insan ve robot sayısı eşit olacak - (Dünya Ekonomik Forumu)

8.1.1 RPA Nedir?

Robotik süreç otomasyonu (RPA), dijital sistemler ve yazılımlarla etkileşime giren insan eylemlerini taklit eden yazılım robotlarının oluşturulmasını, dağıtılmasını ve yönetilmesini kolaylaştıran bir yazılım teknolojisidir. Tıpkı insanlar gibi, yazılım robotları da bir ekranda ne olduğunu anlamak, doğru tuş vuruşlarını tamamlamak, sistemlerde gezinmek, verileri belirlemek ve ayıklamak ve

çok çeşitli tanımlanmış eylemleri gerçekleştirmek gibi şeyler yapabilir. Ancak yazılım robotları, kalkmaya, gerilmeye veya bir kahve molası vermeye gerek kalmadan bunu insanlardan daha hızlı ve daha tutarlı bir şekilde yapabilir. (www.uipath.com, 2021)

8.1.2 RPA'in Sağladığı Faydalar

RPA, bir insanın ekranda yaptığı tüm işlemleri taklit edebilmektedir. Bu özellikleri ile rutin ve kural tabanlı süreçlerde RPA kolaylıkla uygulanabilmektedir. Böylece herhangi bir değer katmayan, sıkıcı, verimliliğe katkı sağlamayan işlerin robotlara aktarılması çalışan için de motive edici unsurlardan olmaktadır. Çünkü bu tür işleri yaparken, çalışanlarda zamanla isteksizlik, dikkat kaybından kaynaklı hata oranlarının artması, yeteneklerin körelmesi gibi birçok olumsuz etkisi olabilmektedir. Belirtilen işleri, sanal asistanlara bırakıp, sektörün ihtiyaçlarını daha iyi belirlemek, piyasa analizi, farklı satış yöntemleri, yeni ürün ve tasarım fikirleri gibi insanın yetkinliklerini ortaya çıkaracak işlere yönelmek çok daha fazla fayda sağlayacaktır.

Robotların arz ve taleplerinin yüksek olduğu bu noktaya gelmesini sağlayan konulardan bazıları şunlardır;

- Verimlilik
- Maliyet
- Implementasyon
- Müşteri Memnuniyeti
- Kalite Artışı
- Denetim

Robot tarafından yapılan işlere bakıldığında, içerisinde katma değerli olmayan ve tekrarlı işlerin olduğu görülmektedir. Bu sebeple artık sanal asistan olarak adlandırılan yazılımsal robotlar devreye girmiştir.

8.1.3 RPA Teknolojisi İle Yapılan Bazı İşlemler

Dosya-Klasör İşlemleri RPA ile istenilen bir/birden fazla klasörden dosyalar alınıp, işlenebilir, dosya isimleri değiştirilebilir, dosyalar silinebilir, taşınabilir.

Pdf Okuma pdf formatındaki dosyaların içerisinden istenilen herhangi bir bilgi okunabilir, işlenebilir. Örneğin bir faturadan istenilen alanları (tutar, tarih, firma vs) okuyup, istenilen sisteme bu faturayı işleyebilir, veriler tanımlanmış kurallara uymadığı takdirde iş birimine bilgilendirme yapabilir.

Excel Aktiviteleri .xlsx / .xls formatındaki dosyalar okunabilir, içerisindeki veriler işlenebilir. Örnek olarak excelde yer alan düşeyara özelliği, tabloların okunup, birleştirme, istenilen satırları tekil yapma, matematiksel işlemler, filtreleme gibi işlemleri de yapabilmektedir.

Mail Dinleme RPA teknolojisi ile belirtilen mail adreslerine gelen mailler filtrelenebilir, body kısmındaki veriler okunabilir, mail ekleri alınabilir ve işlenebilir. Tanımlanan kullanıcı bilgisi ile robot istenilen ek/leri, subject, body kısımlarını da doldurarak mail gönderebilir, gelen maili karşılayabilir. Tanımlı kurallara göre aksiyon alabilir.

Web Tarama-Data Çıkarım Robot tıpkı bir insanın yaptığı gibi kullanıcı ve şifre ile bir web sitesine/programa login olabilir. Tanımlı kurallar çerçevesinde ilgili web sitesinde verileri okuyabilir, işleyebilir. İşlemleri bitirdikten sonra logout olarak işlemini sonlandırabilir. Anahtar kelimeler kullanarak arama motorlarında araştırma yapıp çıkan sonuçları bir rapor haline getirebilir. Yine belli anahtar kelimelerle sosyal medyada hashtag taraması yapıp rapor hazırlayabilir.

Robotic Process Automation (RPA) yukarıda bahsedilen konular hakkında hizmet veren yeni

nesil teknolojilerden biridir. Yani kural tabanlı, tekrarlı rutin işleri otomotize edilmesidir. RPA'in bu kadar revaçta olmasının sebepleri sadece bu işleri yapması değil, hızlı ve daha minimum hata ile kendisine tanımlanan işleri yapmasıdır.

Gartner Raporuna göre şirketlerin %32'si, kuruluşlarında robotik uygulama sürecindedir. Böylece manuel çalışmaya göre %25 ila %50 maliyet tasarrufu sağlandığı görülmüştür. (`www.gartner.com/en/finance/trends/robotic-process-automation/building-a-robotics-roadmap`) Takip eden bölümlerde, yeni nesil teknolojiler, RPA ürünleri, RPA'in dünü, bugünü, yarınından bahsedilecektir.

8.1.4 RPA Ürünleri

Global RPA ürünlerinden bazıları şu şekildedir (Şekil 8.1):

UiPath: 2005 yılında Daniel Dines liderliğinde Romanya'da küçük bir ekiple kurulmuştur. UiPath gerek ücretsiz akademi platformu gerekse ücretsiz community studio ve robot desteği sağlamasıyla birlikte sertifikalı 35.000'den fazla geliştiriciye RPA öğretmiştir. Şu an dünyada en çok kullanılan RPA üründür ve şirket 30 milyar doların üstünde marka değerine sahiptir.

Automation Anywhere: 2003 yılında California'da kurulan ve daha sonradan Automation Anywhere olarak isim değiştiren şirket, oldukça revaçta olan RPA ürünlerinden biri olmuştur. Automation Anywhere Üniversitesi ile RPA geliştiricilerine teknik eğitim sağlamaktadır. (`www.automationanywhere.com`, 2021)

Blue Prism: Davis Moss tarafında 2001 yılında Warrington'da kurulmuştur. RPA teknolojisinin öncülerinden olan şirket Blue Prism Üniversitesi ile RPA geliştiricilerine eğitim desteği sağlamaktadır. Ürün yapısal olarak flowchart mantığı ile süreç tasarımı sağlamaktadır. (`www.blueprism.com`, 2021)

MS Power Automate: Microsoft şirketi tarafından sunulan RPA ürünü henüz yeni olmasına rağmen Gartner raporlarında yer almaktadır. (`powerautomate.microsoft.com`, 2021)

Şekil 8.1: RPA Raporu Kaynak: 2021 Gartner Magic Quadrant for Robotic Process Automation (RPA)

8.2 Literatür Taraması

(ÇALIŞKAN & KIRAN2, 2020) çalışmalarında, RPA'in şirketlere sağladığı faydalar anlatılmıştır. Bir otomotiv firmasında gerçekleştirilen RPA projesi sonrası proje katılımcıları ile anket ve mülakat yapılarak sonuçları değerlendirilmiştir. Çalışma sonucunda uygun iş süreçleri seçildiğinde RPA'nın faydalı bir teknoloji olduğu kanısına varılmıştır.

(YETİZ, TURAN, & CANPOLAT, 2021) çalışmalarında, banka personellerinin daha verimli çalışarak hedeflerini gerçekleştirmişlerdir. Böylece müşteri memnuniyetinin arttığı görmüşlerdir. Çalışmada, RPA'nın maliyet tasarrufu sağladığı, insan kaynağına duyulan ihtiyacı azalttığı böylece banka personelin diğer bankacılık işlemlerini yaparken daha verimli çalışabileceği ve personelin daha verimli alanlarda kullanılmasına olanak sağladığı sonucuna ulaşılmıştır.

(KAYA, TURKYILMAZ, & BIROL, 2019) çalışmalarında, RPA teknolojisinin, ERP ve MRP değişen maliyet muhasebesi ve finansal raporlama kavramları kapsamında uygulama ve gelişme alanlarını analiz etmişlerdir. Ayrıca RPA teknolojisinin geleneksel muhasebe ve maliyet muhasebesi süreçleri üzerindeki etkisi incelenmiştir. (KESTANE, 2021) çalışmasında, zeki otomasyon teknolojisi kullanımı ve iç denetim mesleğinin geleceği belirlenmesi amaçlamıştır.

8.3 Örnek Uygulama

Automation Anywhere'in Ağustos 2021 Challenge olarak duyurduğu uygulamada süreç yapısı şu şekildedir (Şekil 8.2):

1. Challenge sitesinde yer alan excel dosyası indirilir.
2. Excel dosyası okunur ve veriler alınır.
3. Verilen kullanıcı adı ve şifre ile Lab sitesine giriş yapılır.
4. Challenge ekranında ekrandan "PO Number" alınır:
   (a) Lab ekranında aratılır.
   (b) Çıkan sonuçlarda; "Ship Date" ve "Order Total" bilgileri alınır. "Order Total" verisinden "$" sembolü atılır.
   (c) Aynı ekrandan alınan "State" bilgisi excelde aratılır ve bulunan değer Challenge ekranındaki "Agent" alanına yazılır.
5. Bu işlemler tüm "PO Number" değerleri için yapılır.
6. Tüm satılar doldurulduktan sonra "Submit" butonuna basılır.

UiPath ile yapılan süreç adımları ekranda şu şekildedir:

Excel okunur ve bir datatable değişkenine atanır (Şekil 8.3).

Lab sitesine giriş için veriler tanımlanır. Kullanıcı adı ve şifre gibi bilgiler gerçek süreçlerde credential olararak Windowstan veya orchestratorden alınmalıdır (Şekil 8.4).

Challenge sitesi açılır (Şekil 8.5).

Get text aktivitesi ile PO Number alınır ve bir değişkene atanır (Şekil 8.6).

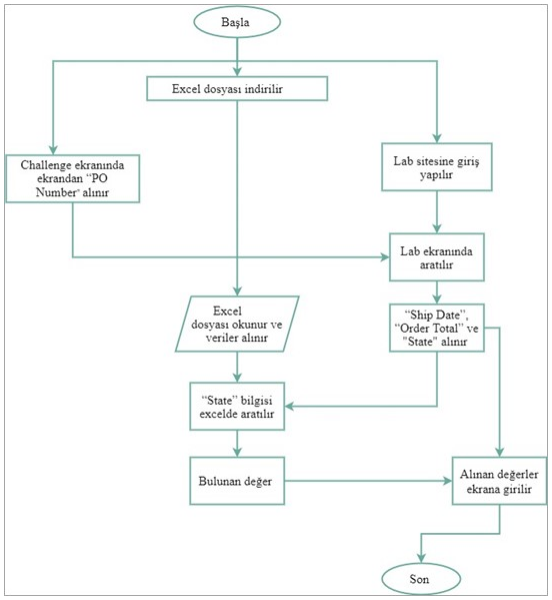
Alınan PO Number, lab ekranında aratılır ve çıkan değerler yine Get Text aktivitesi ile alınır (Şekil 8.7).

State değeri ise süreç yapısında belirtildiği gibi excelde aratılır (Şekil 8.8).

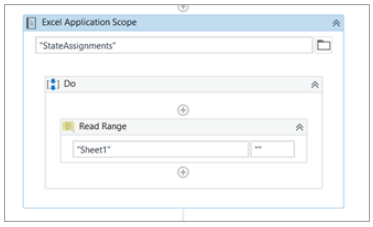
Alınan tüm veriler Challenge sitesinde ilgili alanlara girilir (Şekil 8.9).

Bu işlemler ekrandaki tüm PO Number değerleri için yapılması gerektiğinden, döngü içerisinde işlemler yapılır.

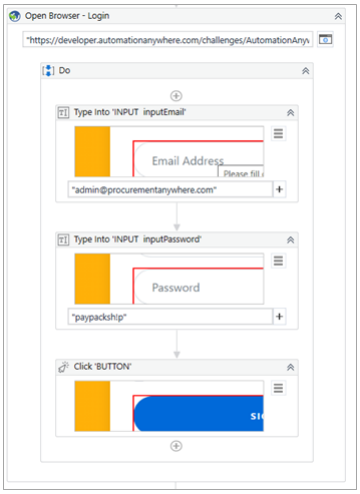
Tüm satırlar tamamlandığında Submit butonu tıklanır (Şekil 8.10).



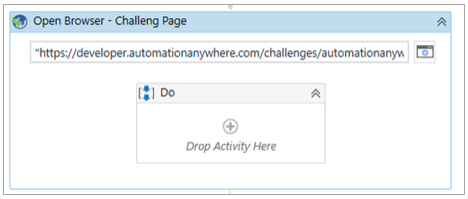
Şekil 8.2



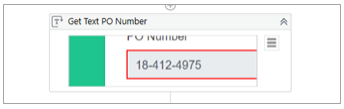
Şekil 8.3



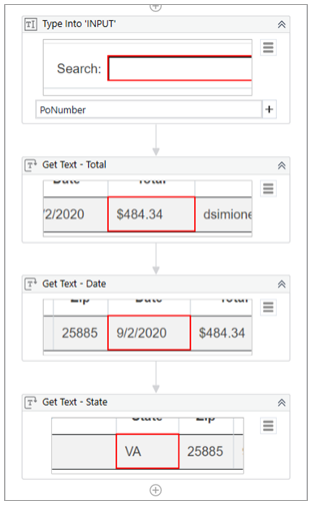
Şekil 8.4



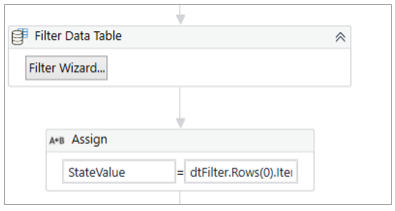
Şekil 8.5



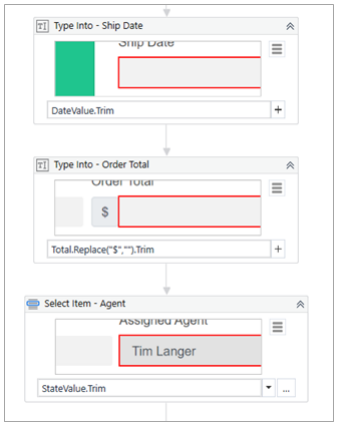
Şekil 8.6



Şekil 8.7



Şekil 8.8



Şekil 8.9

Şekil 8.10

8.4 Sonuç

Her geçen gün gelişen teknoloji ile birlikte, artan insan ihtiyaçlarını karşılama konusunda yeni ve farklı yöntemler ortaya çıkmıştır. Pandemi ile birlikte homeoffice çalışma şekliyle, birçok işin, mekan sınırlaması olmaksızın her noktada yönetebilir duruma getirebilmesi teknolojinin hayatımıza ne kadar girdiği ve ne derecede katkı sağladığını göstermektedir.

Her geçen gün gelişen teknoloji ile birlikte, artan insan ihtiyaçlarını karşılama konusunda yeni ve farklı yöntemler ortaya çıkmıştır. Pandemi ile birlikte evlerden çalışırken de birçok işi dışarı çıkmadan yönetebilir duruma getirebilmek de teknolojinin getirilerindendir. Teknoloji sektörü, üretim sektörü gibi sektörlerle karşılaştırıldığında mekan sınırlaması olmaksızın üretime devam edilebilmesi de teknolojinin hayatımıza ne kadar girdiği ve ne derecede katkı sağladığını göstermektedir.

Genel olarak robot deyince aklımıza gelen, uzuvlarla insana benzetilmiş, ya da tamamen insan görünümünde olan robotların yanı sıra sanal çalışanlar (yazılımsal robotlar) da RPA teknolojisi ile işleri oldukça kolaylaştırmıştır. Bu sanal çalışanlar, bizler için sağlık, finans, sigorta, enerji, otomotiv, yeme-içme, lojistik gibi birçok alanda çalışabilmektedir. Belli periyotlarla yapılan işler, robotlara atandığında istenilen saatte aralıksız çalışabilmektedir. Örneğin sanal çalışanlarımız bizler için sistemdeki stokları kontrol edip, belirtilen miktarın altındaysa sipariş geçebilir, kişiye özel hediye kartları düzenleyebilir, sigorta poliçelerini okuyup müşteri ile iletişime geçebilir, belgeleri isimlendirip arşivleyebilir, işe giriş ve çıkış için sisteme kayıt yapabilir, faturaları okuyup gerekli alanları alıp işleyebilir. Farklı teknolojilerle birleşerek, müşteriden gelen opsiyonel süreçleri de otomotize edebilir.

Sanal asistanların hayatımıza bu kadar hızlı girmesi iş kayıpları korkusunu da beraberinde getirse de aslında gereksiz işleri çalışanın üzerinden alarak, çalışanın potansiyel verimliliğini arttıracak yeni görevler ve meslekler de ortaya çıkarmıştır. İş kayıplarıyla ilgili yaygın korkuların aksine, Dünya Ekonomik Forumu tarafından otomasyonun net 58 milyon iş artışıyla sonuçlanacağı tahmin

edilmektedir. (https://www.weforum.org/agenda/2021/02/world-economic-forum-automation-create-jobs-employment-robots/, 2021)

Dolayısı ile bu korkuları bir kenara bırakıp gelişen teknolojiye hızlı bir şekilde uyum sağlamak, işletmelerin sektördeki yerini sağlamlaştırmaya, hızlı, güvenli, hata payı azaltılmış bir şekilde hizmet vermeye, akabinde müşteri memnuniyetin artışı ve karşılıklı kazanımlara olanak sağlayacaktır.

8.5 Kaynaklar

(2021). www.automationanywhere.com

(2021). www.blueprism.com

(2021). https://www.gartner.com/en/documents/4001926-market-share-analysis-robotic-process-automation-worldwi. adresinden alındı

Çalışkan, L. S., & Kıran2, S. (2020). İş Süreçlerinin Otomasyonunda Rso'nın Faydaları. Yönetim Bilişim Sistemleri Dergisi, 13. Dünya Ekonomik Forumu, İ.

Kaya, C. T., Turkyılmaz, M., & Bırol, B. (2019). Impact of RPA Technologies on Accounting Systems. Muhasebe ve Finansman Dergisi, 15. Kestane, A. (2021). İç Denetimde Akıllı Otomasyon Teknolojilerinin Kullanımı: Robotik Süreç Otomasyonu Ve Bilişsel Zekâ. Muhasebe ve Vergi Uygulamaları Dergisi, 22.

powerautomate.microsoft.com. (2021).

www.gartner.com/en/finance/trends/robotic-process-automation/building-a-robotics-roadmap

www.uipath.com. (2021).

YETİZ, F., Turan, Y., & Canpolat, İ. (2021). Bankacılık Sektöründe Robotik Süreç Otomasyonu Ve Verimlilik İlişkisi: Bir Banka Örneği. Verimlilik Dergisi / Journal of Productivity, 15.

(https://www.weforum.org/agenda/2021/02/world-economic-forum-automation-create-jobs-employment-robots/, 2021)

https://developer.automationanywhere.com/challenges/automationanywherelabs-supplychainmanagement.html

https://developer.automationanywhere.com/challenges/AutomationAnywhereLabs-POTrackingLogin.html

python powered

print("Hello, world!")

9. Sosyal Medyadan Veri Çekme Örnekleri

Python ile Sosyal Medyadan Veri Çekme Örnekleri

Tijen ÖVER ÖZÇELİK*, **Alpaslan KİBAR***, **Muhammed Emin BAL***

***Sakarya Üniversitesi**

9.1 Giriş

Günümüzde veri hacmi ve veri çeşitliliği, daha önce hiç görülmemiş bir hızla artış göstermiş, özellikle internet teknolojileri, mobil teknolojiler ve sosyal medyanın hayatın her evresine girmesiyle, sadece işletmeler değil insanlar da günlük faaliyetlerinde veri üretir duruma gelmiştir. Veri artışına yönelik geliştirilen bir hipoteze göre, 1900'lere kadar bilgi, yaklaşık her yüzyılda ikiye katlanırken, 1950'ler den sonra bu süre 25 yıla, günümüzde ise neredeyse bir yıla düştüğü iddia edilmektedir. Hatta IBM'in yaptığı bir araştırmaya göre, nesnelerin internetinin artması ile gelecek yıllar da her 12 saatte bir bilginin ikiye katlanmasının söz konusu olacağı iddia edilmektedir. Verinin ve bilginin bu hızlı artışı ile birlikte bir diğer önemli konu da verinin işlenmesi ve analiz edilmesi, kısaca veriden üretilen "Bilgi ve Değer" le ilgilidir. Bu çalışmada gerçekleştirilen uygulamanın temel yapı taşı "veri"dir. Veri, "tek başına bir anlam ifade etmeyen veya kullanılamayan ama bilgiye temel oluşturan ilişkilendirilmeye, gruplandırılmaya, yorumlanmaya, anlamlandırılmaya ve analiz edilmeye gereksinim duyulan ham bilgi" şeklinde tanımlamıştır. Veriler genellikle; Yapılandırılmış, Yarı yapılandırılmış ve Yapılandırılmamış şeklinde gruplanmaktadır. Verinin bu yapılandırılmıştan yapılandırılmamışa doğru seyrinde büyük veri gündeme gelmekte ve veri analitiği çalışmalarına gereksinim duyulmaktadır.

Günümüzde verinin birçok kaynaktan, farklı şekillerde ve katlanarak üretilmesi "Büyük Veri" kavramını gündeme getirmiştir. Büyük veri kavramı ile ifade edilmek istenen, verinin sahip olduğu yüksek hacim, hız ve çeşitliliktir. Birçok farklı sektör ve işletme de yeni veriler üreterek büyük veri için kaynak durumuna gelmiştir. İşletmelerin rekabet avantajı sağlamada bilginin önemli bir faktör olarak görülmesi, karar verme süreçlerini etkilemesi, verinin ileri analitik yöntemlerle işlenmesi

gerekliliğini gündeme getirmiş ve büyük veri analitiğinin önemini artırmıştır. Bu amaçla kullanılan yöntemlerden bazıları; Veri Madenciliği ve Metin Madenciliğidir. Çalışmada temel alınan yöntem olan Metin Madenciliği, özellikle web teknolojileri ile üretilen metin halindeki büyük verinin analiz edilmesine yönelik istatistiksel bir tekniktir.

Kısacası, büyük verinin günümüzde bu kadar önemli bir konuma gelmesindeki temel neden; küresel olarak tüm işletme ve kurumların ürettiği ürün ve hizmetlerde internet tabanlı teknolojileri kullanması ile artan ve değerlendirilmeyen verilerin öneminin anlaşılmasıdır. Bu bağlamda çalışmada, büyük veri konusunu ve işlenmesini mümkün kılan teknolojiler, verinin ön işlemesi, temizlenmesi, birleştirilmesi, dönüştürülmesi, indirgenmesi ve python ile web sitelerden veri çekme örneklerine yer verilmiştir.

9.2 Veri ve Veri Tabanı

Deney, gözlem, sayım, ölçme vb. yöntemler ile elde edilen, işlenmemiş, enformasyon (bir konuda hakkında toplanmış bilgi parçası) parçacığına verilen isimdir (Paul BOCIJ, 2008). Türk Dil Kurumunun (TDK) sözlüğüne göre ise veri, "Bir araştırmanın, bir tartışmanın, bir muhakemenin temeli olan ana öge, muta, done." (TDK, 2021) olarak tanımlanmaktadır.

Örneğin, covid-19 hakkında yapılacak araştırmalar için, hastaların yaş, cinsiyet, kilo, sahip olunan kronik hastalıklar vb. veriler ile çalışılabilir.

Veri, yapısına göre genellikle "Yapısal, Yarı Yapısal ve Yapısal Olmayan" şeklinde üç şekilde sınıflandırılmaktadır. Yapısal veriler, MSSQL, DB2, Oracle vb. Veri Tabanı Yönetim Sistemleri (VTYS) tarafından, veriler arasındaki ilişkilere dikkat edilerek tutulan, gerektiğinde kolaylıkla erişilip üzerinde sorgular ile çalışılabilen (seçme, değiştirme, ekleme, silme vb.) verilerdir. Örneğin, bir üniversitenin öğrencilerinin bilgilerinin saklandığı birbirleri ile ilişkili tablolar ve bu tablolardaki alanlarda saklanan verilerden oluşan bir veri tabanı yapısal verileri içermektedir. Karar Destek Sistemlerinde (KDS) yapısal veriler ile çalışılarak (örneğin SQL sorgularıyla), yöneticilere karar önerileri oluşturmak, yarı yapısal ve yapısal olmayan verilere göre çok daha kolaydır. Bunun nedeni yapısal olmayan verilerin birbirleri ile ilişkili herhangi bir düzene sahip olmamalarıdır. Ayrıca yapısal olmayan verilerin boyutu genellikle büyük olur. Boyutu büyük olan ve aralarında ilişki bulunmayan verilerden de anlam dolayısıyla karar üretmek zor olabilmektedir. Örneğin videolar, fotoğraflar, uzun metin dosyaları, loglar vb. yapısal olmayan verilerdendir. Yarı yapısal veriler ise yapısal veri ile yapısal olmayan verinin bir çeşit karışımı gibidir. Veriler yapılandırılmamışlardır ancak NoSQL gibi sistemler sayesinde XML, JSON, web servisleri, HTML5 etiketleri vb. formatlardaki veriler işlenebilirler. Örneğin bir XML belgesinde, hastanın isim verisinin ne olduğu, yaş verisinin değeri, cinsiyet verisinin nerede saklandığı XML etiketleri sayesinde kolaylıkla belirlenebilir.

1960 'lı yılların başında ilk veri tabanı CODASYL (Conference/Committee on Data Systems Languages) ismiyle Hiyerarşik Modeli (parent/child) ile kullanılmaya başlanmıştır.

1970 'lerde ilişkisel veri tabanı yönetim sistemleri olan (RDBMS) System R IBM tarafından oluştururken 1979 'da Oracle Şirketi tarafından Oracle geliştirildi.

1980 'lerde ilişkisel veritabanları oldukça yaygınlaşmaya başladılar ki Microsoft MSSQL' i 1980 'lerin sonunda piyasaya sürmüştür. Yine bu yılarda SQL dili, ilişkisel veri tabanlarında standart sorgulama dili (Microsoft T-SQL, Oracle PL-SQL) hale gelmiştir.

1990 'larda MySQL, PostgreSQL VTYS' lerinin yanısıra Nesne Tabanlı Veri Tabanı Yönetim Sistemi (Object Database Management Systems - ODBMS) modeli ortaya çıkmıştır.

2000 'lerde bilhassa 2010 'lardan sonra internet kullanımın yaygınlaşmasıyla ortaya çıkan büyük miktardaki yarı yapısal/yapısal olmayan verinin (bigdata) ihtiyaçlarına cevap verebilmek amacıyla

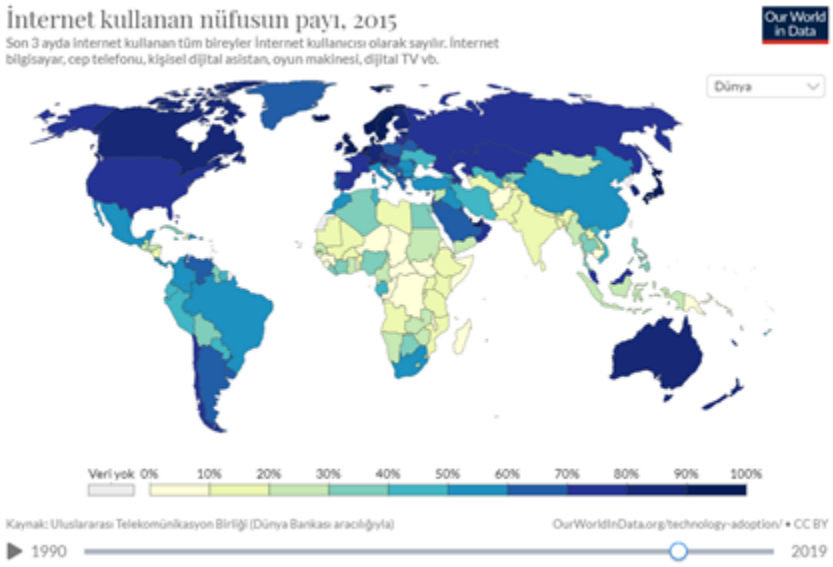
NoSQL modelleri kullanılmaya başlanmıştır. Birçok NoSQL modeli olmasına rağmen çoğunlukla ortak özellikleri;

- Genellikle açık kaynak kodları kullanmaları,
- Yatay mimari ile kolayca ölçeklenebilmeleri ve
- Veri tiplerinin önceden belirlenmesi zorunluluğunun

olmaması şeklinde sıralanabilir.

9.3 Veri Kaynakları

2000 yılından önce, firmaların büyük çoğunluğu çalışmalarında internet kaynaklı verileri kullanmadıklarından çok büyük oranda ilişkisel veri tabanı modeli kullanan VYTS' leri kullanıyorlardı. Gerekli veriler firmaların sunucularından sağlanıyorlardı. Gerektiğinde donanımlara CPU, RAM takviyeleri ile donanım ihtiyaçları kolaylıkla karşılanabiliyordu.

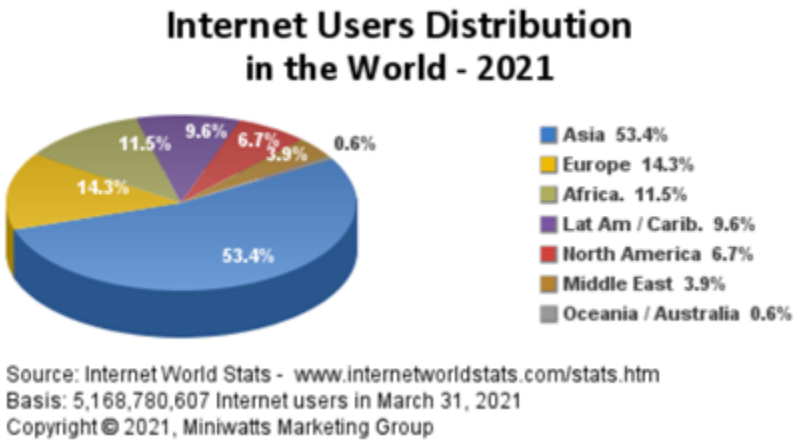


Şekil 9.1: Makine Öğrenmesi Sürecinin Akışı

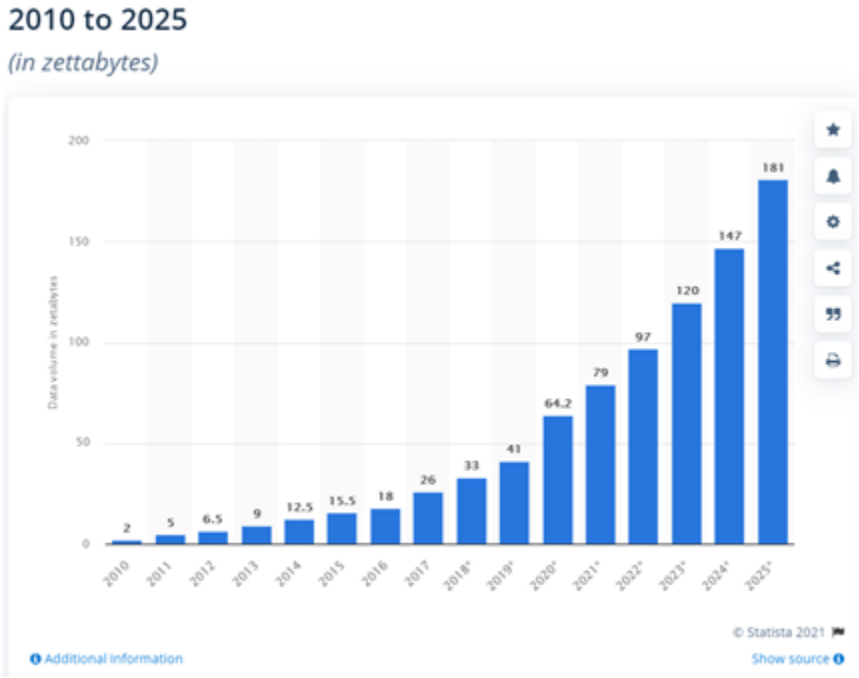
Şekil 1 'de Türkiye'de yaşayanların internet kullanım oranı 2015 yılında %53 olarak görünürken Türkiye İstatistik Kurumu (TÜİK) tarafından yayınlanan 2021 araştırmasına (TÜİK, 2021) göre Türkiye' de internet erişim oranı %92'ye yükselmiştir.

Global Web Index' e göre 31 Mart 2021 tarihinde dünyadaki internet kullanıcılarının sayısı 5 milyar 168 milyon 780 bin 607' ye yükselmiştir (GWI, 2021). Şekil 2'de Dünyada internet kullanıcılarının sayısı grafik olarak gösterilmiştir. Grafiklerden de görüldüğü üzere internet kullanım oranı oldukça artmaktadır.

Statista' ya göre üzere internet üzerinde oluşturulan, kopyalanan, tüketilen verinin zettabyte türünden hacmi yıllar geçtikçe artmaya (STATISTA, 2021) devam etmektedir. Şekil 3'de Dünyada yıllara göre internet üzerinde oluşturulan, kopyalanan, tüketilen verinin zettabyte türünden hacmi gösterilmiştir. 2020 yılında dünyada kişi başına üretilen veri saniyede ortalama 1.7 MB olarak belirlenmiştir (BULAO, 2021). Yine 2022 de Dünya ekonomisinin %70' inin dijitalleşmeden geçeceği tahmin edilmektedir. 2021 'de Instagram kullanıcılarının %68' i markaların fotoğraflarını görüntülemektedir. 2020 'de kullanıcılar günde ortalama 500 milyon tweet göndermiştir. 2021 Ocak



Şekil 9.2: Dünyada internet kullanıcılarının sayısı



Şekil 9.3: Dünyada internet kullanıcılarının sayısı

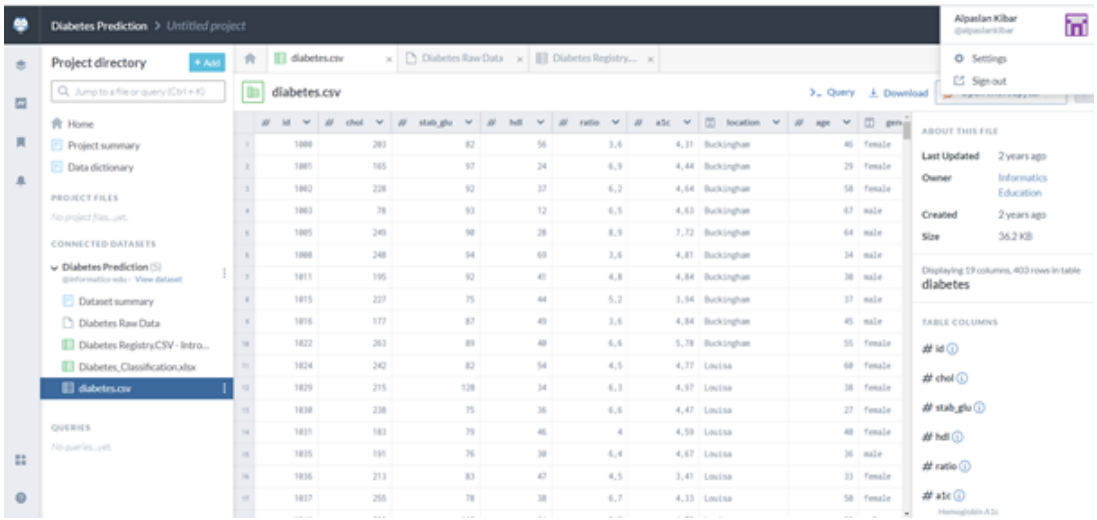
ayında dünyada aktif 4.66 milyar internet kullanıcısı bulunmaktadır. 2020 yılında internete 319 milyon yeni kullanıcı eklenmiştir. Sadece 2021 yılı boyunca 2 trilyon Google araması yapılmış olması beklenmektedir. Bu da günlük yaklaşık 6 milyar aramaya denk gelmektedir.

Bu bilgiler ışığında en büyük veri kaynağı olarak nitelenebilecek internetten ve diğer cihazlardan metin, işlem ayrıntıları (loglar), video, resim, konum bilgileri vb. türlerdeki verilerin web sayfaları, sosyal ağlar vb. ortamlardan sağlanabileceği söylenebilir.

Veriler sitelerden yazılımlar sayesinde alınabilecekleri gibi paylaşılan hazır veri setleri de kullanılabilir. Örnek veri seti paylaşım siteleri:

UCI Machine Learning Repository (https://archive.ics.uci.edu/ml/index.php); Makine öğrenmesi çalışmaları için web üzerinden veri seti paylaşan en eski sitelerdendir.

Data.World (https://data.world/); Çok geniş veri setlerine sahiptir. Örneğin 20 Ağustos 2021 tarihinde Covid-19 ile alakalı, Amerika Birleşik Devletlerinin CDC (Centers for Disease Control and Prevention) kurumunun paylaştığı veri setinde (Şekil 4) 20 milyondan fazla kayıt (yaklaşık 4 GB yer kaplıyor) bulunmaktaydı.



Şekil 9.4: Veri seti paylaşım sitelerinden data.word' de diabet hastalığı ile ilgili veri setlerinden birinin paylaşım ekranı

Kaggle (https://www.kaggle.com/datasets); Çok değişik alanlarda çok kapsamlı veri setlerinin paylaşıldığı bir başka platformdur.

Veri Setleri neden paylaşılır

Veriyi paylaşanlar ile veriyi kullananlar arasında bilimsel işbirliği fırsatları ortaya çıkabilir. Paylaşılan veriler aynı alanda yapılan birçok araştırma ile araştırma sonuçlarının doğrulanmasına yardımcı olurlar. Paylaşılan veriler eğitim/öğretim amaçlı kullanıldığında öğrenciler gerçek veriler üzerinde çalışabilmiş olurlar, sahte veri üretimi gibi sakıncalı yöntemlere karşı faydalı olur. Veriyi paylaşan kurumların şeffaflık ve hesap verilebilirliğini arttırır. Bunun yanında Avrupa Komisyonu gibi birçok araştırma fonu sağlayan kurum verilerin paylaşımını zorunlu tutmaktadırlar.

9.4 Veri Ön İşleme

İnternetteki verilerin en büyük problemi verilerin büyük kısmının yapısal olmaması veya yarı yapısal olmasıdır. Ayrıca veriler arasında yanlış olanlar, eksik olanlar, gereksiz olanlar vb. bulunabilir. Bu nedenle yapılacak çalışmalar öncesinde bu veriler üzerinde veri ön işleme adımları uygulanmalıdır. Bu veri ön işleme aşamaları araştırmacılar tarafından manuel yapılabileceği gibi bu işlemleri yapan yazılımlar veya paket programlar da kullanılabilir. Veri ön işleme teknikleri şu şekilde sıralanabilir (OĞUZLAR, 2003):

- Veri Temizleme
- Veri Birleştirme
- Veri Dönüştürme
- Veri İndirgeme

9.5 Veri Temizleme

Veri kaynağından elde edilen kayıtların bazı alanlarının (değişkenler) verileri eksik olabilir. Bu durumda verinin özelliğine, yapılan çalışmanın ortak kabullerine göre kaydı yok sayma, eksik veriyi doldurma (elle sezgisel olarak, frekansı en yüksek bir değerle, ortalamayla, en olası değerle) gibi yöntemler kullanılabilir. Kaydı yok sayma yani kaydın çalışmadan çıkarılması/silinmesi yönteminin en büyük tehlikesi o kayıttaki verilerin araştırma için çok önemli olmasıdır. Silinen bu verilerin sonuca etkileri önceden kestirilebilmelidir.

Veri setinde yanlış girilmiş veriler var ise (örneğin yaş -33, çoğu kişinin doğum günü 1 Ocak), verilerde tutarsızlıklar var ise (yaş 44, doğum yılı 1960) bu veriler üzerinde de düzeltme yapılmalıdır.

Veri temizleme yöntemlerinin kullanılabildiği bir diğer sorun ise gürültülü verilerdir. Gürültü, elde edilen verideki varyans veya rassal bir hatadır. Bu tür gürültülü verilerin tespiti için binning, kümeleme analizi, regresyon gibi yöntemler kullanılabilir.

Binning

Küçükten büyüğe veya büyükten küçüğe sıralanmış verileri içeren veri setlerinde kullanılabilen bir tekniktir. İşlemin adımları:

- İlk adımda sıralanmış veriler eşit sayıda veri içeren bin'lere dağıtılır.
- İkinci adımda bin' ler bin ortalama, bin medyan veya bin sınırları kullanılarak düzeltilir.

Örneğin veri seti 2, 5, 7, 10, 15, 18, 21, 26, 30, 35, 36, 47 değerlerinden oluşsun.

Bin' ler eşit sayıda veri içereceğinden,

1. Bin: 2, 5, 7, 10
2. Bin: 15, 18, 21, 26
3. Bin: 30, 35, 36, 47

Olarak oluşturulur. Bin' ler ortalama ile düzeltilecek ise,

1. Bin: 4, 4, 4, 4
2. Bin: 20, 20, 20, 20
3. Bin: 37, 37, 37, 37

olmalıdır.

Kümeleme

Birbirine benzer, yakın değerler aynı kümede/grupta olacağından, aykırı değerler de kümelerin dışında kalacaklardır. Böylece aykırı değerler belirlenmiş olacaktır.

Regresyon

Verilere uygun regresyon fonksiyonu oluşturulduğunda bu fonksiyona uymayan noktalar/değerler aykırı değerler olarak belirlenebilirler.

Web Tabanlı Veriler için Veri Temizleme Web sayfalarından elde edilen verilerin temizlenmesi aşağıdaki adımlar önerilir.

- Gerekli olmayan tüm etiketler kaldırılır, sadece ihtiyaç duyulan etiketler belirlenir.
- HTTP durum kodu 299 dan büyük olan veya 200 den küçük olanlar kaldırılır. http kodu 200 ile 299 arasında olanlar başarılı sonuç döndürenlerdir.
- POST veya HEAD metodu kullanan tüm kayıtlar kaldırılmalıdır, sadece GATE metodlu kayıtlar kalmalıdır.

Bu işlemler yapıldığında, tüm verilerden, araştırma ile alakasız olan %50-%60 arası kayıt temizlenmiş olur (LOSARWAR, 2012).

9.6 Veri Birleştirme

Farklı veri kaynaklarından elde edilen veriler, Meta Datalar kullanılarak, veriler arasında tutarsızlıklara (il bilgisinin bir kaynaktan 34 diğer kaynaktan İstanbul olarak alınması vb.) yer vermeyecek şekilde birleştirilir. Veri birleştirme sırasında ID kullanılarak yapısallaştırma sağlanmaya çalışılabilir. Araştırmadaki veri kaynaklarından erişilen veriler birleştirildiğinde ihtiyaç duyulandan daha fazla veri toplanmış olabilir. Bunun sonucu olarak kullanılan tekniklerin işlem performanslarında azalmalar olabilir. Kullanılan veri kaynaklarından elde edilen farklı alan/değişken ismine sahip ancak benzer verileri içeren aralarında yüksek korelasyon ilişkisi olan alan/değişkenlerden biri araştırmadan çıkarılabilir.

9.7 Veri Dönüştürme

Bu aşamada verinin daha kolay anlaşılabilmesi sağlanabilir. Veriler veri dönüştürme aşamasında, kullanılacak yöntem/yöntemler için daha uygun formatlara dönüştürülebilirler. Genelleştirme, normalleştirme, birleştirme bu dönüştürme yöntemlerinden bazılarıdır. Örneğin günlük satış miktarlarından aylık/yıllık satış miktarları hesaplanabilir. Alan/değişkendeki verilerin değerleri arasında çok fark var ise Min-Max normalizasyonu (0 – 100 arası), z Dönüşümü (-1 ile +1 arası vb.) gibi işlemler yapılabilir. Örneğin, öğrencilerin Ödev, Vize, Final notlarından Yıl Sonu başarı notları hesaplanabilir.

9.8 Min-Max Normalizasyonu

Veri dönüştürme aşamasında en sık kullanılan normalleştirme yöntemlerinden biridir. Verilerin değerleri çok büyük bir aralıkta değişirken, bu verilerin değerlerini -1 ile +1 arasında, 0 ile +1 arasında gibi, daha küçük aralıkta göstererek, verinin daha kolay anlaşılması ve işlenmesi sağlanmaya çalışılır.

Örneğin veri seti 10, 15, 25, 80, 130, 350, 360, 470, 780, 810 değerlerinden oluşsun. Bu verilerin en küçük 0 (sıfır) en büyük 1 (bir) değerini alabilecek şekilde dağılması istenirse;

VyeniDeger=(V-Mineski)/(Maxeski-Mineski)*(Maxyeni-Minyeni)+Minyeni formülü ile verilerin yeni değerleri hesaplanabilir. Bu formülde;

VyeniDeger : Veri setindeki bir verinin yeni aralıkta alacağı yeni değeri gösterir.

V : Veri setindeki bir verinin eski aralıktaki, işlemden geçmemiş ilk değerini gösterir.

Mineski : Veri setindeki işlenmemiş en küçük verinin değerini gösterir.

Maxeski : Veri setindeki işlenmemiş en büyük verinin değerini gösterir.

Maxyeni : Yeni aralıktaki en büyük verinin değerini gösterir. Örneğin verilerin değerlerinin 0 ile 1 arasında dağılması isteniyorsa bu değer 1 (bir) olacaktır.

Minyeni : Yeni aralıktaki en küçük verinin değerini gösterir. Örneğin verilerin değerlerinin 0 ile 1 arasında dağılması isteniyorsa bu değer 0 (sıfır) olacaktır.

10 değerine sahip ilk veri için formül kullanıldığında,

VyeniDeger = (10-10)/(810-10)*(1-0)+0 = 0

80 değerine sahip dördüncü veri için formül kullanıldığında,

VyeniDeger = (80-10)/(810-10)*(1-0)+0 = 0.0875

360 değerine sahip yedinci veri için formül kullanıldığında,

VyeniDeger = (360-10)/(810-10)*(1-0)+0 = 0.4375

810 değerine sahip sonuncu veri için formül kullanıldığında,

VyeniDeger = (810-10)/(810-10)*(1-0)+0 = 1

yeni değerleri elde edilecektir. Çalışmada 10, 15, 25, 80, 130, 350, 360, 470, 780, 810 değerlerinin yerine, 0, 0.00625, 0.01875, 0.0875, 0.15, 0.425, 0.4375, 0.575, 0.9625, 1 değerleri kullanılacaktır.

9.9 Veri İndirgeme

Her çalışmada eldeki tüm verilerin kullanılması gerekli olmayabilir. Gereğinden daha büyük hacme sahip bir veri seti Zaman ve Maliyet olarak ek yük getirebilir. Temel Bileşen Analizleri yapılarak sonuca etkisi önemsenmeyecek kadar az olan değişkenlerin verileri çalışmadan çıkarılabilir. Sonucu benzer şekilde etkileyecek, aralarında yüksek korelasyon ilişkisi bulunan değişkenlerden bazıları da çalışmadan çıkarılabilir. Gerekli olan durumlarda evreni temsil edecek şekilde benzer sonuçlara ulaşılmasını sağlayacak oranda kayıt sayısında da azaltmaya da gidilebilir. Sonuç olarak asıl hedef, büyük hacimli veri kümesi ile çalışmak yerine daha küçük hacme sahip, daha duru, daha sade fakat çalışmanın amacına uygun sonuçları üretebilecek veri kümesine ulaşmaktır.

9.10 Sosyal Medyadan Veri Çekme Örnekleri

Bir önceki bölümde internet sitelerinde hazır veri setlerinin paylaşıldığından bahsedilmişti. Bazı durumlarda bu paylaşılan veri setleri yerine anlık, güncel, istenilen özelliklere sahip verilerin toplanması gerekebilir. Bu tür durumlarda kullanılabilecek birçok yöntem vardır. Bu çalışma kapsamında, emlakjet ve twitter'dan, python programlama dili ile sosyal medyadan veri çekme yöntemlerinden Requests ile veri çekme ve Selenium ile veri çekme örnekleri uygulamalı olarak gösterilmiştir.

9.11 Kullanılacak Kütüphaneler

Python dili ile büyük veri (big data) üzerinde çalışırken kullanılabilecek çok sayıda komut kütüphaneler altında toplanmışlardır. Burada veri çekmek için kullanılacak iki yöntemde tanımlanan kütüphanelerden ve bu kütüphanelerin kullanım amaçlarından/avantajlarından bahsetmekte fayda olacaktır.

Requests Kütüphanesi

- Bu kütüphane ile web sitelerine istekler gönderilebilir,
- Bu amaçla get, post gibi komutlar kullanılabilir,
- Çok kullanışlı/avantajlı bir kütüphanedir ve

- İçeriği hedef alınan siteye login işlemleri oluşturulabilir.

Selenium Kütüphanesi

- Bilgisayara kurulabilecek bir driver ile tarayıcı açarak hedef alınan web sitelerine erişime ve oralarda gezintiye olanak sağlayabilir,
- Test işlemleri için de oldukça rağbet gören bir kütüphanedir,
- Kendine özgü veri alma yöntemleri vardır ve
- Requests kütüphanesine göre oldukça yavaş olabilse de hedef sitenin özelliklerine göre tercih edilebilir.

BeautifulSoup

Alınan verilerin istenilen parçalara ayrılması için kullanılabilir.

Lxml

HTML etiketlerinin BeautifulSoup ile parçalanması aşamasında lxml komutları kullanılabilir.

Time

Selenium kütüphanesi ile hedef siteye erişim sağlanıldığında sitenin yüklenmesi beklemek yani kodu bekletmek için kullanılabilen komutları içerir.

Pandas

Verilerin kolaylıkla işlenmesi, xml, csv vb. formatlara kaydedilebilmesi vb. komutları içerir.

Sık Karşılaşılan http Durum Kodları

http durum kodları sayesinde, hedef siteye istek gönderildiğinde, isteğin başarılı olup olmadığı hakkında bilgi edinilebilir. Her ne kadar çok fazla durum kodu olsa da en sık karşılaşılanlar şunlardır:

- HTTP 200: Gönderilen isteğin başarılı olduğunu gösterir.
- HTTP 403: Siteye ulaşılıyor ancak site erişim izni vermiyor.
- HTTP 404: Hedef sayfa bulunamıyor.
- HTTP 500: İstek gönderiliyor ancak istek tamamlanamıyor.

Requests Örneği

Öncelikle kullanılacak kütüphaneler projeye alınır.

```
import requests
from bs4 import BeautifulSoup
import pandas as pd
```

Hedef sitenin adres bilgileri değişkenlere atanır.

```
baseurl = "https://www.emlakjet.com/vitrin/"
url_head = "https://www.emlakjet.com"
```

Hedef adrese istek gönderilir, gelen veriler BeautifulSoup ile istenilen şekilde düzenlenir. url parametresinin veri tipi string' dir.

```
def get_url(url):
        r = requests.get(url)
        soup = BeautifulSoup(r.content, "lxml")
        return soup
```

Sayfadaki bir etiket toplanır.

```
def scrape_url(soup):
        columns = soup.find_all("div", attrs={"class": ...
                "styles_gridColumn__1hxWa"})
        url_list = [url_head + col.a.get("href") for col in columns]
```

```
        return url_list
```

Diğer sayfalara da geçilerek bir etiket toplanır. last_page parametresinin veri tipi int, baseurl değişkeninin veri tipi string' dir.

```
def scrape_all_page(last_page, baseurl):
        home_url_list = []
        for page in range(1, last_page + 1):
                print(page)
                mainurl = baseurl + str(page)
                print(mainurl)
                soup = get_url(mainurl)
                home_url_list.append(scrape_url(soup))
                print(home_url_list)
                print("************")
        return home_url_list
```

Oluşturulan listeler tek bir listeye dönüştürülürler. liste parametresinin veri tipi list' dir.

```
def convert_single_list(liste):
        single_list = [item for elem in liste for item in elem]
        return single_list
```

Etiketlerin özellikleri toplanır. home_urls parametresinin veri tipi list' dir.

```
def scrape_property(home_urls):
        home_info = []
        for ln in home_urls:
        soup = get_url(ln)
        title = ""
        price = ""
        property_list = []

        try:
        title = soup.find("h1", attrs={"class": ...
                "styles_detailTitle__qBXKm"}).text.strip()
        price = soup.find("div", attrs={"class": ...
                "styles_price__1e65F"}).text.strip()
        property_area = soup.find("div", attrs={"class": ...
                "styles_properties__12d_v"})
        property_list = [data.text.strip() for data in ...
                property_area.find_all("span")]
        except:
        pass

        home_info.append([ln, title, price, property_list])
        return home_info
```

Elde edilen veriler Excel dosyasına yazdırılır. propertylist parametresinin veri tipi list, name parametresinin veri tipi string' dir.

```
def save_excel(propertylist, name):
        df = pd.DataFrame(propertylist)
        df.columns = ["url", "title", "price", "properties"]
        df.to_excel(f"{name}.xlsx")

        urls = convert_single_list(scrape_all_page(3, baseurl))
        home_features = scrape_property(urls)
        save_excel(home_features, "home")
```

Selenium Örneği

Kullanılacak kütüphaneler projeye alınır.

```
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
import time
from bs4 import BeautifulSoup
```

Twitterdan girilen hesabın tweet linkleri toplanır.

```
        def collect_links(word, scrollnumber):
        driver_path = "D:\PROGRAM_SETUP\chromedriver.exe"
        driver = webdriver.Chrome(driver_path)
        driver.get("https://twitter.com/explore")
        time.sleep(3)
```

Arama çubuğuna aranacak kelime girilir.

```
search_box = driver.find_element_by_xpath("//*[@id='react-
root']/div/div/div[2]/main/div/div/div/div[1]/div/div[1]/div[1]
/div/div/div/div/div[1]/div[2]/div/div/div/form/div[1]/div/div/
label/div[2]/div/input")
        search_box.send_keys(word)
        time.sleep(3)
```

İlk gelen seçeneğe tıklanır.

```
enter = driver.find_element_by_xpath("//*[@id='typeaheadDropdown-1']
/div[3]/div")
enter.click()
time.sleep(3)
tweetlistlink = []
for i in range(1, scrollnumber):
page = driver.page_source
soup = BeautifulSoup(page, "lxml")
tweet_column = soup.find("div", attrs={"class":
        "css-1dbjc4n"})
tweets = tweet_column.find_all("div", attrs={"class":
        "css-1dbjc4n r-j5o65s r-qklmqi r-1adg3ll r-1ny4l3l"})
twitter_head = "https://twitter.com"
for j in tweets:
```

```
tweeturllast = j.find("a", attrs={
"class": "css-4rbku5 css-18t94o4 css-901oao
r-14j79pv r-1loqt21 r-1q142lx r-37j5jr r-a023e6 r-16dba41
r-rjixqe r-bcqeeo r-3s2u2q r-qvutc0"}).get(
"href")
print(twitter_head + tweeturllast)
tweetlistlink.append(twitter_head + tweeturllast)
```

Sayfanın kaydırılabilmesi için Scroll özelliği kullanılabilir.

```
driver.execute_script("window.scrollTo(0,5000);")
time.sleep(3)
return tweetlistlink
```

Tweetlerin içerikleri alınır.

```
def getcontent(liste):
driver_path = "D:\PROGRAM SETUP\chromedriver.exe"
driver = webdriver.Chrome(driver_path)
time.sleep(4)
content = []
for ln in liste:
driver.get(ln)
time.sleep(3)
page = driver.page_source
soup = BeautifulSoup(page, "lxml")
tweettext = soup.find("div", attrs={"class":
        "css-1dbjc4n r-1s2bzr4"}).text.strip()
date = soup.find("a", attrs={
"class": "css-4rbku5 css-18t94o4 css-901oao
css-16my406r-14j79pv r-1loqt21 r-poiln3 r-bcqeeo r-qvutc0"})
text.strip()
user = soup.find("div", attrs={
"class": "css-901oao css-bfa6kz r-1awozwy
r-18jsvk2 r-6koalj r-37j5jr r-a023e6 r-b88u0q r-rjixqe r-bcqeeo
r-1udh08x r-3s2u2q r-qvutc0"}).text.strip()
content.append([ln, user, tweettext, date])
return content
```

Toplanan veriler excel belgesine yazdırılır.

```
tweetlistlink = collect_links("deep learning turkiye",3)
list_data = getcontent(tweetlistlink)
df = pd.DataFrame(list_data)
df.to_excel("tweet_data.xlsx")
```

En büyük veri kaynaklarından olan internetten veri toplamanın çok farklı yolları bulunmakta ve veri özellikleri dikkate alınarak yöntem tercihleri yapılabilmektedir. Bu çalışmada kullanılabilecek yöntemlerden sadece ikisine yer verilmiştir. Bazı çalışmalarda araştırmacıların aradıkları verilere çok kolay ulaşamayacakları iddia edilmektedir, fakat bu çalışmada belli teknolojiler yardımı ile bazı

verilerin, çok da zor olmayan yöntemlerle, ulaşılabilir olduğu gösterilmeye çalışılmıştır. Ayrıca elde edilen verilerin doğrudan kullanımı yerine veri ön işleme teknikleri yardımı ile araştırmanın ihtiyacına göre hazırlanması üzerinde durulmuştur. Günümüzde birçok çalışma ve işletme veriye ihtiyaç duymakta ve yanlış veri ise çözüm algoritması doğru olsa bile araştırmayı yanlış sonuçlara ulaştırmaktadır.

9.12 Kaynaklar

BULAO, J. (2021, Eylül 1). How Much Data Is Created Every Day in 2021? techjury.net: https://techjury.net/blog/how-much-data-is-created-every-day/#gref adresinden alındı

GWI. (2021, Ekim 10). Internet World Stats - Usage and Population Statistics. Global Web İndex: https://www.internetworldstats.com/stats.htm adresinden alındı

LOSARWAR, V. D. (2012). Data Preprocessing in Web Usage Mining. International Conference on Artificial Intelligence and Embedded Systems (ICAIES'2012) (s. 15-16). Singapore: ICAIES'2012.

Max Roser, H. R.-O. (2021, 09 01). Internet. ourworldindata: https://ourworldindata.org/internet adresinden alındı

OĞUZLAR, A. (2003). VERİ ÖN İŞLEME. Erciyes Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, Sayı: 21, 67-76.

Paul BOCIJ, A. G. (2008). Business Information Systems: Technology, Development and Management for the E-Business (4th Edition) 4th Edition. Pearson Education Canada.

STATISTA. (2021, Eylül 01). Volume of data/information created, captured, copied, and consumed worldwide from 2010 to 2025. Statista: https://www.statista.com/statistics/871513/worldwide-data-created/ adresinden alındı

TDK. (2021, Eylül 01). Güncel Türkçe Sözlük. Türk Dil Kurumu Sözlükleri: https://sozluk.gov.tr/ adresinden alındı

TÜİK. (2021, Eylül 01). Hanehalkı Bilişim Teknolojileri (BT) Kullanım Araştırması, 2021. Türkiye İstatistik Kurumu: https://data.tuik.gov.tr/Bulten/Index?p=Hanehalki-Bilisim-Teknolojileri-(BT)-Kullanim-Arastirmasi-2021-37437 adresinden alındı

Facebook Kats ile Zaman Serisi Analizi ve Tahminleme

Yunus ÖZEN*

*Yalova Üniversitesi, Bilgisayar Mühendisliği, Yalova, Türkiye

10.1 Giriş

Zaman serileri, zamana göre indekslenmiş veriler kümesidir. Zaman serileri genellikle zamanda eşit aralıklı noktalar olarak işaretlenmiş dizilerdir. Zaman serilerine bir havayolu şirketinde bir hat yönünde uçan günlük yolcu sayısı, bir konumda kaydedilen saatlik hava sıcaklığı bilgileri, bir ürünün günlük satış miktarı, bir hisse senedinin borsadaki günlük kapanış değerleri gibi örnekler verilebilir. Zaman serileri bir takım dalgalanmaların etkisi altında kalırlar. Bunların başında trend, mevsimsel dalgalanmalar, konjonktürel dalgalanmalar ve düzensiz dalgalanmalar gelmektedir (Lütkepohl, 2005).

Zaman serilerinin analiz edilmesi ise istatistiksel özelliklerini çıkarmak için bu verilerin işlenmesine dönük metotlardan oluşmaktadır. Bunu yaparken geçmiş verilerden yararlanılır ve bu veriler arasında çeşitli ilişkiler aranır (Shumway vd, 2000). Zaman serileri üzerinde ileriye dönük tahminler yapmak için çeşitli modeller geliştirilmiş bulunmaktadır (Vishwas vd, 2020).

Bu bölüm kapsamında Facebook tarafından geliştirilmiş bulunan Kats kütüphanesi ile zaman serisi analizi, çeşitli istatistiksel özelliklerin çıkarılması ve geleceğe dönük tahminleme işlemleri açıklanmaktadır. Kütüphanede yer alan modeller örnekler üzerinden açıklanmakta ve mevcut metotlar hakkında bilgi verilmektedir. Kats çatısı aktif olarak geliştirmesi devam eden açık kaynaklı bir çatı olduğundan, gelecekte eklenecek işlevsellikler hakkında fikir vermekte ve çatı boyunca kullanılan kullanım şekilleri açıklanmaktadır. Bu sayede sonraki sürümlerde eklenecek olan işlevselliklerin de hızlı öğrenilmesi daha kolay olacaktır.

10.2 Facebook Kats Kütüphanesi

Kats kütüphanesi, zaman serisi analizi için Facebook tarafından geliştirilmiş ve açık kaynak olarak yayınlanmış bir çatıdır. Zaman serisi analizi, tahminleme, anomali tespiti, özellik çıkarımı gibi işlemleri yapmak için gerekli modelleri barındıran genel amaçlı, hafif bir çatıdır. Pek çok zaman serisi analizi ve tahmini kütüphanesi olmakla birlikte, Kats kütüphanesi zaman serisi verilerinin modellenmesi ve analizi ile ilgili bütün teknikleri bir arada toplayan kapsamlı bir kütüphane olması sebebiyle öne çıkmaktadır (Zhang vd, 2021).

Kats kütüphanesinin temel işlevlerini aşağıdaki şekilde gruplandırmak mümkündür:

- Tahminleme
- Uç değer ve değişim noktası tespiti
- Özellik çıkarımı

10.2.1 Temel Kats Veri Yapıları

Kats kütüphanesinin temel veri yapısı TimeSeriesData nesnesidir. TimeSeriesData nesnesini iki farklı şekilde ilklendirmek mümkündür.

- TimeSeriesData(df) şeklinde, içine bir Pandas.DataFrame göndermek suretiyle ilklendirmek. Gönderilen DataFrame içerisinde bir kolon "time" olarak isimlendirilmiş olmalıdır.
- TimeSeriesData(time, value) şeklinde ilklendirmek. time parametresi bir Pandas.Series ya da Pandas.DatetimeIndex olabilmektedir. value parametresi tekil veri için Pandas.Series, çoklu veri için ise Pandas.DataFrame olabilmektedir.

TimeSeriesData nesnesi, yaygın kullanılan Pandas kütüphanesinde olduğu gibi aşağıdaki listede yer alanları da içeren çeşitli şekillerde zaman verisi kabul edebilmektedir.

- datetime tipi.
- Pandas.TimeStamp.
- String veri tipi. Standart formatlardan birinde yazılmamışsa date_format parametresi ile birlikte kullanılmalıdır.
- Unix Epoch olarak kullanmak istenirse, int veri tipi.

Verinin okunması ve TimeSeriesData nesnesine dönüştürülmesi işleminin örnek kodları Şekil 10.1 üzerinde görülmektedir. Pandas DataFrame olarak okunan havayolu uçuş verileri TimeSeriesData nesnesine dönüştürülmüş ve çizdirilmiştir. Python'da yaygın kullanılan çizim nesnesi olan matplotlib yardımıyla üretilen şekil diske kaydedilmiştir.

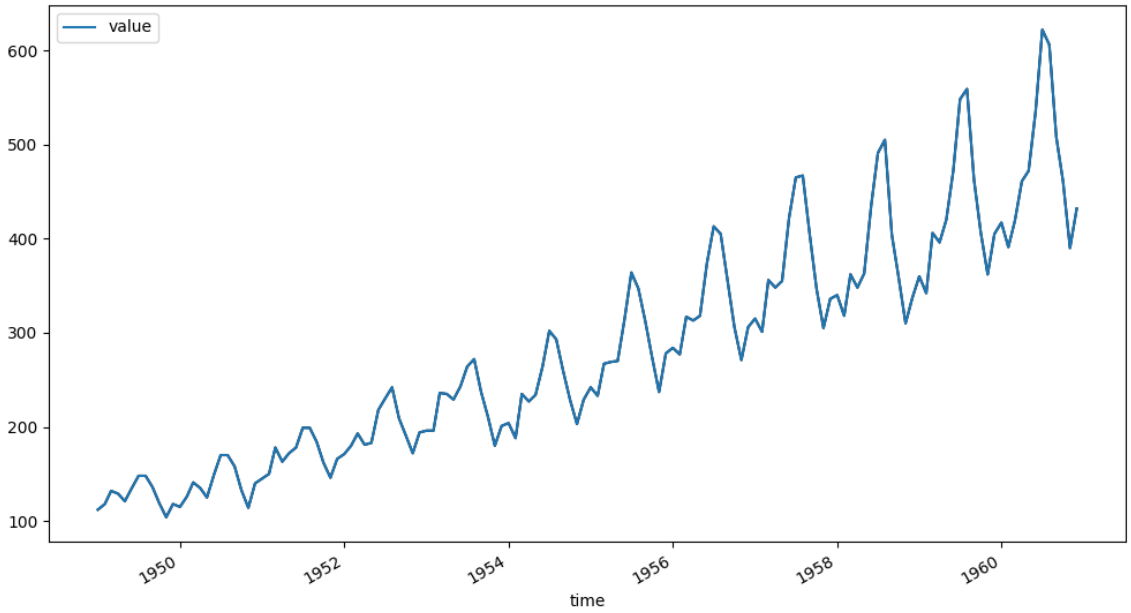
Çizdirilen havayolu yolcu verisi Şekil 10.2 üzerinde gösterilmektedir. Şekilde 1949-1960 yılları arasında Amerika Birleşik Devletleri'nde aylık uçuş yapan yolcu sayısı verisi gösterilmektedir. Ortalamanın belirli bir trend dahilinde sürekli artması ile birlikte uçan yolcu sayısının belli aylarda artıp belli aylarda azaldığı şekil üzerinde görülmektedir.

TimeSeriesData nesneleri üzerinde dilimleme yapmak, seriyi genişletmek, diziye dönüştürmek, DataFrame nesnesine dönüştürmek, matematiksel işlemler yapmak da mümkün olmaktadır. Şekil 10.3 üzerinde bu işlemlerin yapıldığı örnek kodlar görülmektedir.

Şekil 10.3́e yer alan kodların ekran çıktısı da Şekil 10.4 üzerinden görülmektedir.

```
1  import sys
2  import numpy as np
3  import pandas as pd
4  import matplotlib.pyplot as plt
5
6  from kats.consts import TimeSeriesData
7
8  df = pd.read_csv("data/air_passengers.csv")
9  df.columns = ["time", "value"]
10 ts = TimeSeriesData(df)
11 ts.plot(cols=["value"])
12 plt.savefig("tsdata_fig.png")
```

Şekil 10.1: TimeSeriesData kullanımı.



Şekil 10.2: Havayolu uçuş verileri.

```
1  import sys
2  import numpy as np
3  import pandas as pd
4  import matplotlib.pyplot as plt
5
6  from kats.consts import TimeSeriesData
7
8  df = pd.read_csv("data/air_passengers.csv")
9  df.columns = ["time", "value"]
10 ts = TimeSeriesData(df)
11 ts_from_series = TimeSeriesData(time=df.time,
   value=df.value)
12
13 ts_1 = ts[0:2]
14 ts_2 = ts[2:5]
15 ts_1_2 = ts_1.extend(ts_2)
16
17 print(ts[1:5])
18 print(ts[1:5] + ts[1:5])
19 print(ts == ts_from_series)
20 print(ts_1_2)
```

Şekil 10.3: TimeSeriesData işlemleri.

```
(venv) yunus@fb-kats-time-series-analysis$ python tsdata_ops.py
        time  value
0 1949-02-01    118
1 1949-03-01    132
2 1949-04-01    129
3 1949-05-01    121
        time  value
0 1949-02-01    236
1 1949-03-01    264
2 1949-04-01    258
3 1949-05-01    242
True
        time  value
0 1949-01-01    112
1 1949-02-01    118
2 1949-03-01    132
3 1949-04-01    129
4 1949-05-01    121
```

Şekil 10.4: TimeSeriesData işlemleri kodunun çıktısı.

10.3 Kats ile Tahminleme Yapmak

Zaman serileri üzerinde tahminleme yapmak için geliştirilmiş pek çok model bulunmaktadır. Kats kütüphanesi içerisinde, literatürde yaygın olarak kullanılan modeller yer almaktadır. Mevcut sürümde yer alan modeller aşağıda listelenmektedir:

1. ARIMA
2. SARIMA
3. AR-Net
4. Prophet
5. Holt-Winters
6. Theta
7. Linear
8. Quadratic
9. VAR
10. LSTM

Literatürde yaygın olarak kullanılan Scikit-Learn (Kramer, 2016) kütüphanesinin kullanım tarzı Kats'da da benimsenmiştir. Bu adımlar önce modelin oluşturulması, sonra da sırasıyla fit ve predict metotlarının çağrılması şeklindedir.

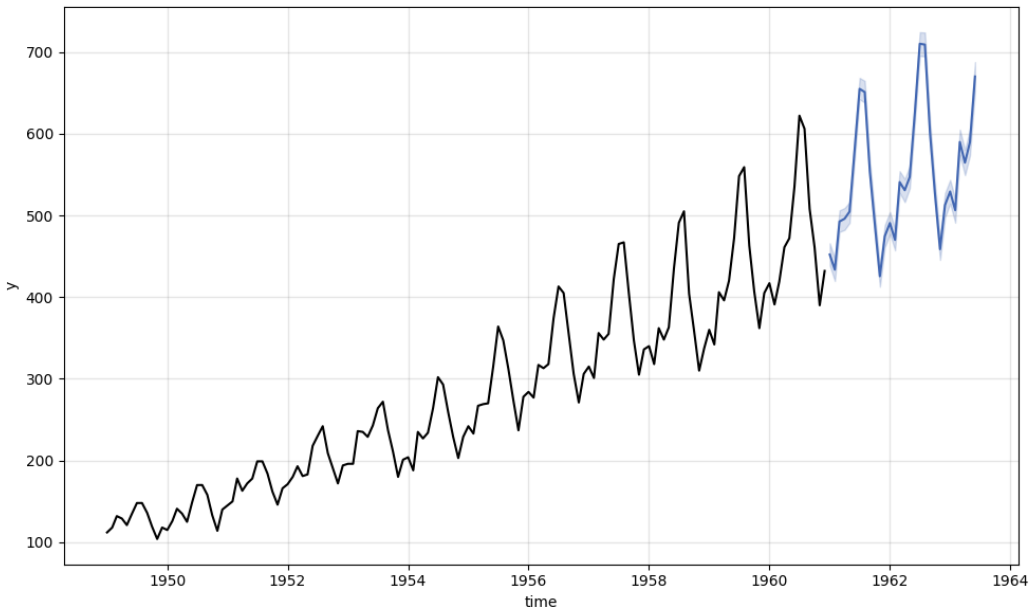
Prophet modeli (Toharudin vd, 2020) ile yapılmış bir tahminleme işleminin örnek kodları Şekil 10.5 üzerinde görülmektedir. kats.models.prophet modülünden ProphetModel ve ProphetParams sınıfları projeye import ile eklenmiştir. ProphetParams nesnesi oluşturularak modele ait parametreler doldurulmuştur. Bu parametreler ile ProphetModel nesnesi oluşturulmuştur. Mevsimsellik modu multiplicative olarak seçilmiştir. Üzerinde çalışılan datanın bilinmesi bu parametrelerin doğru ayarlanması için önemlidir. Model sırasıyla fit ve predit metotları çağrılarak eğitilmiştir ve örnekte verilen şekliyle önümüzdeki 30 ay için tahminleme yapılmıştır.

Mevcut verinin ve tahminleme ile üretilmiş verinin grafik üzerinde gösterilmesi için Kats içerisinde yer alan bütün modeller gibi ProphetModel de bir plot metoduna sahiptir. Böylece sonuç

```python
import sys
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

from kats.consts import TimeSeriesData
from kats.models.prophet import ProphetModel, ProphetParams

df = pd.read_csv("data/air_passengers.csv")
df.columns = ["time", "value"]

ts = TimeSeriesData(df)
params = ProphetParams(seasonality_mode="multiplicative")
m = ProphetModel(data=ts, params=params)
m.fit()
fcst = m.predict(steps=30, freq="MS")
m.plot()
plt.show()
```

Şekil 10.5: Prophet model örneği.

çizdirilerek program sonlandırılmıştır. Şekil 10.6 üzerinde plot metodunun çıktısı görülmektedir. Tahminlenen 30 aylık veri mevcut verinin devamında farklı renkle çizdirilmiş bulunmaktadır.



Şekil 10.6: Prophet modeli ile tahminleme.

Theta modeli ile de tahminleme yapılmaktadır. Theta metodu (Assimakopoulos 2000) tek değişkenli bir tahmin metodudur. (Hyndman vd, 2003)'de bu metodunun basit üstel yumuşatma

metodu ile benzer bir başarım sağladığını gösterilmektedir.

Kats içerisinde yer alan ThetaModel, R dili için açık kaynaklı olarak yayınlanmış bulunan thetaf fonksiyonu ile benzer şekilde gerçeklenmiştir.

ThetaModel de gerekli parametreler ayarlandıktan sonra Scikit-Learn kütüphanesinin kullanım tarzına benzer şekilde kullanılmaktadır. Theta modeli ile yapılmış bir tahminleme işleminin örnek kodları Şekil 10.7 üzerinde görülmektedir. Yıllık sezonsallığı olan aylık verilerden oluşan bir veriseti ile çalışıldığı için m değeri 12 olarak seçilmiştir. Model nesnesi oluşturulurken frekans değeri verilmemiş, zaman endeksinden çıkarması istenmiştir. Güven aralığının anlamlılık değerini ifade eden alpha parametresi 0.2 olarak seçilmiştir. Varsayılan değer 0.05 olarak verilmiştir.

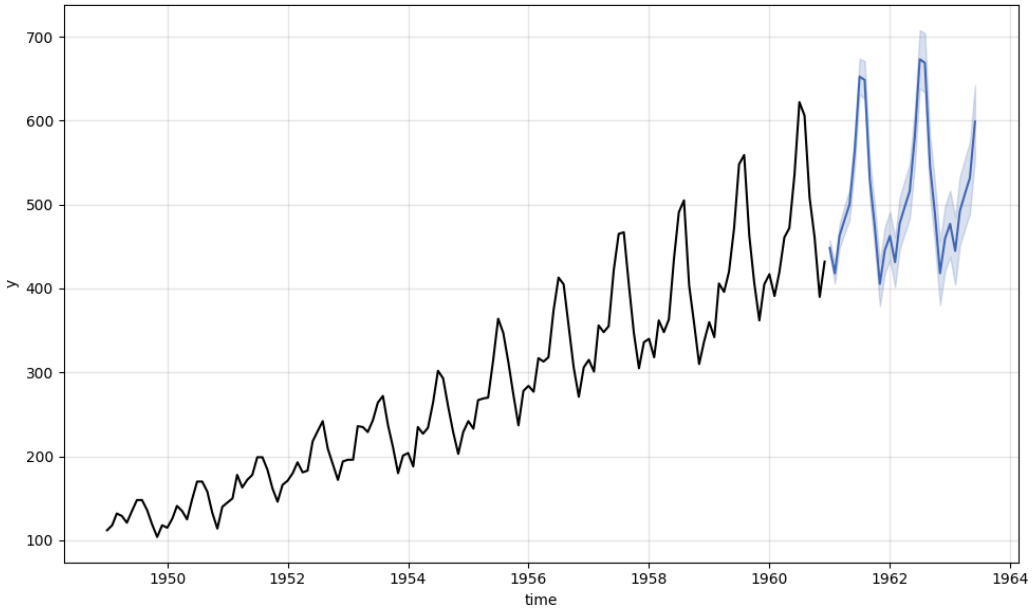
```
1  import sys
2  import pandas as pd
3  import numpy as np
4  import matplotlib.pyplot as plt
5
6  from kats.consts import TimeSeriesData
7  from kats.models.theta import ThetaModel, ThetaParams
8
9  df = pd.read_csv("data/air_passengers.csv")
10 df.columns = ["time", "value"]
11
12 ts = TimeSeriesData(df)
13 params = ThetaParams(m=12)
14 m = ThetaModel(data=ts, params=params)
15 m.fit()
16 fcst = m.predict(steps=30, alpha=0.2)
17 m.plot()
18 plt.show()
```

Şekil 10.7: Theta model örneği.

ThetaModel nesnesinin plot metodu yardımıyla sonuç çizdirilerek program sonlandırılmıştır. Şekil 10.8 üzerinde plot metodunun çıktısı görülmektedir. Tahminlenen 30 aylık veri mevcut verinin devamında farklı renkle çizdirilmiş bulunmaktadır.

10.4 Kats ile Tespit Yapmak

Uç değer tespiti, değişim noktası tespiti ve trend değişimlerinin tespiti için de Kats içerisinde modeller ve algoritmalar bulunmaktadır. Uç değerler zaman serisi içerisindeki anormal değerlerdir ve bunların tespiti için Kats içerisinde OutlierDetector algoritması yer almaktadır. Zaman serisi içerisinde belirli bir noktadan itibaren istatistiksel özellikler değişim göstermeye başladıysa bu noktalara değişim noktası denilmektedir ve bunların tespiti için Kats içerisinde yaygın kullanılan CUSUM (Cumulative Sum Control Chart) Detection (Chang, 1995), Bayesian Online Change Point Detection (BOCPD) (Agudelo-Espana vd, 2019) ve Statistical Significance Detection (Xie, 2020) algoritmaları yer almaktadır. Yavaş trend değişimlerinin tespiti için ise Mann-Kendall tespit algoritması bulunmaktadır.



Şekil 10.8: Theta modeli ile tahminleme.

10.4.1 Kats ile Değişim Noktası Tespiti

Kats içerisinde değişim noktası tespiti için CUSUM Detection algoritması CUSUMDetector nesnesi ile yapılmaktadır. CUSUM bir zaman serisi içerisinde ortlamanın küçük artırımlı değişimlerini tespit etmek için geliştirilmiş olan ve yaygın kullanılan bir algoritmadır.

CUSUM önce bir değişim noktası tespit eder. Belirli bir iterasyon boyunca yeni bir değişim noktası bulunana kadar ya da iterasyon sayısı tamamlanana kadar devam eder. Bulunan değişim noktasının istatistiksel olarak anlamlı olup olmadığı test edilir. CUSUMDetector içerisinde artış ve azalış yönünde ya da her iki yönde de tesipt yapması için change_directions parametresine sahiptir. İstatistiksel olarak anlamlılık seviyesi için eşik değeri belirlenmek için threshold parametresi bulunmaktadır. CUSUM yönteminin diğer ihtiyaç duyulan parametreleri de yine CUSUMDetector içerisinde tanımlanmıştır.

Şekil 10.9 üzerinde CUSUM algoritması ile değişim tespiti yapan kod örneği yer almaktadır. numpy kütüphanesi yardımıyla 30 adet ortalaması 1 olan, 30 adet de ortalaması 2 olan normal dağılımlı nokta oluşturulup birleştirilerek bir seri oluşturulmuştur. Bir CUSUMDetector nesnesi oluşturulmuştur ve bütün tespit nesnelerinde olduğu gibi CUSUMDetector de detector isimli bir metot yardımıyla tespit işlemini yapmaktadır.

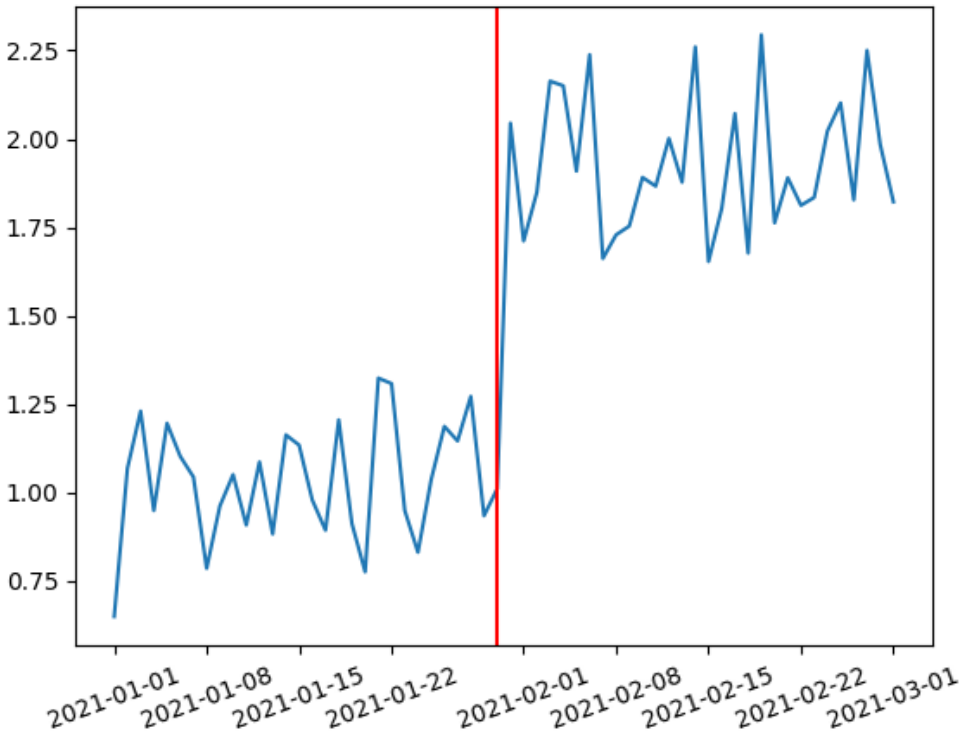
Tespit edilen nokta ya da noktalar yine CUSUMDetector nesnesinin plot metodu yardımıyla çizdirilmiştir. İlk 30 noktadan sonra oluşan değişim noktası da Şekil 10.10 üzerinde görülmektedir.

CUSUMDetector nesnesinin detector metodundan List[Tuple[TimeSeriesChangePoint, CUSUMMetadata]] tipinde bir dönüş olmaktadır. Tutarlı ve kolay öğrenilir bir yapı olması açısından Kats içerisindeki bütün detector metotları benzer bir dönüş tipine sahiptirler. CUSUMMetadata tipi sayesinde de değişim noktası hakkında bir çok veriye ulaşmak mümkün olmaktadır. Şekil 10.11 üzerinde örnek uygulama için hesaplanan ve yazdırılan meta veri değerleri görülmektedir.

Bayesian Online Change Point Detection (BOCPD) algoritması da bir değişim noktası tespit algoritmasıdır. Bu algoritma online çalışmaktadır, bu yüzden zaman serisinin bütün değerlerine

```python
import numpy as np
import pandas as pd
import sys
import matplotlib.pyplot as plt
from pprint import pprint

from kats.consts import TimeSeriesData
from kats.detectors.cusum_detection import CUSUMDetector

np.random.seed(100)
df = pd.DataFrame(
    {
        "time": pd.date_range("2021-01-01", periods=60),
        "value": np.concatenate([np.random.normal(1,0.2,30), np.random.normal(2,0.2,30)])
    }
)
ts = TimeSeriesData(df)

detector = CUSUMDetector(ts)
points = detector.detector()

plt.xticks(rotation=20)
detector.plot(points)
changepoint, metadata = points[0]
pprint(metadata.__dict__)
plt.show()

```

Şekil 10.9: CUSUM algoritması ile değişim noktası tespiti kod örneği.



Şekil 10.10: CUSUM algoritması ile değişim noktası tespiti.

```
{'_cp_index': 29,
 '_delta': 0.897441831093756,
 '_direction': 'increase',
 '_llr': 120.7293148677213,
 '_llr_int': inf,
 '_mu0': 1.03290840117611 78,
 '_mu1': 1.9303502322698738,
 '_p_value': 0.0,
 '_p_value_int': nan,
 '_regression_detected': True,
 '_stable_changepoint': True}
```

Şekil 10.11: CUSUMMetadata değerleri.

ihtiyaç duymamaktadır. detector metodunun lag parametresi ile ne kadar geriye bakması gerektiği belirtilir. Bayesci bir yaklaşıma sahiptir. Bir değişim noktasının olasılığı ve kullanacağı tahmin modeli verilmelidir. Algoritmanın mevcut gerçeklemesi normal dağılım, trend değişim dağılımı ve poison proses modeli adı verilen tahminleme modellerini desteklemektedir.

BOCPD algoritması ile yapılan örnek bir değişim noktası tespit kodu Şekil 10.12 üzerinde görülmektedir. Sentetik zaman serisi üretmek için Kats içerisinde Simulator nesnesi yer almaktadır. Bu örnekte düzey kayması kullanılarak değişim noktaları oluşturulmuştur. Diğer tespit algoritmaları ile BOCPD de benzer metotlara sahiptir.

Şekil 10.13 üzerinde ise üretilen zaman serisi sinyali ve onun üzerinde BOCPD algoritması ile tespit edilen değişim noktaları görülmektedir.

BOCPD algoritması ile elde edilen değişim noktalarına ait meta veriler de BOCPDMetadata nesnesi üzerinden elde edilmektedir. Elde edilen ilk değişim noktasına ait meta veriler Şekil 10.14 üzerinde görülmektedir.

10.4.2 Kats ile Uç Değer Tespiti

Bir veri işleme çalışmasının amacı genellikle veriden tutarlı çıkarımlar yapabilmektir. Veriden tutarlı çıkarımlar yapmayı zorlaştıran engellerden bir tanesi de veride yer alan uç (outlier) değerlerdir. Veri analizine başlarken öncelikle veri setinde uç değerler varsa tespit etmeli ve gerekli tedbirler alınmalıdır. Diğer gözlemlerden kayda değer derecede uzakta olan gözlemlere uç değer ya da aykırı değer adı verilmektedir (Singh vd, 2012). Uç değerler, veri setinin geri kalanından farklı davranış gösterirler.

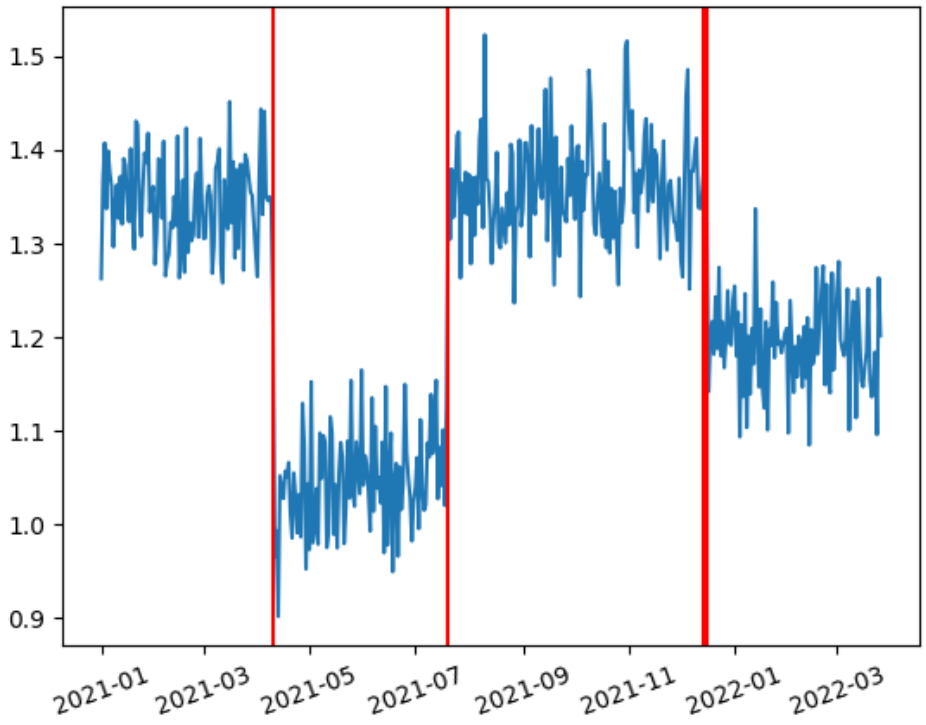
Kats içerisinde tek değişkenli (univariate) zaman serileri için, çok değişkenli zaman serileri için (multivariate) ve düzensiz günlük desenler için tespit algoritmaları yer almaktadır.

OutlierDetector modülü zaman serisi içerisindeki uç değerleri tespit etmektedir. Bu modül öncelikle mevsimsel ayrıştırma işlemini yapar. Bunun için toplamsal ayrıştırma ya da çarpımsal ayrıştırma seçilebilir. Varsayılan olarak toplamsal ayrıştırma seçilmiş durumdadır. Bu değer decomp parametresi yardımıyla değiştirilebilir. Böylece mevsimsellikten ve trendden arındırılmış zaman serisi oluşturulur. Sonra çeyrekler arası açıklık kullanılarak uç değerler tespit edilir. Varsayılan değer 3 olarak belirlenmiştir. Bu değer iqr_mult parametresi yardımıyla değiştirilebilir.

OutlierDetector modülü yardımıyla yapılan örnek uç değer tespit kodu Şekil 10.15 üzerinde görülmektedir. Havayolu uçuş veriseti üzerinde algoritma uygulanmıştır. Verinin farklı iki nok-

```
import numpy as np
import pandas as pd
import sys
import matplotlib.pyplot as plt
from pprint import pprint

from kats.consts import TimeSeriesData
from kats.detectors.bocpd import BOCPDetector
from kats.utils.simulator import Simulator

sim = Simulator(n=450, start="2021-01-01", freq="D")
ts = sim.level_shift_sim(noise=0.05, seasonal_period=1)

detector = BOCPDetector(ts)
points = detector.detector()

plt.xticks(rotation=20)
detector.plot(points)
changepoint, metadata = points[0]
pprint(metadata.__dict__)
plt.show()
```

Şekil 10.12: BOCPD algoritması ile değişim noktası tespiti kod örneği.



Şekil 10.13: BOCPD algoritması ile değişim noktası tespiti.

```
{'_detector_type': <class 'kats.detectors.bocpd.BOCPDetector'>,
 '_model': <BOCPDModelType.NORMAL_KNOWN_MODEL: 1>,
 '_ts_name': 'value'}
```

Şekil 10.14: BOCPDMetadata değerleri.

tasındaki değerler farklılaştırılarak uç değerler oluşturulmuştur. OutlierDetector nesnesi toplamsal ayrıştırma yöntemi ile oluşturulmuştur ve çeyrekler arası açıklık değeri de varsayılan olarak bırakılmıştır.

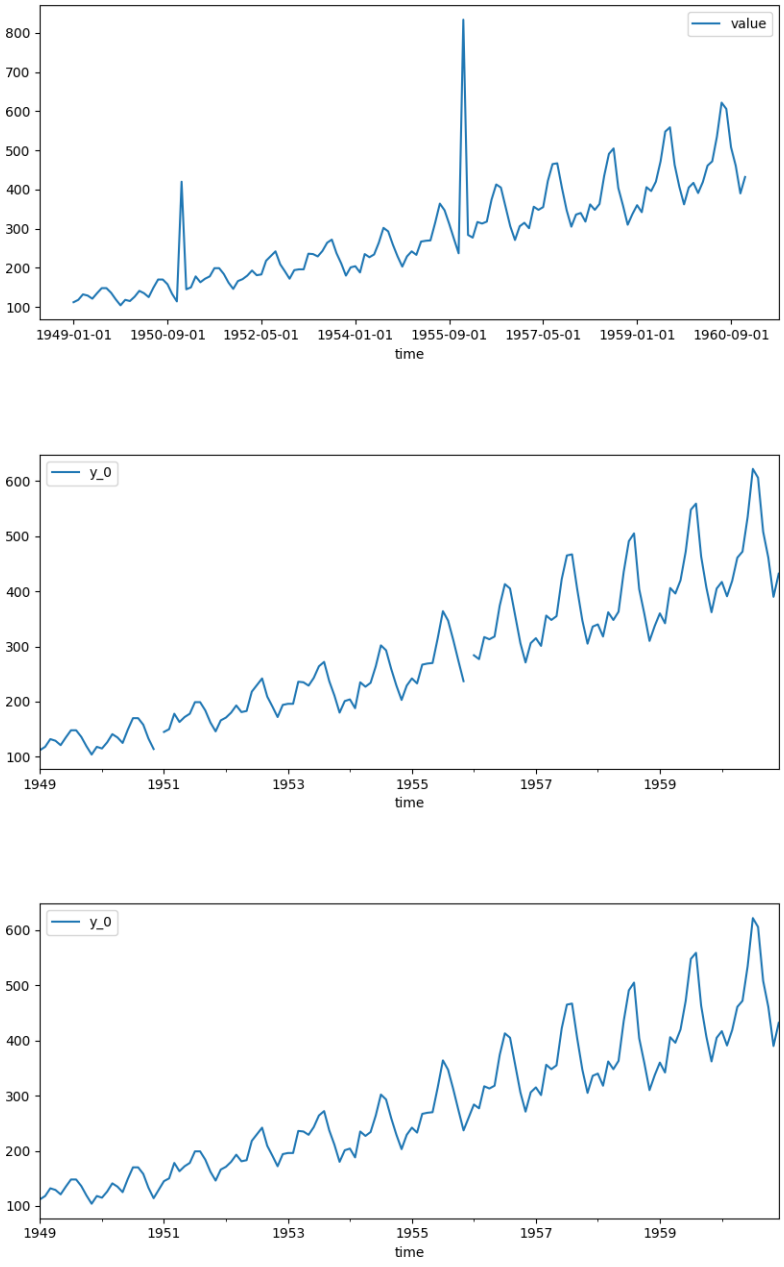
```
import sys
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from pprint import pprint

from kats.consts import TimeSeriesData
from kats.detectors.outlier import OutlierDetector

df = pd.read_csv("./data/air_passengers.csv")
df.columns = ["time", "value"]

df.loc[df.time == "1950-12-01", "value"] *= 3
df.loc[df.time == "1955-12-01", "value"] *= 3

ts = TimeSeriesData(df)
ts_detector = OutlierDetector(ts, "additive")
ts_detector.detector()
pprint(ts_detector.outliers)
ts_no_int = ts_detector.remover(interpolate=False)
ts_int = ts_detector.remover(interpolate=True)

fig, ax = plt.subplots(figsize=(10,15), nrows=3, ncols=1)
df.plot(x="time", y="value", ax = ax[0])
ts_no_int.to_dataframe().plot(x="time", y="y_0", ax=ax[1])
ts_int.to_dataframe().plot(x="time", y="y_0", ax=ax[2])
fig.tight_layout(pad=7.0)
plt.show()
```

Şekil 10.15: OutlierDetector modülü ile uç değer tespiti kod örneği.

Şekil 10.16 üzerinde sırasıyla uç değer içeren zaman serisi, uç değerlerin çıkarılması ile elde edilen zaman serisi ve bu çıkarılan uç değerlerin yerine ara değerler ataması (interpolation) yapılmış zaman serisi alt alta görülmektedir. Varsayılan olarak doğrusal ara değer atama yapılmıştır. Zaman serisinin durumuna göre linear, time, pad, nearest, slinear gibi değerler de seçilebilmektedir.

10.4.3 Kats ile Trend Tespiti

Zaman serilerinin uzun dönemli eğilimini gösteren düzenli harekete trend adı verilmektedir. Trendler serilerin ortalama düzeyini gösteren değerlerdir ve çeşitli şekillerde olabilmektedir. Herhangi bir trendin yönü ya aşağı eğimli ay da yukarı eğimlidir. Bununla birlikte dik veya daha yumuşak, üstel veya yaklaşık olarak doğrusal biçimde de değerlendirilmektedir.



Şekil 10.16: Uç değerler içeren, uç değerler çıkarılmış ve ara değerler eklenmiş zaman serileri.

Trendin belirlenebilmesi için serinin uzun bir zaman diliminde gözlenmiş değerlerine ihtiyaç vardır. Trendin varlığının test edilmesi için parametrik ve parametrik olmayan testler yapılmaktadır. Parametrik testler ilgili anakütle parametresi hakkında herhangi bir varsayımı belirli bir anlam düzeyinde test eder. Parametrik olmayan testler ise genellikle anakütle dağılımı bilinmediği zaman parametrik testlerin uygulanamadığı örneklemlerde tercih edilmektedir. Parametrik olmayan testler genellikle zaman serisi ile zaman arasındaki korelasyon katsayısına dayanmaktadır.

Mann-Kendall (MK) testi de parametrik olmayan bir testtir. Kats içerisinde yer alan MKDetector algoritması MK testi temel alınarak oluşturulmuştur. Kendall'ın Tau katsayılarından faydalanılmaktadır. Trend yoğunluk değeri ve anlamlılık oranı parametre olarak verilir. Trend belirli bir pencere içerisinde aranır. Hangi yönde bir trend varlığı test edilecekse belirtilir. Aşağı, yukarı ya da her iki yöne doğru da trend varlığı test edilebilmektedir. Mevsimsellik tipi de haftalık, aylık, yıllık gibi belirtilebilmektedir (Yue, 2004).

MKDetector modülü yardımıyla yapılan örnek trend tespit kodu Şekil 10.17 üzerinde görülmektedir. Bir Simulator nesnesi yardımıyla sentetik zaman serisi verisi oluşturulmuştur. Haftalık mevsimsellik içeren bir data üretilmiştir.

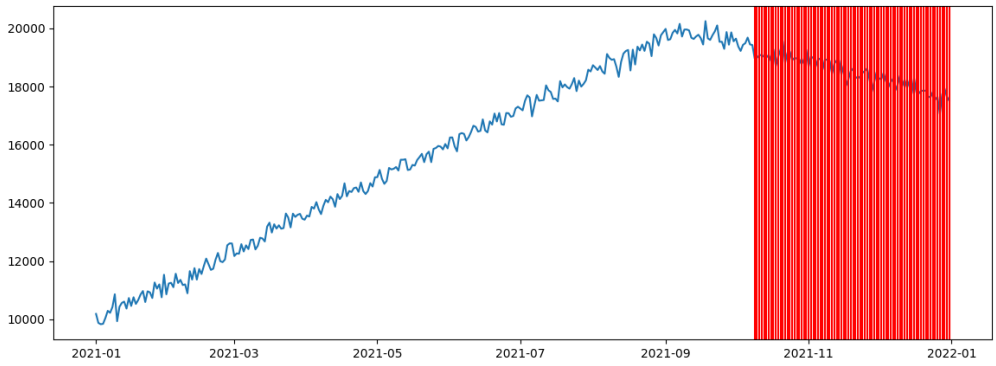
```
from kats import detectors
import numpy as np
import pandas as pd
import sys
import matplotlib.pyplot as plt
from pprint import pprint

from kats.consts import TimeSeriesData
from kats.detectors.trend_mk import MKDetector
from kats.utils.simulator import Simulator

sim = Simulator(n=365, start="2021-01-01", freq="D")
ts = sim.trend_shift_sim(
    noise=200,
    seasonal_period=7,
    seasonal_magnitude=0.007,
    cp_arr=[250],
    intercept = 10000,
    trend_arr=[40,-20]
    )

mk_detector = MKDetector(ts, threshold=.8)

points = mk_detector.detector(direction="down", window_size=30, freq="weekly")
mk_detector.plot(points)
plt.show()

changepoint, metadata = points[0]
pprint(metadata.__dict__)
```

Şekil 10.17: Trend tespiti kod örneği.

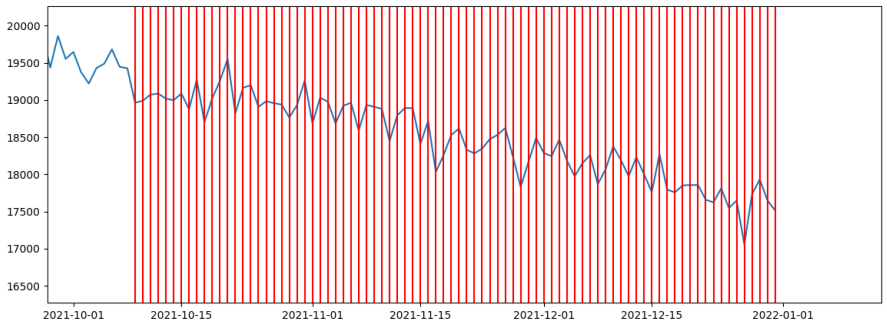
Şekil 10.18 üzerinde, yapılan aşağı yönlü trend tespiti sonucu görülmektedir. Aşağı yönde, belirlenen pencere aralığında oluşan trendlerin başlangıç ve bitiş noktaları değişim noktaları olarak işaretlenmiştir.

Şekil 10.19 ise trend grafiğinin sağ tarafının yakınlaştırılmış halini göstermektedir. Şekil üzerinde oluşan trendlerin başlangıç ve bitiş noktalarının yakalandığı görülmektedir.

Tespit edilen değişim noktaları hakkında da meta veriler oluşmaktadır. Şekil 10.20 oluşan noktalardan ilkinin meta verilerini göstermektedir. Şekil üzerinde trendin yönü, Tau değeri, değişken tipi ve detector tipi görülmektedir.



Şekil 10.18: Trendin olduğu yönde işaretlenmiş denetim noktaları.



Şekil 10.19: Trendin olduğu yönde işaretlenmiş denetim noktalarının bulunduğu bölge.

```
{'_Tau': -0.8344827586206897,
 '_detector_type': <class 'kats.detectors.trend_mk.MKDetector'>,
 '_is_multivariate': False,
 '_trend_direction': 'decreasing'}
```

Şekil 10.20: Değişim noktası meta verileri.

10.5 Kats ile Zaman Serisinden Özellik Çıkarımı

Zaman serilerinin çeşitli istatistiksel özelliklerinin çıkarılmasına ihtiyaç duyulabilmektedir. Özellikle makine öğrenmesi algoritmaları gibi yöntemlerde kullanılmak üzere bu özellikler çıkarılmaktadır. Ayrıca serinin çeşitli istatistiksel yöntemlerle değerlendirilebilmesi için de özellik çıkarımı yapılmaktadır.

Kat içerisinde yer alan TsFeatures modülü bu amaçla geliştirilmiş çeşitli metolar sunmaktadır.

Şekil 10.21 üzerinde otel veriseti yüklenmiş ve TsFeatures modülü yardımıyla özellik çıkarımı yapılmıştır.

```
1  from kats import detectors
2  import numpy as np
3  import pandas as pd
4  import sys
5  import matplotlib.pyplot as plt
6  from pprint import pprint
7  from kats.consts import TimeSeriesData
8
9  from kats.tsfeatures.tsfeatures import TsFeatures
10
11 df = pd.read_csv("./data/air_passengers.csv")
12 df.columns = ["time", "value"]
13
14 ts = TimeSeriesData(df)
15
16 model = TsFeatures()
17
18 output = model.transform(ts)
19
20 pprint(output)
```

Şekil 10.21: TsFeatures kod örneği.

Çıkarılan özellikler Şekil 10.22 üzerinde gösterilmektedir. Şekilde de görüldüğü gibi, istatistiksel özelliklerin yanında çeşitli tahmin ve tespit yöntemleri için ihtiyaç duyulacak parametreler de hesaplanmıştır.

10.6 Tartışma ve Sonuçlar

Kitabın bu bölümünde Facebook şirketi tarafından 1 Haziran 2021́de açık kaynak hale getirilerek yayınlanan Kats kütüphanesinin modülleri anlatılmıştır. Kütüphanede yer alan temel modüller ve metotları örnekler verilerek açıklanmıştır. Pek çok zaman serisi analizi ve tahmini kütüpahanesi olmakla birlikte, Kats zaman serisi verisinin modellenmesi ve analizi ile ilgili bütün teknikleri bir arada toplayan kapsamlı bir kütüphanedir. Ayrıca yaygın kullanılan Scikit-Learn gibi kütüphanelerin kullanım şekli benimsenerek bu ve benzeri kütüphanelerde deneyimi olan kullanıcıların da kolay

```
{'binarize_mean': 0.4444444444444444,
 'crossing_points': 7,
 'diff1y_acf1': 0.30285525815216935,
 'diff1y_acf5': 0.2594591065999471,
 'diff1y_pacf5': 0.2194123478008142,
 'diff2y_acf1': -0.19100586757092733,
 'diff2y_acf5': 0.13420736423784568,
 'diff2y_pacf5': 0.2610103428699484,
 'entropy': 0.4287365561752448,
 'firstmin_ac': 8,
 'firstzero_ac': 52,
 'flat_spots': 2,
 'heterogeneity': 126.06450625819339,
 'histogram_mode': 155.8,
 'holt_alpha': 0.9950208833037283,
 'holt_beta': 0.0042554614726079635,
 'hurst': -0.08023291030513457,
 'hw_alpha': 0.9999999850988388,
 'hw_beta': 4.392391307552844e-15,
 'hw_gamma': 1.3205947301007158e-08,
 'length': 144,
 'level_shift_idx': 118,
 'level_shift_size': 15.599999999999966,
 'linearity': 0.853638165603188,
 'lumpiness': 3041164.5629058965,
 'mean': 280.2986111111111,
 'peak': 6,
 'seas_acf1': 0.6629043863684492,
 'seas_pacf1': 0.15616955255589093,
 'seasonality_strength': 0.3299338017939569,
 'spikiness': 111.69732482853489,
 'stability': 12303.627266589507,
 'std1st_der': 27.206287853461966,
 'trend_strength': 0.9383301875692747,
 'trough': 3,
 'unitroot_kpss': 0.12847508180149078,
 'var': 14291.97333140432,
 'y_acf1': 0.9480473407524915,
 'y_acf5': 3.392072131604336,
 'y_pacf5': 1.003288249401529}
```


Şekil 10.22: TsFeatures ile oluşturulan örnek çıktı.

uyum sağlaması hedeflenmiştir. Kats kütüphanesi sürekli yeni modüllerle ve topluluk katkısı ile büyümeye devam etmektedir. Gelecekte bir çok modülün hızlıca denenip sonuçlarını karşılaştırmalı olarak almak için de olanaklar olacaktır. Bununla birlikte henüz paralel çalışma gibi özelliklerden yoksundur. Çok büyük verilerle çalışmak için paralelleştirme gibi özelliklerin dışarıdan dahil edilmesi gerekmektedir. Bulut ortamında çalışmak ve konteyner teknolojileri ile ölçeklemek mümkün olduğundan komut satırında çalışıp sonuç üreten bir araç olarak da öne çıkmaktadır. Araştırma projelerinde ve üniversitelerde de ticari rakiplerine göre benzer sonuçları üreten ve hatta daha fazla özellik sunan açık kaynaklı araçların tercih edilmesi hem topluluğun üreteceği kollektif tecrübeyi kullanmak hem de kaynak tasarrufu sağlamaktadır.

10.7 Kaynaklar

Agudelo-Espana, D., Gomez-Gonzalez, S., Bauer, S., Scholkopf, B., & Peters, J. (2019). Bayesian Online Detection and Prediction of Change Points. arXiv preprint arXiv:1902.04524.

Assimakopoulos, V., & Nikolopoulos, K. (2000). The theta model: a decomposition approach to forecasting. International journal of forecasting, 16(4), 521-530.

Chang, T. C., & Gan, F. F. (1995). A cumulative sum control chart for monitoring process variance. Journal of Quality Technology, 27(2), 109-119.

Hyndman, R. J., & Billah, B. (2003). Unmasking the Theta method. International Journal of Forecasting, 19(2), 287-290.

Kramer, O. (2016). Scikit-learn. In Machine learning for evolution strategies (pp. 45-53). Springer, Cham. (Kramer, 2016)

Lütkepohl, H. (2005). New introduction to multiple time series analysis. Springer Science & Business Media.

Shumway, R. H., Stoffer, D. S., & Stoffer, D. S. (2000). Time series analysis and its applications (Vol. 3). New York: springer.

Singh, K., & Upadhyaya, S. (2012). Outlier detection: applications and techniques. International Journal of Computer Science Issues (IJCSI), 9(1), 307.

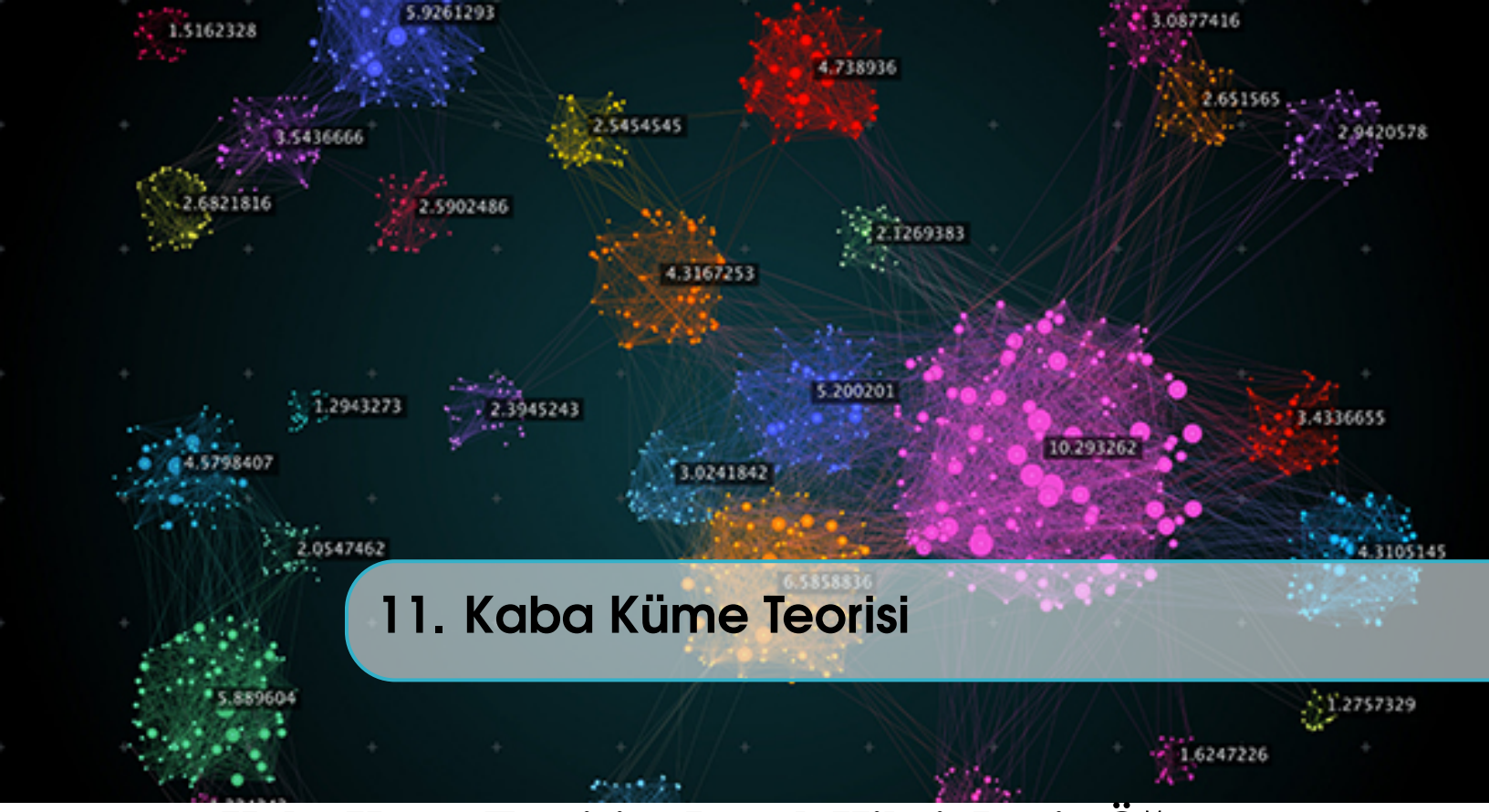
Toharudin, T., Pontoh, R. S., Caraka, R. E., Zahroh, S., Lee, Y., & Chen, R. C. (2020). Employing long short-term memory and Facebook prophet model in air temperature forecasting. Communications in Statistics-Simulation and Computation, 1-24.

Vishwas, B. V., & Patel, A. (2020). Hands-on Time Series Analysis with Python. Apress.

Xie, Y., Zhou, X., & Shekhar, S. (2020). Discovering interesting subpaths with statistical significance from spatiotemporal datasets. ACM Transactions on Intelligent Systems and Technology (TIST), 11(1), 1-24.

Yue, S., & Pilon, P. (2004). A comparison of the power of the t test, Mann-Kendall and bootstrap tests for trend detection/Une comparaison de la puissance des tests t de Student, de Mann-Kendall et du bootstrap pour la détection de tendance. Hydrological Sciences Journal, 49(1), 21-37.

Zhang, P., & Jiang, X., & Holt, G., & Laptev, N., & Komurlu, C., & Gao, P., & Yu, Y. (2021). Self-supervised learning for fast and scalable time series hyper-parameter tuning. arXiv preprint arXiv:2102.05740.



11. Kaba Küme Teorisi

Kaba Küme Teorisine Dayalı Hibrit Derin Öğrenme Algoritmasının Geliştirilmesi ve Uygulanması

Safiye TURGAY*, Orhan TORKUL*

***Sakarya Üniversitesi, Endüstri Mühendisliği**

11.1 Giriş

Yapay sinir ağları, bir öğrenme modelindeki hesaplama birimlerini insan nöronlarına benzer bir şekilde işleyerek, makine öğrenimi görevleri için insan sinir sisteminin benzetimini yapmak için geliştirilmiştir. Özellikle derin öğrenme algoritmaları ile insan düşünme yapısı ve karar modeline en yakın hesaplama yapısına yakın algoritmaların geliştirilmesi hedeflenmiştir. Bu bağlamda analitik hesaplama gücünün yanında daha karmaşık yapıdaki karar mekanizmasının geliştirilmesinde derin öğrenme algoritmaları ile görsel nesnelerin algılanması, işlenmesi ve karar sürecinde kullanılması bu çalışmaları bir adım daha ileriye taşımıştır. Yapay sinir ağlarının en büyük hedefi, insan sinir sistemindeki hesaplamaları temel alan yapıları inşa eden yapay zeka tekniklerini geliştirmektir. Yapay sinir ağları, teorik olarak yeterli eğitim verisi ile herhangi bir matematiksel fonksiyonu öğrenme yeteneğine sahiptir ve tekrarlayan sinir ağları gibi bazı varyantların eksiksiz olduğu bilinmektedir. Kaba kümeleme yaklaşımı ile öğrenmeden bilgi edinme, bilgiyi akılda tutma öncülüğünde, bilgi azaltma ve muhakeme yoluyla değişkenler arasındaki ilişkiyi değerlendirme gibi avantajlara sahiptir. Sınıflandırmayı belirli bir uzaydaki eşdeğerlik ilişkisi olarak anlayan sınıflandırma mekanizmasına dayanmaktadır ve eşdeğerlik ilişkisi, uzayın bölünmesini oluşturmaktadır. Önerilen algoritmada, aşamalar benzer bir ağ yapısını paylaşır. İlk aşama hariç, aşamaların geri kalanı, önceki aşamanın ara tahminini ve ince bilginin azaltılmış sıralı modelini girdi olarak (ağın eğitilebilir kısmına) uyarlayacaktır. İlk aşamanın girdisi, indirgenmiş modeli temsil etmektedir. Her aşamanın öğrenme hedefi aynı indirgenmiş modeli (kaba ölçekli çözüm) göstermektedir. Bununla birlikte, ilk aşamada kaba bir model eğitmek, tahmine ve ek bilgilere dayanarak düzeltme yapmaktır. İkinci ve sonraki aşamalar, birinci aşama ile benzer yapıyı paylaşacak, ancak önceki aşamanın tahminini ve mevcut

aşamayı birleştirecek olan ek birleşim modülüne sahip olacaktır.

Önerilen algoritmada, ilk aşamada kaba bir model eğitmek, tahmine ve ek bilgilere dayanarak düzeltme yapmaktır. İkinci ve sonraki aşamalar, birinci aşama ile benzer yapıyı paylaşacak, ancak önceki aşamanın tahminini ve mevcut aşamayı birleştirecek ek kombinasyon modülüne sahip olacaktır. Kısaca aşamalar benzer bir ağ yapısını paylaşır ve ilk aşamanın girdisi, indirgenmiş bir modeldir ve her aşamanın öğrenme hedefi aynı indirgenmiş modeldir (kaba ölçekli çözüm).

Sonuç olarak, kaba küme teorisi ve derin öğrenme algoritmalarının birleşimi analiz edilir. Kaba kümelere ve farklı sinir ağlarına dayalı birkaç tahmin modelinin doğruluk karşılaştırması yapılır.

11.2 İlgili Çalışmalar

Bir yapay sinir ağı, giriş nöronlarından hesaplanan değerleri çıkış nöronlarına yayarak ve ağırlıkları ara parametreler olarak kullanarak girdilerin bir fonksiyonunu hesaplar. Öğrenme işlemi ise nöronları birbirine bağlayan ağırlıkların değiştirilmesiyle gerçekleşir. Biyolojik organizmalarda öğrenme için dış uyaranlara ihtiyaç duyulduğu gibi, yapay sinir ağlarındaki dış uyaran da öğrenilecek fonksiyonun girdi-çıktı çiftlerinin örneklerini içeren eğitim verileriyle sağlanır. Eğitim verileri, çıktı olarak girdi verilerini ve açıklamalı etiketlerini temsillerini içerebilir. Bu eğitim veri çiftleri, çıktı etiketleri hakkında tahminler yapmak için giriş temsillerini kullanarak sinir ağını besler. Eğitim verileri, belirli bir girdi için tahmin edilen çıktının olasılığı eğitim verilerindeki açıklamalı çıktı etiketiyle ne kadar iyi eşleştiğine bağlı olarak sinir ağındaki ağırlıkların doğruluğuna ilişkin geri bildirim sağlar. Benzer şekilde, nöronlar arasındaki ağırlıklar, tahmin hatalarına yanıt olarak bir sinir ağında ayarlanır. Ağırlıkları değiştirmenin amacı, gelecekteki yinelemelerde tahminleri daha doğru hale getirmek için hesaplanan işlevi değiştirmektir. Bu nedenle, bu örnekteki hesaplama hatasını azaltmak için ağırlıklar matematiksel olarak doğrulanmış bir şekilde dikkatlice değiştirilir. Kaba küme tabanlı derin öğrenme algoritmasında ise ağırlıklar daha duyarlı bir biçimde ele alınarak incelenmekte ve öğrenme kalitesinin daha kısa zamanda yakalanması hedeflenmektedir.

Kaba küme tabanlı yapay sinir ağı modelleri, makine öğrenmesi algoritmasını içeren bazı çalışmalar yapılmış fakat kapsamlı olarak kaba küme tabanlı derin öğrenme algoritması geliştirilmemiştir. Bu çalışmada önerilen algoritma ile bu alandaki eksikliğin giderilmesi hedeflenmiştir. Yapılan bazı çalışmalar ise, Skowron ve Dutta kaba kümeleme yapısı ile birlikte yaklaşım uzayı içerisinde strateji aramanın önemine değinmiş ve karmaşık uyarlanabilir sistemler içerisinde akıllı sistemler gibi davranan bir yapı ile birlikte kaba kümelemenin etkin bir biçimde kullanabileceğini vurgulamışlardır. Gerçek sistemlerde kaba kümeleme ve uygulanabilirliğini göstermektedir. Anlık çıkarsama ve öğrenme yapılarına dikkat çekilmiştir. Tan ve arkadaşları(2019), öznitelik alt kümesi yaklaşımına veri madenciliği bakış açısı ile birlikte sezgisel bulanık kaba bir model önermişlerdir. Özellikle alt ve üst yaklaşımların hiyerarşik yapısı ile birlikte EĞER ilişkilerine dayalı bir yapı ile bilgi sistemi kurulumuna sezgisel bulanık kaba kümeleme yaklaşımını önermişlerdir. Durairaj ve Meena, kaba küme teorisi ve yapay sinir ağlarını tıbbi verilerin işlenmesi sürecinde kullanmış ve kaba kümeleme tabanlı bir hibrit model sunmuşlardır. Kaya (2013) göğüs kanserinin teşhisinde, kaba kümeleme ve extreme makine öğrenmesi sürecinde zeki sınıflama yaklaşımını önermiştir. Yıldız ve Karadeniz, göğüs kanseri teşhisinde derin öğrenmeyi kullanmışlardır. Rectified Unit ve Sigmoid Activation Function kullanmışlardır. Aynı zamanda hedef ve tahminler arasındaki yakınlık derecesini bulmak için ise entropi yöntemini kullanmışlardır. Hassan ise kaba kümeleme tabanlı öğrenme algoritmasını gerçek bir problem üzerinde uygulamış ve sınıflandırma işlemini yapmıştır. Lei ve arkadaşları ise kaba kümeleme ve derin öğrenme algoritmalarını birlikte kullanarak bir enerji tahmin modeli geliştirmişlerdir. Özellikle sisteme fazla etki eden faktörleri daha tarafsız hale getirmek ve diğer

faktörlerin de etki derecesini artırmak için kaba kümeleme yaklaşımını kullanmışlardır. Daha sonra nötr hale getirilen faktörler, derin sinir ağları ile analiz edilmiştir. Aynı zamanda derin sinir ağının tahmin sonuçları ile karşılaştırılmıştır. Jahangir ve arkadaşları ise kısa dönem rüzgâr hızını tahmin eden bir yapı için kaba küme tabanlı yapay sinir ağları modelini önermiştir. Bu çalışma ile kaba küme tabanlı derin öğrenme algoritması sunulmuş ve uygulama işlemleri detaylı olarak gösterilmiştir.

11.3 Kaba Kümeleme ve Temel Özellikleri

Bu bölümde kaba kümeleme yapısı ve temel özellikleri hakkında bilgi verilmiş ve daha sonra derin öğrenme ile ilgili temel özelliklere değinilmiştir. Kaba kümeleme özellikle verilerdeki belirsizlik, eksiklik durumlarını dikkate alarak veri nitelik seçiminde etkin bir araç olarak kullanılabilmektedir. Deneyimli bilgi öğrenme yaklaşımı ile bilgiyi akılda tutma özelliği ile birlikte bilgi azaltma ve muhakeme özellikleri ile birlikte değişkenler arasındaki ilişkiyi değerlendirme avantajına sahiptir. Nitelik seçiminde anahtar parametre seçiminde önem durumunun değerlendirilmesi açısından kaba kümeleme yaklaşımı kullanılabilmektedir. Eşdeğer ilişkisi, uzayın bölünmesi özellikleri ile sınıflandırma işlemlerinde rahatlıkla kullanılabilmektedir.

Kaba küme teorisi temel olarak eksik verilerin ve belirsiz bilgilerin öğrenilmesi ve tümevarımı için kullanılır. Bilgi arasındaki potansiyel ilişkiyi araştırmak ve etkileyen önemli faktörleri çıkarmak için çok uygun kılan temel özellikleri bulmakta etkindir.

Bu çalışma tanımlanan $S = \{U, A, K, f\}$ nesneleri sınıflandırmak için bir bilgi ifade sistemi olarak. $U = \{x_1, x_2, x_3, ..., x_n\}$etki alanıdır, burada x_i bir dizi veri örneğini temsil eden bir nesnedir. $A = \{a_1, a_2, a_3, ..., a_n\}$ boş olmayan sonlu bir nitelikler kümesidir ve $a_i\left(x_j\right), a$ niteliğine kaydedilen x değeridir.

A kümesindeki öğeler, giriş parametrelerine ve bina enerji tüketimine atıfta bulunur. $a\left(x\right)$ her bir etki faktörüne karşılık gelen verilerdir. K, A öznitelik kümesinin değer aralığıdır. f, U ve A nın bilgi işlevi kümesidir ve bir eşlemeyi temsil eder. $U \times Ar \rightarrow K$ her nesnenin her özelliğine bir bilgi değeri atayan, yani, $\forall a \in A, \quad x \in U, Fx, \quad a \in K$.

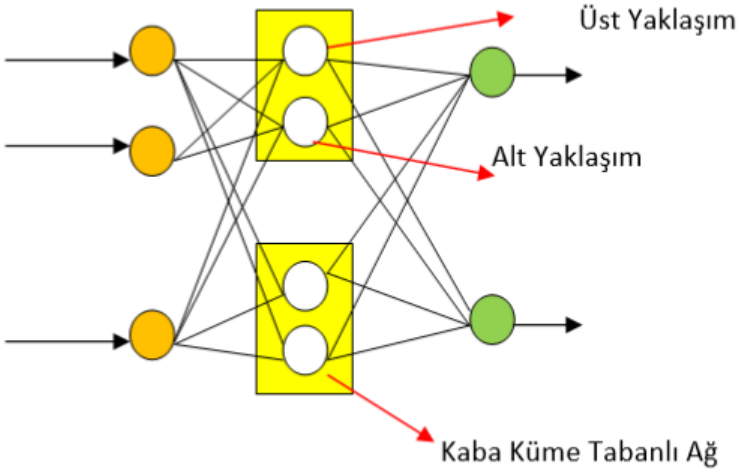
Burada U tüm nesnelerin boş olmayan sonlu bir kümesini temsil eder, $a_i\left(x_j\right), a$A=CUD öznitelik kümesinin boş olmayan sonlu bir özelliğidir C , koşul özniteliği alt kümesi anlamına gelir ve D, karar özniteliği alt kümesi anlamına gelir, $C \cap D = \varnothing$, V bir öznitelik değerleri kümesidir ve F , her nesnenin öznitelik değerini belirlemek için bir bilgi işlevidir, yani $F : U \times A \rightarrow V$. S bilgi sisteminde öznitelik alt kümesi vardır.

Kaba nöron yapısı şekilde gösterilmektedir. Kaba Nöron, üst sınır ve alt sınır olarak adlandırılan bir çift nöron olarak düşünülebilir. Kaba nöronlarda, x bir özellik değişkeni olarak kabul edildiğinde $\underline{x}$ ve semboller sırasıyla alt sınır ve üst sınır değişkeni olarak tanımlanır. R-NN'ler, çevresel gürültünün giderilmesinde ve girdi verilerinin belirsizliğinin ele alınmasında etkili araçlardır. Bu bakımdan bu konudaki en iyi uygulamalar arasındadırlar. Şekilde önerilen R-NN'nin ileri beslemeli denklemleri aşağıdaki gibi tanımlanır.

Amacımız mümkün olan en düşük farkla giriş verilerini çıkış verilerine dönüştüren kaba küme tabanlı derin öğrenme yapısını tasarlamaktır. Geri yayılımlı öğrenme ile girdi verilerinin etkileyen önemli kriter durumlarını belirlemeye çalışıyoruz. Önerilen yapıda ise aşağıdaki bölümler yer almaktadır. Bunlar;

Kayıp fonksiyonu - Reform katmanı - Gizli katman - Giriş katmanı Giriş ve yeniden yapılandırılmış veriler arasındaki genel yapı aşağıdaki şekilde verilmiştir:

ön eğitim tekniğinde girdi verileri 0-1 arasında yer alır.



Şekil 11.1: Kaba küme tabanlı derin öğrenme algoritması

$$H = f\left(w'X + b'\right) \tag{11.1}$$

Gizli katman değeri

$$X' = f\left(w'H + b'\right) \tag{11.2}$$

Bu tekrarlayan işlemler denetimsiz yapıda hedef giriş verilerinin kalitesini yükseltmek üzere yapılmaktadır.

Bu işlem denetimsiz öğrenme uygulamasına girmektedir. Hedef giriş verileri ile işleme devam edilmektedir.

Burada kullanılan alt sınır ve üst sınır nöronları ile birlikte değerlendirme işlemi gerçekleştirilmektedir. Değerlendirme işlemleri esnasında aynı zamanda en küçük kareler regresyonu ve lojistik regresyon gibi geleneksel makine öğrenimi algoritmaları kullanılmıştır. Yapay sinir ağları, tahmin hatasını en aza indirmek için farklı birimlerin ağırlıklarını ortaklaşa öğrenerek daha etkin bir biçimde çalışır.

Birden çok birimi birleştirerek, temel makine öğreniminin temel modellerinde bulunandan daha karmaşık verilerin işlevlerini öğrenmek için modelin gücü artırılır. Derin öğrenmede, öncelikle yeterli verinin bulunması ve geliştirilen algoritma ile birlikte hesaplanması önemlidir. Derin sinir ağlarının üstün performansı, biyolojik sinir ağlarının güçlerinin çoğunu derinlikten de kazandığı gerçeğini yansıtır. Ayrıca biyolojik ağlar, tam olarak anlamadığımız şekillerde birbirine bağlıdır. Biyolojik yapının bir düzeyde anlaşıldığı birkaç durumda, bu hatlar boyunca yapay sinir ağları tasarlanarak önemli ilerlemeler elde edilmiştir.

Eğitim verilerinin veya hesaplama gücünün karmaşıklığı mevcut mimari yapısına eklenen ve çıkarılan nöronlar ile ifade edilebilir. Sinir ağlarının son zamanlardaki başarısının büyük bir kısmı, modern bilgisayarların artan veri kullanılabilirliği ve hesaplama ile geleneksel makine öğrenme algoritmalarının birlikte kullanılabilmesi ile açıklanabilir.

Kayıp fonksiyonu formülü şu şekilde tanımlanır:

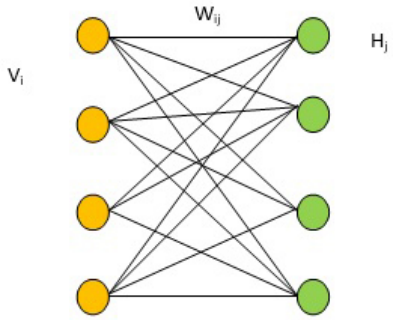
$$L(X,X') = \| X - X' \|^2 \tag{11.3}$$

Gürültü giderme fonksiyonu olarak;

$$E_{RBM}(V,H) = -\sum_{i=1}^{p} b_i V_i - \sum_{j=1}^{q} b_j H_j - \sum_{i=1}^{p}\sum_{i=1}^{q} V_i H_j W_{ij} \tag{11.4}$$

$$z_i = \frac{1}{1+\exp(-W_{LR}H + b_{LR})} \tag{11.5}$$

Bu fonksiyon ile girdi verileri yeniden düzenlenerek analiz işlemi esnasında çıktının doğru tahmin edilmesini sağlayarak öğrenme performansını yükseltmektir. Sistemde bulunan gizli katmanlara ne kadar sağlam yapıda girdi verisi sunarsak sistemin performansı da o denli yüksek olur.



Şekil 11.2: Gürültü giderme fonksiyonu elemanlarının gösterimi

Kayıp fonksiyonu formülü şu şekilde tanımlanır:

$$L(X - X') = \left\| X - X' \right\|^2 \tag{11.6}$$

Gürültü giderme sürecinde, yeniden yapılandırılmış verilerden özellik çıkarımı esnasında gürültü giderme fonksiyonundan yararlanılmaktadır.

$$L(X - X') = -\sum_{j=1}^{n_0} [X_i \log(z_i) + (1 - X_i)\log(1 - z_i)] \tag{11.7}$$

Tüm katmanların eğitilerek öğrenme süreç aşamasında entropi yönteminden yararlanılmaktadır. Ön eğitim işleminde 0.05 ile 0.001 değerleri arasında önceden eğitilmiş ağın performans değeri ile birlikte ağın eğitimi süreci tamamlanmış olmaktadır.

Burada kullanılan alt sınır ve üst sınır nöronları ile birlikte değerlendirme işlemi gerçekleştirilmektedir. Girdi katmanının herhangi bir hesaplama yapmamasına ve yalnızca öznitelik değerlerini iletmesine rağmen, algılayıcının iki katman içermesi dikkat çekicidir. Giriş katmanı, bir sinir ağındaki katmanların sayısına dahil edilmez. Algılayıcı tek bir hesaplama katmanı içerdiğinden, tek

katmanlı bir ağ olarak kabul edilir. Birçok ortamda, tahminin önyargı olarak adlandırılan değişmez bir kısmı vardır. Momentum faktörü 0 ile 1 arasında tanımlanır ve ağırlıklar, önyargılar ve Kaba küme gibi eğitilebilir her parametre nöron parametrelerinden ibarettir.

Böyle bir durumda, yukarıda belirtilen yaklaşım tahmin için yeterli değildir. Tahminin bu değişmez kısmını yakalayan ek bir önyargı değişkeni b eklememiz gerekiyor:

$$f\left(W_U^2 O^1 + b_U^2\right) > f\left(W_A^2 O^1 + b_A^2\right) \tag{11.8}$$

daha sonra, farklı parametrelerin eğitim rotası aşağıdaki gibi tarif edilebilir:

$$O_U^1 = Max\left(f\left(W_U^1 O^1 + b_U^1\right), f\left(W_A^1 O^1 + b_A^1\right)\right) \tag{11.9}$$

$$O_L^1 = Min\left(f\left(W_U^1 O^1 + b_U^1\right), f\left(W_A^1 O^1 + b_A^1\right)\right) \tag{11.10}$$

$$O^1 = \alpha^1 O_U^1 + \beta^1 O_A^1 \tag{11.11}$$

Kaba küme tabanlı eğitim sürecinde 2. dereceden elde edilen çıktı parametreleri ise,

$$O_U^2 = Max\left(f\left(W_U^2 O^1 + b_U^2\right), f\left(W_A^2 O^1 + b_A^2\right)\right) \tag{11.12}$$

$$O_L^2 = Min\left(f\left(W_U^2 O^1 + b_U^2\right), f\left(W_A^2 O^1 + b_A^2\right)\right) \tag{11.13}$$

$$O^2 = \alpha^2 O_U^2 + \beta^2 O_A^2 \tag{11.14}$$

Yerel en iyi sonuçlardan kaçınmak ve bu ağlarda eğitim işleminin yakınsamasını iyileştirmek, başlangıç değerini bulmak için ön işleme yöntemi kullanılmalıdır. Başlangıçta ağırlıklar ve eğitim işlemi düşük öğrenme oranına sahiptirler. Ağırlıkların başlangıç değerleri belirlendiğinde rastgele, eğitimde ağın yakınsamasını kaybedebiliriz, bu durumda uygun olmayan veya uygun olmayan rastgele ağırlıkları seçerek, eğitim hatası ile her eğitim döneminde artacaktır; bu durumda, eğitim işleminin yakınsaması kaybolabilir. Bu olumsuz durumu engellemek için başlangıç ağırlığı bağımsız olarak belirlemek gerekir.

Geri yayılım denklemleri aşağıdaki gibi tanımlanır:

$$\varphi(k) = \varphi(k-1) - \eta \frac{\partial E(k-1)}{\partial \varphi(k-1)} + \gamma \Delta\varphi(k-1) \tag{11.15}$$

Önerilen Algoritma Adımları:

Adım 1: Veriler toplanır ve iki kümeye bölünür (%90 eğitim seti ve %10 test seti).

Adım 2: Kaba küme teorisine dayalı olarak tüm parametrelerin önemi analiz edilir.

(1) Eşit frekans ölçekleyici algoritma kullanılarak veriler ayrıklaştırılır,

(2) Kaba küme teorisini kullanarak, karar niteliğine göre koşul niteliğinin indirgeme bölgesi elde edilir,

(3) Hibrit algoritma ile öznitelik önemi hesaplanır,

(4) Eşleme deseni öznitelik önemine göre oluşturulur.

Adım 3: Önerilen model çalıştırılır.

(1) Kaba bilgilere dayanarak farklı girdi-çıktı yapılarını oluşturulur;

(2) Önerilen algoritma çalıştırılır.

Adım 4: Eğitilmiş modeli kullanarak tahminlerin çıktısı elde edilir.

Sonuç olarak, kaba küme teorisi ve derin öğrenme algoritmalarının kombinasyonu analiz edilir. Kaba kümelere ve farklı sinir ağlarına dayalı birkaç tahmin modelinin doğruluk karşılaştırması yapılır.

$$w_{jk}^{i}(yeni) = w_{jk}^{i}(eski) + \alpha E_{k}^{i+1} x_{jk} \tag{11.16}$$

w_{jk}^{i} -i nci katmanının j nci elemanına olan $(i+1)$ nci katmanın k ncı elemanına ait atanan ağırlık değerini ifade eder.

α – Öğrenme katsayısı

E_{k}^{i+1}- $(i+1)$ nci katmanın k ncı elemanına ait atanan hata katsayısı

$\left(x_{jk}^{i}\right)$- $(i+1)$ nci katman $\left(O_{j}^{i}\right)$ nci katmandan i katmanındaki $j$değerinin girdi olarak ele alınması

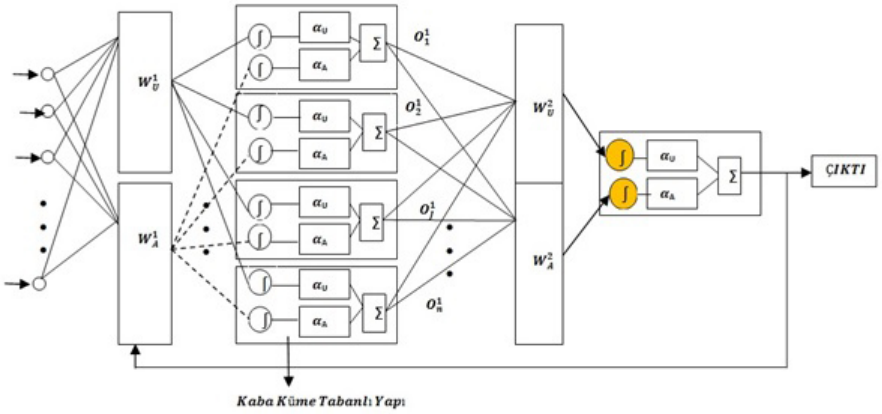
$$E_n = O_n(1-O_n)\sum w_{nj}E_j \tag{11.17}$$

11.3.1 Önerilen Tahmin Modülü

Bu kısımda, tahmin görevi gürültü giderme işleminden sonra başlayacaktır. Gürültüsüz veriler Kaba Küme Tabanlı derin öğrenmenin girdileri olarak kabul edilir. Şekil 3'de gösterildiği gibi, bu bölümde kullanılan ağ iki kaba katmana sahiptir. Giriş verileri Kaba Küme Tabanlı derin öğrenmenin alt ve üst kısmına ait verileri eğitmek için kullanılır. Kaba nöron katsayıları ile her iki katmanın alt sınır ağırlıkları (α ve β) eğitilir. Bunun asıl amacı, gürültü giderme fonksiyonu ile birlikte derin öğrenme sürecini gerçekleştirmektir. Kaba küme yapılı nöronlar ile tahmin kısmı ise sadece iki kaba katmandan oluşur ve derin öğrenme süreci sadece gürültü giderme bölümünde kullanılır.

11.3.2 Son Eğitim Süreci

Kaba Küme Tabanlı Derin Öğrenme yapısının optimizasyonundan sonra, gürültü giderme ve uygulamak için tahmin modülleri birlikte sıralanır ve düzenlenir. Derin Öğrenme yapısına sahip nihai tahmin ağı ile gürültü giderme işlemleri de tamamlanıp, net sonuca ulaşılabilmesi mümkündür. Gürültü giderme modülündeki eğitim parametrelerinin bireysel olarak eğitilir, başlangıç değerleri olarak tanımlanır ve tüm parametreler (sessizleştirme ve tahmin modülleri) bu konuda eş zamanlı olarak eğitilir. Şekil 3 de ise kaba küme tabanlı derin öğrenme mimari yapısının detaylı biçimi yer almaktadır.



Şekil 11.3: Kaba Küme Tabanlı Derin Öğrenme Mimarisi

11.3.3 Hata Hesaplama Kriterleri

Bu çalışmada araştırılan farklı verilerin doğruluğunu karşılaştırmak için YSA'lar da farklı hata kriterleri kullanılmıştır. Ortalama Mutlak Hata (OMH), Ortalama Mutlak Yüzde Hatası (OMYH) ve Ortalama Karesel Hata yöntemi (OKHY) ele alınmıştır. (18-20), sırasıyla OMH, OMYH ve OKHY formüllerini gösterir.

$$OMH = \frac{1}{n_0}\sum_{g=1}^{n_0}\left|\hat{Y}_g - Y_g\right| \tag{11.18}$$

$$OMYH = \frac{1}{n_0}\sum_{g=1}^{n_0}\left(\left|\frac{\hat{Y}_g - Y_g}{\hat{Y}_{mean}}\right|\right) \tag{11.19}$$

$$OKHY = \sqrt{\frac{1}{n_0}\sum_{g=1}^{n_0}\left|\hat{Y}_g - Y_g\right|^2} \tag{11.20}$$

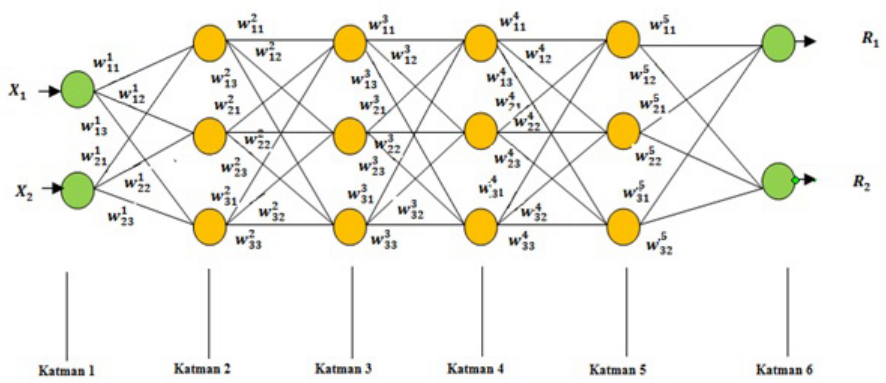
11.4 Uygulama

2 girdiden ve 2 çıktıdan oluşan, her biri 3 katlı, 4 gizli katmanlı bir kaba küme tabanlı hibrit derin öğrenme modelinin çözümüne ait işlem basamakları detaylı olarak verilmiştir (Şekil 4).

$$[2\times 3\times 3\times 3\times 3\times 2]$$

boyutlu ağ yapısında 4 gizli katman bulunmaktadır ve derin öğrenme yapısı ile birlikte sistem analizi çalışması detaylı olarak aşağıda verilmiştir.

İlk veri olan $A = \{a_1, a_2, a_3, ..., a_n\}$, $x_2 = 0,02$ değerleri dikkate alınarak işlemler gerçekleştirilmiştir. Başlangıç ağırlık değerleri Tablo 1'de verilmiştir. Kaba küme tabanlı derin öğrenme algoritmasına ait başlangıç eğitim aşamaları aşağıda verilmiştir.



Şekil 11.4: $[2\times3\times3\times3\times3\times2]$ boyutlu ağ yapısı

$$O = \frac{1}{1+\exp\left(-\sum x_i w_i - t\right)} \tag{11.21}$$

Tablo 11.1: $[2\times3\times3\times3\times3\times2]$ boyutlu ağ yapısında kullanılan ağırlık değerleri

	1. AŞAMA			2. AŞAMA			3. AŞAMA			4. AŞAMA			5. AŞAMA	
Ağırlık	Üst Sınır	Alt Sınır	Ağırlık	Üst Sınır	Alt Sınır	Ağırlık	Üst Sınır	Alt Sınır	Ağırlık	Üst Sınır	Alt Sınır	Ağırlık	Üst Sınır	Alt Sınır
W_{11}^1	0,6	0,4	W_{11}^2	0,2	0,15	W_{11}^3	0,18	0,15	W_{11}^4	0,2	0,18	W511	0,55	0,5
W_{12}^1	0,5	0,3	W_{12}^2	0,65	0,45	W_{12}^3	0,35	0,32	W_{12}^4	0,32	0,3	W512	0,48	0,4
W_{13}^1	0,2	0,1	W_{13}^2	0,35	0,28	W_{13}^3	0,45	0,4	W_{13}^4	0,45	0,4	W521	0,35	0,3
W_{21}^1	0,2	0,1	W_{21}^2	0,24	0,18	W_{21}^3	0,28	0,25	W_{21}^4	0,28	0,25	W522	0,25	0,2
W_{22}^1	0,6	0,5	W_{22}^2	0,48	0,4	W_{22}^3	0,55	0,48	W_{22}^4	0,35	0,3	W531	0,55	0,5
W_{23}^1	0,2	0,1	W_{23}^2	0,35	0,3	W_{23}^3	0,7	0,6	W_{23}^4	0,48	0,45	W532	0,18	0,15
			W_{31}^2	0,28	0,22	W_{31}^3	0,35	0,3	W_{31}^4	0,5	0,45			
			W_{32}^2	0,18	0,14	W_{32}^3	0,15	0,1	W_{32}^4	0,25	0,2			
			W_{33}^2	0,65	0,58	W_{33}^3	0,45	0,4	W_{33}^4	0,18	0,15			

2. katmana ait kaba küme tabanlı üst değere ait çıktı verileri aşağıda detaylı olarak gösterilmiştir.

$$O_1^2 = \frac{1}{1+\exp\{-[(0,05\times0,6)+(0,02\times0,20)-0.0]\}} = 0,508499181$$

$$O_2^2 = \frac{1}{1+\exp\{-[(0,05\times0,5)+(0,02\times0,60)-0.0]\}} = 0,509248945$$

$$O_3^2 = \frac{1}{1+\exp\{-[(0,05\times0,2)+(0,02\times0,20)-0.0]\}} = 0,503499943$$

3. katmana ait kaba küme tabanlı üst değere ait çıktı verileri aşağıda detaylı olarak gösterilmiştir.

$$O_1^3 = \frac{1}{1+\exp\{-[(0,508499181\times0,2)+(0,509248945\times0,20)+(0,503499943\times0,20)-0.0]\}} = 0,590226$$

$$O_2^3 = \frac{1}{1+\exp\{-[(0,508499181\times0,65)+(0,509248945\times0,48)+(0,503499943\times0,18)-0.0]\}} = 0,660516$$

$$O_2^3 = \frac{1}{1+\exp\{-[(0,508499181\times 0,35)+(0,509248945\times 0,35)+(0,503499943\times 0,65)-0.0]\}} = 0,660516$$

4. katmana ait kaba küme tabanlı üst değere ait çıktı verileri aşağıda detaylı olarak gösterilmiştir.

$$O_1^4 = \frac{1}{1+\exp\{-[(0,590226\times 0,18)+(0,660516\times 0,28)+(0,660516\times 0,35)-0.0]\}} = 0,62803$$

$$O_2^4 = \frac{1}{1+\exp\{-[(0,590226\times 0,35)+(0,660516\times 0,35)+(0,660516\times 0,15)-0.0]\}} = 0,66140$$

$$O_3^4 = \frac{1}{1+\exp\{-[(0,590226\times 0,45)+(0,660516\times 0,7)+(0,660516\times 0,45)-0.0]\}} = 0,73633$$

5. katmana ait kaba küme tabanlı üst değere ait çıktı verileri aşağıda detaylı olarak gösterilmiştir.

$$O_1^5 = \frac{1}{1+\exp\{-[(0,66140\times 0,2)+(0,066140\times 0,28)+(0,73633\times 0,5)-0.0]\}} = 0,663507$$

$$O_2^5 = \frac{1}{1+\exp\{-[(0,66140\times 0,32)+(0,066140\times 0,35)+(0,73633\times 0,25)-0.0]\}} = 0,649432$$

$$O_3^5 = \frac{1}{1+\exp\{-[(0,66140\times 0,45)+(0,066140\times 0,48)+(0,73633\times 0,18)-0.0]\}} = 0,675381$$

6. katmana ait kaba küme tabanlı üst değere ait çıktı verileri aşağıda detaylı olarak gösterilmiştir.

$$O_1^6 = \frac{1}{1+\exp\{-[(0,6635070\times 0,55)+(0,649432\times 0,35)+(0,73633\times 0,55)-0.0]\}} = 0,72386$$

$$O_2^6 = \frac{1}{1+\exp\{-[(0,6635070\times 0,48)+(0,649432\times 0,25)+(0,73633\times 0,18)-0.0]\}} = 0,646208$$

Belirlenen hata oranları;

$$R_1 = E_1^6 = O_{1-gerek-değer}^6 - O_1^6 = 1.0 - 0,72386 = 0,27614$$

$$R_2 = E_2^6 = O_{2-gerek-değer}^6 - O_2^6 = 0 - 0,646208 = -0,646208$$

Bu durumda R1 alternatifi seçilerek işleme devam edilir. Aynı işlemler bütün deneme veri seti için öğrenmenin gerçekleştirilmesi için devam edilir. 5. katman hata değerleri

$$E_1^5 = 0,663507\,(1-0,663507)\,[(0,55\times 0,27614)+(0,48\times -0,646208)] = -0,035435$$

$$E_2^5 = 0,649432\,(1-0,649432)\,[(0,35\times 0,27614)+(0,25\times -0,646208)] = -0,0147764$$

$$E_3^5 = 0,675381\,(1-0,675381)\,[(0,55\times 0,27614)+(0,18\times -0,646208)] = 0,00779616$$

4. katman hata değerleri

$$E_1^4 = 0,62803\,(1-0,6283)\,[(0,2\times -0,0353435)+(0,32\times -0,0147764)+(0,45\times 0,00779616)] = -0,00193657$$

$$E_2^4 = 0,0661400\,(1-0,661400)\,[(0,28\times -0,0353435)+(0,32\times -0,0147764)+(0,45\times 0,00779616)] = -0,007388003$$

$$E_3^4 = 0,73633\,(1-0,73633)\,[(0,5\times -0,0353435)+(0,25\times -0,0147764)+(0,18\times 0,00779616)] = -0,005721712$$

3. katman hata değerleri

$E_1^3 = 0,590226\,(1-0,59226)\,[(0,18\times -0,001936357)+(0,5\times -0,0073883)+(0,45\times 0,00779616)] = -0,002087$

$E_2^3 = 0,660516\,(1-0,66516)\,[(0,28\times -0,001936357)+(0,55\times -0,007388003)+(0,7\times 0,00779616)] = -0,000555$

$E_3^3 = 0,664516\,(1-0,664516)\,[(0,35\times -0,001936357)+(0,15\times -0,007388003)+(0,45\times 0,00779616)] = -0,00113$

2. katman hata değerleri

$E_1^2 = 0,508499181\,(1-0,508499181)\,[(0,2\times(-0,002087))+(0,65\times(-0,000555))+(0,35\times(-0,00113))] = -0,0002$

$E_2^2 = (1+0,000555)\,[(0,24\times(-0,002087))+(0,48\times(-0,000555))+(0,35\times(-0,00113))] = -0,0002906$

$E_3^2 = (1+0,00113)\,[(0,28\times(-0,00287))+(0,18\times(-0,000555))+(0,65\times(-0,00113))] = -0,0003547$

5.katman için güncellenmiş yeni ağırlıklar

$w_{11}^5 = 0,55+0,3\times 0,27614\times 0,663507 = 0,604966$

$w_{21}^5 = 0,48+0,3\times 0,27614\times 0,64943207 = 0,403800$

$w_{31}^5 = 0,35+0,3\times 0,27614\times 0,675381 = 0,60595$

$w_{12}^5 = 0,55+0,3\times(-0,64621)\times 0,663507 = 0,351371$

$w_{22}^5 = 0,48+0,3\times(-0,64621)\times 0,649432 = 0,124100$

$w_{32}^5 = 0,35+0,3\times(-0,64621)\times 0,675381 = 0,049069$

4.katman için güncellenmiş yeni ağırlıklar

$w_{11}^4 = 0,2+0,3\times(-0,0353435)\times 0,62803 = 0,193340994$

$w_{21}^4 = 0,32+0,3\times(-0,0353435)\times 0,6614 = 0,272987146$

$w_{31}^4 = 0,45+0,3\times(-0,0353435)\times 0,73633 = 0,492192631$

$w_{22}^4 = 0,35+0,3\times(-0,0147764)\times 0,62803 = 0,347068057$

$w_{32}^4 = 0,48+0,3\times(-0,0147764)\times 0,73633 = 0,246735885$

$w_{13}^4 = 0,5+0,3\times 0,00779616\times 0,62803 = 0,451468862$

$w_{23}^4 = 0,25+0,3\times 0,00779616\times 0,6614 = 0,481546914$

$w_{33}^4 = 0,18+0,3\times 0,00779616\times 0,736334 = 0,18172217$

1 3.katman için güncellenmiş yeni ağırlıklar

$w_{11}^3 = 0,18+0,3\,(-0,001936357)\times 0,590226 = 0,1796571$

$w_{21}^3 = 0,35+0,3\times(-0,001936357)\times 0,660516 = 0,2796163$

$w_{31}^3 = 0,45+0,3\times(-0,001936357)\times 0,66516 = 0,349614$

$$w_{12}^3 = 0,28 + 0,3 \times (-0,007388003) \times 0,590226 = 0,3486918$$

$$w_{22}^3 = 0,55 + 0,3 \times (-0,00738803) \times 0,660516 = 0,548536$$

$$w_{32}^3 = 0,7 + 0,3 \times (-0,007388003) \times 0,664516 = 0,1485272$$

$$w_{13}^3 = 0,35 + 0,3\,(-0,005721712) \times 0,590226 = 0,4489869$$

$$w_{23}^3 = 0,15 + 0,3\,(-0,005721712) \times 0,660516 = 0,6988662$$

$$w_{33}^3 = 0,45 + 0,3 \times (-0,005721712) \times 0,660516 = 0,488593$$

Bu işlemler ile ağırlıklar, hatalar ve girdi verilerinin durumları karşılaştırılarak geçerli olup olmadığının kontrolü yapılabilir.

11.5 Sonuçlar

Bu çalışmanın temel hedefleri arasında yapay sinir ağları ile derin öğrenme arasındaki bağlantının ortaya çıkartılması ve gerçekleştirilen hesaplamaların önerilen kaba küme tabanlı derin öğrenme algoritması ile birlikte gerçekleştirmektir.

Sonuç olarak yukarıdaki işlem basamakları kullanılarak kaba küme tabanlı derin algoritma ağının hesaplama süreci detaylı olarak gösterilmiştir. Bu çalışma ile birlikte bu konu daha kapsamlı bir biçimde ileri çalışmalarda başka makine öğrenmesi algoritmaları ile birlikte kolaylıkla, geniş bir uygulama alanına adapte edilebilmesi mümkündür.

11.6 Kaynaklar

Durairaj, M., Meena, K., A Hybrid Prediction System Using Rough Sets and Artificial Neural Networks, Internatıonal Journal of Innovatıve Technology & Creatıve Engıneerıng (ISSN:2045-8711) , Vol.1 No.7 July 2011.

Hassan Y., Tazaki E., Rough Set and Genetic Programming. In: Inuiguchi M., Hirano S., Tsumoto S. (eds) Rough Set Theory and Granular Computing. Studies in Fuzziness and Soft Computing, Vol 125. Springer, Berlin, Heidelberg. (2003) https://doi.org/10.1007/978-3-540-36473-3_19

Jahangira, H., Golkara, M.A., Alhamelib, F., Mazouzd, A., Ahmadiane,A., Elkamelf, A. Short-term wind speed forecasting framework based on stacked denoising auto-encoders with rough ANN, Sustainable Energy Technologies and Assessments 38 (2020) 100601

Kaya, Y., A new intelligent classifier for breast cancer diagnosis based on a rough set and extreme learning machine: RS+ELM, Turkish Journal of Electrical Engineering & Computer Sciences, (2013) 21: 2079-2091. doi: 10.3906/elk-1203-119.

Lei, L., Chen,W., Wu, B., Chen, C., Liu, W. A building energy consumption prediction model based on rough set theory and deep learning algorithms, Energy & Buildings 240 (2021) 110886,

Öztemel, E., Yapay Sinir Ağları, İstanbul: Papatya Yayıncılık Eğitim, 2012 xxii, 232s. Rehbein, D.A., Maze, S.M. , Havener, J.P. "The application neural networks in the process industry", ISA Transactions, 1992

Ross, T., Fuzzy Logic with Engineering Applications, Wiley & Sons Ltd, 2004, Second Edition, ISBN 0-470-86074-X (Cloth)

Sevi, M. and Aydın, İ. "COVID-19 Detection Using Deep Learning Methods," 2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI) (2020) pp. 1-6, doi: 10.1109/ICDABI51230.2020.9325626.

Skowron, A., Dutta, S. Rough sets: past, present, and future. Nat Comput 17, 855–876 (2018) https://doi.org/10.1007/s11047-018-9700-3

Tan, A., Wu, W., Qian, Y., Liang, J., Chen, J. and Li, J., "Intuitionistic Fuzzy Rough Set-Based Granular Structures and Attribute Subset Selection," in IEEE Transactions on Fuzzy Systems, vol. 27, no. 3, pp. 527-539 (March 2019) doi: 10.1109/TFUZZ.2018.2862870.

Yasli, F., Irem Türkmen, H. İ., Güvensan, M. A., "Longterm Traffic Speed Estimation via Regression using Weekly Day Patterns"(2019) Innovations in Intelligent Systems and Applications Conference (ASYU).

Yıldız, İ., Karadeniz, A.T., Enhancement of Breast Cancer Diagnosis Accuracy with Deep Learning, Avrupa Bilim ve Teknoloji Dergisi,www.ejosat.com: ISSN: 2148-2683, Özel Sayı, S. 452-462, Ekim 2019

12. Sinir Ağları ve Derin Öğrenme

Differences Between Neural Networks and Deep Learning

Safiye TURGAY*, Suat ERDOĞAN*, Orhan TORKUL*
***Sakarya University, Department of Industrial Engineering**

12.1 Introduction

With the development of computer technology, the idea of thinking of the machine as a human was put forward in the 1980s, with the desire of human beings to do all their work together with computers and technology. In the 1990s, there has been an increase in the studies and applications of artificial neural networks. By imitating the structure based on the human brain, the way the neurons work is examined. Artificial neural networks collect information with examples and then make decisions using the information learned from the examples to evaluate examples. Artificial neural networks have an important place in machine learning methodology, thanks to their ability to learn and generalize. There are different definitions of artificial neural networks, some of them are:

"Artificial neural networks (ANNs) are computer systems developed to automatically perform the abilities of the human brain, such as deriving new information, creating and discovering new information through learning, without any help."

"Computer programs follow biological neural networks with parallel and distributed information processing structures, each capable of processing information through neurons, thanks to weighted connections."

It is based on a network structure consisting of interconnected artificial neurons, with each neuron representing an information processing unit. We call each of these neurons a node that can transmit information by interacting with each other. During the information sharing and interaction of the nodes with each other, the nonlinear mathematical function we call the activation function ultimately transforms the input into a result output to be used as a new input processed in the next layers. There are certain weights between nodes. These weights are updated in each learning round

of the model. When the accuracy ratio is high, the weights are not updated. These weights are refreshed each round using certain functions to improve accuracy and increase performance with feedback. The layer between the input and output layers is called the hidden layThe input layer includes categorizing the input data and presenting them to the system with certain weight values according to their properties. The analytical operations necessary to obtain the result value in the output layer are made in the hidden layer. Increasing the number of hidden layers or increasing the nodes in the hidden layers are two different approaches. In some cases, increasing the nodes in the layer can be a more practical way to solve the problem. Artificial intelligence is a branch of science that enables conscious decisions to be made automatically, thanks to mathematical models developed based on human and living mechanisms in nature. Machine learning is a sub-branch of artificial intelligence, which can process data, learn from data and analyze it. Deep learning, as a sub-branch of machine learning, consists of multi-layered large artificial neural networks that can make smart decisions and learn on their own by using big data. Artificial neural networks are based on the human brain, with a limited number of inputs and outputs, consisting of 1 or 3 hidden layers algorithm structures. In this study, the working forms of artificial neural network algorithms are examined in detail, and then the use of artificial neural network algorithms with deep learning is included.

12.2 Literature Survey

Artificial neural network studies started in the 1950s, the geometric and exponential increase in the data accumulated with the internet age, the smartening of mobile applications, the developments in the medical field, and accordingly, the use of standard ANN models for systems to perform functions such as learning, decision making, and reasoning. The inadequacy has brought different perspectives with it.

In 1998, YannLeCun et al., using LeNet architecture and Convolutional Neural Networks (CNN), and in Hinton and Salakhutdinov's "Reinvigoratedresearch in Deep Learning" studies by using Restricted Boltzmann Machine (RBM) popularized the subject of Deep Learning. they have made. The biological neural network is shown in Figure 2 in detail. Artificial neural networks consist of neurons (nerve cells). Neurons can process information. Biological neural networks consist of nerve cells, and there are approximately 1001 nerve cells in the cortex of the human brain. These cells are in connection with the cells between 1000-10000. Neurons connect to form functions. It is estimated that there are 100 billion neurons in our brain. A neuron can make between 50000 and 250000 connections with other neurons, and it is estimated that there are more than 6×10^13 connections in our brain[3]. A nerve cell consists of a cell body, dendrite, and axon.

The main components that make up the nerve cell

Axon (Axon): It is the body where the single wool electrical transmission is provided, where output pulses are produced. Displays the system output.

Dendrites: Collects signals from other cells and expresses system input.

Synapse: It connects the axons of the cells, that is, the output information, with other dendrites.

Myelin Layer (MyelinSheath): It is an insulating material that has a positive effect on the rate of propagation.

Nucleus: Provides periodic regeneration of signals based on output values. The signal carried in the axon is transmitted to the snaps by chemical carriers. The cytoplasm is polarized to -85mV. -40mV(Na+ in): excitation causes (+) current. *90mV(K+ out): suppressing (-) leads to current.

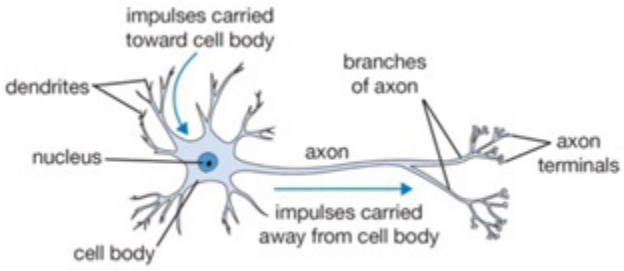


Figure 12.1: Biologic representation of nerve cell(`https://cs231n.github.io/neural-networks-1`)

Above a certain threshold voltage, the cell is excited, while in other cases the cell is suppressed. Output signal generation is called neural computation.

Briefly, the stimuli coming from the nerve cells are carried to the cell body via the dendrites and transmit the chemical process in the destabilization of the intracellular activation to the other cells via axons. When we match the biological system and artificial neural network elements, the neuron corresponds to the processor element, the dendrite collection function to the cell body transfer function, the axons to the artificial neuron output, and the synapses to the weights.

The biological appearance of nerve cells in living things is as we saw above. We have a nucleus and conduction is made along an axon. Here at the output terminals, our sensor data obtained from the dentitions are weighted in the nucleus and transmitted along the axon, and connected to another nerve cell. In this way, communication between nerves is ensured.

12.3 Structure of Artificial Neural Network

Artificial neural networks are structures formed by connecting artificial nerve cells. Artificial neural networks are examined in three main layers; Input Layer, Intermediate (Hidden) Layers, and Output Layer.

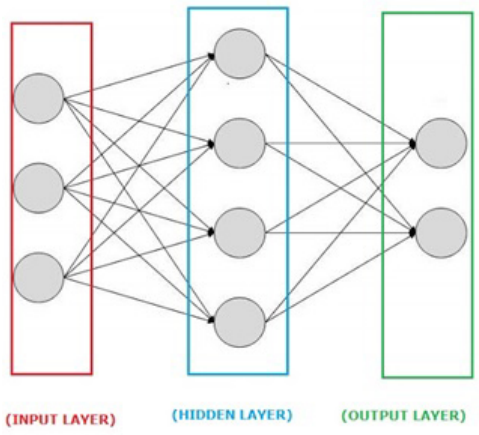


Figure 12.2: Layers of Neural Network

Information is transmitted to the network through the input layer. They are processed in the intermediate layers and sent from there to the output layer. Information processing converts the information coming to the network into output by using the weight values of the network. To network produces the correct outputs for the inputs, the weights must have the correct values.

12.3.1 Artificial Neural Networks Layers

If it consists of many neurons and hidden layers, it is called a multi-layer artificial neural network. If it consists of a single layer, it is called a single-layer artificial neural network. The behavior of ANNs, that is, how they relate input data to output data, is primarily affected by the transfer functions of neurons, how they are connected, and the weights of these connections.

The basic operation in Artificial Neural Networks; is to calculate the w (weight parameter) and b (bias value) parameters that the model will give the best score. Each neuron is calculated in the same way and they are connected in series or parallel. An artificial neuron consists of five parts;

Inputs: Information coming from the outside world to an artificial cell. These are determined by the examples that the network wants to learn.

Weights: This shows the importance of the information coming to an artificial cell and its effect on the cell. The weight wi1 in the figure shows the effect of the xi1 input on the cell. Just because the weights are big or small doesn't mean they're important or unimportant. A weight of zero may be the most important event for that network.

Addition Function: This function calculates the net input to a cell. Different functions are used for this. The most common is the weighted sum. Here, each incoming information is multiplied by its weight. Thus, the net input to the network is found. Some aggregation functions are given in the table below.

Activation Function: This function processes the net input to the cell and determines the output that the cell will produce in response to this input. The activation function is usually chosen as a nonlinear one. The "non-linearity" characteristic of artificial neural networks comes from the nonlinearity of the activation functions. Another point to be considered while choosing the activation function is that the derivative of the function can be easily calculated. Since the derivative of the activation function is also used in feedback networks, a function whose derivative is easy to calculate is chosen so that the calculation does not slow down. Activation functions used in artificial neural networks are listed in Figure 12.3. In the most widely used "Multilayer perceptron" model today, the "Sigmoid function" is generally used as the activation function (for example, ReLU or sigmoid).

Output: It is the output value determined by the activation function. The output produced is sent to the outside world or another cell. The cell can also send its output as input to itself. Although a processing element has more than one output, it can only have one output (Öztemel, 2006).

12.3.2 Effect of Each Layer and Neurons on the Model

Neurons in a layer do not have relationships with each other and carry out the task of transmitting the remaining information in the system to the next layer or output (the task of each neuron). The neurons in two consecutive layers affect each other with various activation values and perform a transfer that determines the learning level of the model

Therefore, the number of neurons in a layer of the model indirectly affects the performance of the system. Although it seems to be said that 'the higher the number of layers, the higher the learning performance', this is not true. Because model performance is only related to the number of inputs and layers, but not determined by it. The influence of many different hyperparameters affects the

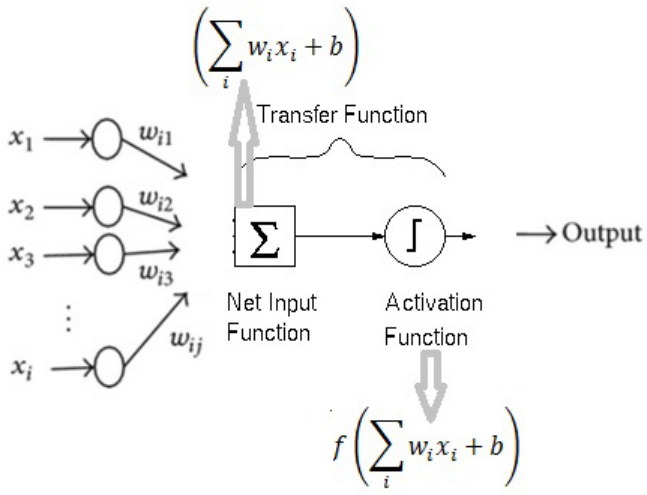


Figure 12.3: Representation of neural structure

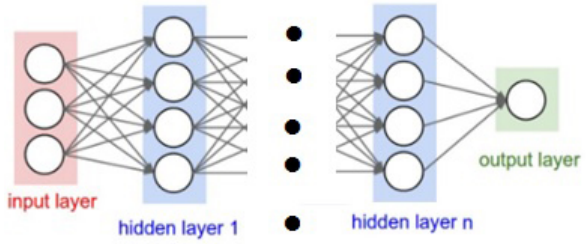


Figure 12.4: n Multi-Layer Neural Network Structure

output performance. In cases where the Model Input (x) is '0', W.x=0 and +b shift the output of the score function. Thus, it allows the model to continue the learning process in the next iteration. This step alone does not affect the learning process. Also, have a look at forwarding and backward propagation. W weight vector number of nodes/neurons (cells), bias (b) values should be equal to the number of nodes in the next layer.

12.4 Deep Learning

Deep learning is a discipline that is based on the structure of artificial neural networks and therefore tries to understand the interaction of neurons with the functioning of the human brain. It allows us to train artificial intelligence to predict outputs with a given dataset. Both supervised and unsupervised learning can be used to train artificial intelligence. We can say that artificial neural networks are the beginning level of deep learning. While all algorithms based on artificial neural networks can be applied in deep learning, the only main difference is the number of hidden layers. In artificial neural networks, the number of hidden layers is limited to a maximum of 3, while in deep learning, the number of hidden layers consists of at least three layers. Today, we can call multilayer artificial neural networks a deep learning approach thanks to the increasing big data and the increasing computing power of computers.

Deep learning is a machine learning method that has developed rapidly in recent years. Deep learning is an artificial neural network. While the number of middle layers in classical artificial neural networks is one or two, in deep learning areas this number can be increased to a minimum of three and this number can be increased according to the person designing the system and the problem. As a big data laboratory, the training started with the "basic neural network" course.

12.5 Application

For the forward computation problem, in 4 input factors, 1 output and 3 hidden layers, examined the change in the weight values of each iteration of the inputs affecting the output factor that is changing in the importance of the input factors.

This example shows how data from an input layer is moved to the output layer. The actions to be taken are valid for only 1 step. Neural network ınput values (I1, I2, I3, and I4) and weight values (A) with nodes are listed below.

$$\mathrm{H1Net1} = \sum_{i=1}^{4} \mathrm{I}_{1} * \mathrm{A}_{1} \rightarrow I_{1} * \mathrm{A}_{1-1} + I_{2} * \mathrm{A}_{2-1} + I_{3} * \mathrm{A}_{3-1} + I_{4} * \mathrm{A}_{4-1}$$

$$\mathrm{H1Net1} = 0,910 * 0,208 + 0,851 * 0,862 + 0,534 * 0,725 + 0,952 * 0,543 = 1,8269$$

$$\mathrm{H1Net2} = \sum_{i=1}^{4} \mathrm{I}_{1} * \mathrm{A}_{1} \rightarrow I_{1} * \mathrm{A}_{1-2} + I_{2} * \mathrm{A}_{2-2} + I_{3} * \mathrm{A}_{3-2} + I_{4} * \mathrm{A}_{4-2}$$

$$\mathrm{H1Net2} = 0,910 * 0,375 + 0,851 * 0,472 + 0,534 * 0,384 + 0,952 * 0,761 = 1,6725$$

$$\mathrm{H1Net3} = \sum_{i=1}^{4} \mathrm{I}_{1} * \mathrm{A}_{1} \rightarrow I_{1} * \mathrm{A}_{1-3} + I_{2} * \mathrm{A}_{2-3} + I_{3} * \mathrm{A}_{3-3} + I_{4} * \mathrm{A}_{4-3}$$

Table 12.1: Input and transition weight values

I1=0,910	I2=0,851	I3=0,534	I4=0,952
A1-1=0,208	A2-1=0,862	A3-1=0,725	A4-1=0,543
A1-2=0,375	A2-2=0,472	A3-2=0,384	A4-2=0,761
A1-3=0,017	A2-3=0,532	A3-3=0,569	A4-3=0,079
A5-1=0,634	A6-1=0,690	A7-1=0,277	A8-1=0,907
A5-2=0,958	A6-2=0,590	A7-2=0,292	A8-2=0,541
A5-3=0,678	A6-3=0,411	A7-3=0,694	A8-3=0,942
A5-4=0,075	A6-4=0,100	A7-4=0,486	
A9-1=0,675	A10-1=0,467	A11-1=0,716	
A9-2=0,727	A10-2=0,208	A11-2=0,702	
A9-3=0,040	A10-3=0,254	A11-3=0,113	

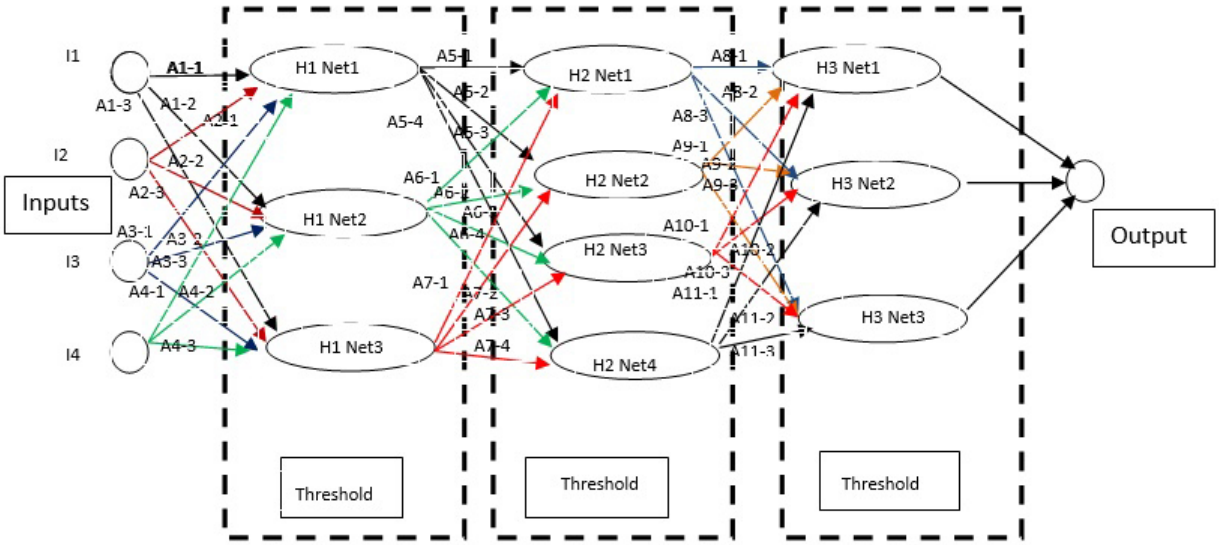


Figure 12.5: Neural network Example

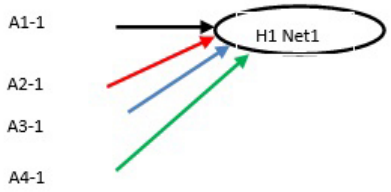


Figure 12.6: H1 Net1 network

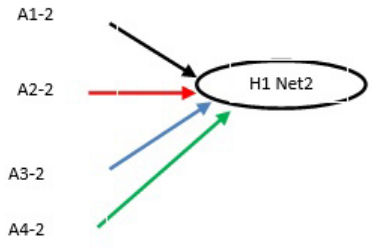


Figure 12.7: H1 Net2 network

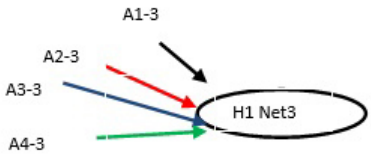


Figure 12.8: H1 Net3 network

$$H1Net3 = 0,910 * 0,017 + 0,851 * 0,532 + 0,534 * 0,569 + 0,952 * 0,079 = 0,8473$$

$$H1Net1 = \frac{1}{1+e^{-H1Net1}} = \frac{1}{1+e^{-1,8269}} = 0,8614$$

$$H1Net2 = \frac{1}{1+e^{-H1Net2}} = \frac{1}{1+e^{-1,6725}} = 0,8419$$

$$H1Net3 = \frac{1}{1+e^{-H1Net3}} = \frac{1}{1+e^{-0,8473}} = 0,7000$$

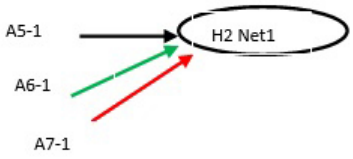


Figure 12.9: H2 Net1 network

$$H2Net1 = \sum_{i=1}^{3} H1Neti * A_i \rightarrow H1Net1 * A_{5-1} + H1Net2 * A_{6-1} + H1Net3 * A_{7-1}$$

$$H2Net1 = 0,8614 * 0,634 + 0,8419 * 0,690 + 0,7000 * 0,277 = 1,3209$$

$$H2Net2 = \sum_{i=1}^{3} H1Neti * A_i \rightarrow H1Net1 * A_{5-2} + H1Net2 * A_{6-1} + H1Net3 * A_{7-2}$$

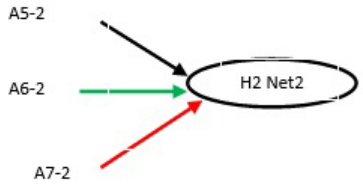


Figure 12.10: H2 Net2 network

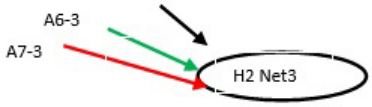


Figure 12.11: H2 Net3 network

$$H2Net2 = 0,8614 * 0,958 + 0,8419 * 0,590 + 0,7000 * 0,292 = 1,5263$$

$$H2Net3 = \sum_{i=1}^{3} H1Neti * A_{i} \rightarrow H1Net1 * A_{5-3} + H1Net2 * A_{6-3} + H1Net3 * A_{7-3}$$

$$H2Net3 = 0,8614 * 0,678 + 0,8419 * 0,411 + 0,7000 * 0,694 = 1,4158$$

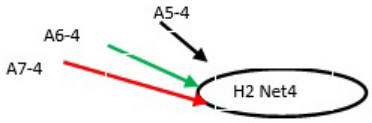


Figure 12.12: H2 Net4 network

$$H2Net4 = \sum_{i=1}^{3} H1Neti * A_{i} \rightarrow H1Net1 * A_{5-4} + H1Net2 * A_{6-4} + H1Net3 * A_{7-4}$$

$$H2Net4 = 0,8614 * 0,075 + 0,8419 * 0,100 + 0,7000 * 0,486 = 0,4889$$

$$F(H2Net1) = \frac{1}{1 + e^{-H2Net1}} = \frac{1}{1 + e^{-1.3209}} = 0,7893$$

$$F(H2Net2) = \frac{1}{1 + e^{-H2Net2}} = \frac{1}{1 + e^{-1.5263}} = 0,8215$$

$$F(H2Net3) = \frac{1}{1 + e^{-H2Net3}} = \frac{1}{1 + e^{-1.4158}} = 0,8046$$

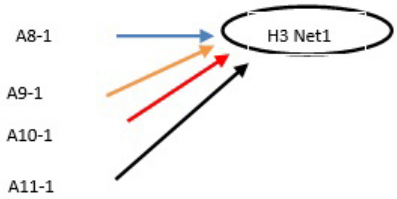


Figure 12.13: H3 Net1 network

$$F(H2Net4) = \frac{1}{1+e^{-H2Net4}} = \frac{1}{1+e^{-0.4889}} = 0,6198$$

$$H3Net1 = \sum_{i=1}^{4} I_i * A_i \rightarrow H2Net1 * A_{8-1} + H2Net2 * A_{9-1} + H2Net3 * A_{10-1} + H2Net4 * A_{11-1}$$

$$H3Net1 = 0,7893 * 0,907 + 0,8215 * 0,675 + 0,8046 * 0,467 + 0,6198 * 0,716 = 2,0899$$

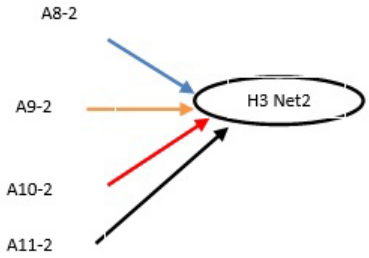


Figure 12.14: H3 Net2 network

$$H3Net2 = \sum_{i=1}^{4} I_i * A_i \rightarrow H2Net1 * A_{8-2} + H2Net2 * A_{9-2} + H2Net3 * A_{10-2} + H2Net4 * A_{11-2}$$

$$H3Net2 = 0,7893 * 0,541 + 0,8215 * 0,727 + 0,8046 * 0,208 + 0,6198 * 0,702 = 1,6695$$

Figure 12.15: H3 Net3 network

$$H3Net3 = \sum_{i=1}^{4} I_i * A_i \rightarrow H2Net1 * A_{8-3} + H2Net2 * A_{9-3} + H2Net3 * A_{10-3} + H2Net4 * A_{11-3}$$

$$\text{H3Net3} = 0,7893 * 0,942 + 0,8215 * 0,040 + 0,8046 * 0,254 + 0,6198 * 0,113 = 1,0508$$

$$\text{F(H3}Net1) = \frac{1}{1+e^{-H3Net1}} = \frac{1}{1+e^{-2.0889}} = 0,8899$$

$$\text{F(H3}Net2) = \frac{1}{1+e^{-H3Net2}} = \frac{1}{1+e^{-1.6695}} = 0,8415$$

$$\text{F(H3}Net3) = \frac{1}{1+e^{-H3Net3}} = \frac{1}{1+e^{-1.050873}} = 0,7409$$

12.6 Conclusion

Artificial neural networks have been developed to simulate the human nervous system for machine learning tasks by processing computational units in a learning model like human neurons. Especially with deep learning algorithms, it is aimed to develop algorithms that are closest to the human thinking structure and decision model. In this context, in addition to analytical computation power, deep learning algorithms in the development of a more complex decision mechanism, detecting, processing, and using visual objects in the decision process have taken these studies one step further.

Neural networks have some advantages as well as some disadvantages. These drawbacks can be listed as follows:

- What is in the system cannot be known.
- Stability analyzes cannot be performed except for some networks.
- May be difficult to apply to different systems.

In briefly,

- Deep Learning uses Artificial Neural Networks to mimic the intelligence of living things.
- There are three types of neuron layers in a neural network: Input Layer, Hidden Layers, and Output Layer.
- Connections between neurons are associated with a weight that determines the importance of the input value.
- Neurons use an Activation Function on the data to “standardize” the output from the neuron.
- A large dataset is needed to train the neural network.
- Iterating over the dataset and comparing the outputs will produce a Cost Function that shows how far the AI is from the actual outputs.
- After each iteration in the dataset, weights between neurons are adjusted using GradientDescent to reduce the cost function.

12.7 References

Balas, V.E., Roy, S.S., Sharma, D., Samui, P., Handbook of Deep Learning Applications, Springer, (2019), Volume 136, ISBN: 978-3-030-11478-7

Bulut, B., Kalın, V., Güneş, B. B., Khazhin, R., Deep Learning Approach For Detection Of Retinal Abnormalities Based On Color Fundus Images, 2020 Innovations in Intelligent Systems and Applications Conference (ASYU), (2020), pp. 1-6, DOI: 10.1109/ASYU50717.2020.9259870.

Baladi, H., Haghighat, A., Machine Learning Guide for Oil and Gas Using Python: A Step-by-Step Breakdown with Data, Algorithms, Codes, and Applications, (2021), Elsevier BV

Bhardwaj, A., Tiwari, A., Chandarana, D., Babel, D., A genetically optimized neural network for classification of breast cancer disease, 2014, 7th International Conference on Biomedical Engineering and Informatics, (2014), pp. 693-698, DOI: 10.1109/BMEI.2014.7002862.

Chow, M., Yee, S. O., Methodology for on-line incipient fault detection in single-phase squirrel-cage induction motors using artificial neural networks, in IEEE Transactions on Energy Conversion, vol. 6, no. 3, pp. 536-545, (Sept. 1991), DOI: 10.1109/60.84332.

Gençdoğmuş, A., Keskin, Ş. R., Doğan, G., Öztürk, Y., A Data-Driven Approach to Kinematic Analytics of Spinal Motion, 2019 IEEE International Conference on Big Data (Big Data), (2019), pp. 2222-2229, DOI: 10.1109/BigData47090.2019.9006164.

Goyal, P., Pandey, S., Jain, K., Deep Learning for Natural Language Processing, Springer Science and Business Media LLC, (2018), ISBN: 978-1-4842-3684-0

İş, H., Tuncer, T. A Profile Analysis of User Interaction in Social Media Using Deep Learning, February (2021), Traitement du Signal 38(1):1-11, DOI:10.18280/ts.380101

Korkmaz, S.A. Classification of histopathological gastric images using a new method. Neural Comput & Applic 33, 12007–12022 (2021). https://doi.org/10.1007/s00521-021-05887-x

Lazzeri, F., Machine Learning for Time Series Forecasting with Python, Wiley, 2020,

Mert, İ, Agnostic deep neural network approach to the estimation of hydrogen production for solar-powered systems, International Journal of Hydrogen Energy, Volume 46, Issue 9, 3 February 2021, Pages 6272-6285

Nandy, A. Biswas, M. "Neural Networks in Unity", Springer Science and Business Media LLC, 2018

Nokeri T.C. Introduction to Financial Markets and Algorithmic Trading. In: Implementing Machine Learning for Finance. (2021), Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-7110-0_1

Öztemel, E., Yapay Sinir Ağları, İstanbul: Papatya Yayıncılık Eğitim, 2012 xxii, 232s.

Qiao Huang, Limin Cui, Design and Application of Face Recognition Algorithm Based on Improved Backpropagation Neural Network, Revue d'Intelligence Artificielle Vol. 33, No. 1, February, (2019), pp. 25-32 Journal homepage: http://iieta.org/journals/ria

Rehbein, D.A., Maze, S.M., Havener, J.P. "The application neural networks in the process industry", ISA Transactions, 1992

Safer, A. M. 'A Comparison of Two Data Mining Techniques to Predict Abnormal Stock Market Returns'. 1 Jan. (2003), 3 – 13.

Sevi, M. and Aydın, İ. "COVID-19 Detection Using Deep Learning Methods," 2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI), (2020), pp. 1-6, DOI: 10.1109/ICDABI51230.2020.9325626.

Singh P. Introduction to Machine Learning. In: Deploy Machine Learning Models to Production. (2021) Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-6546-8_1

Sarang, P., "Artificial Neural Networks with TensorFlow 2", Springer Science and Business Media LLC, 2021

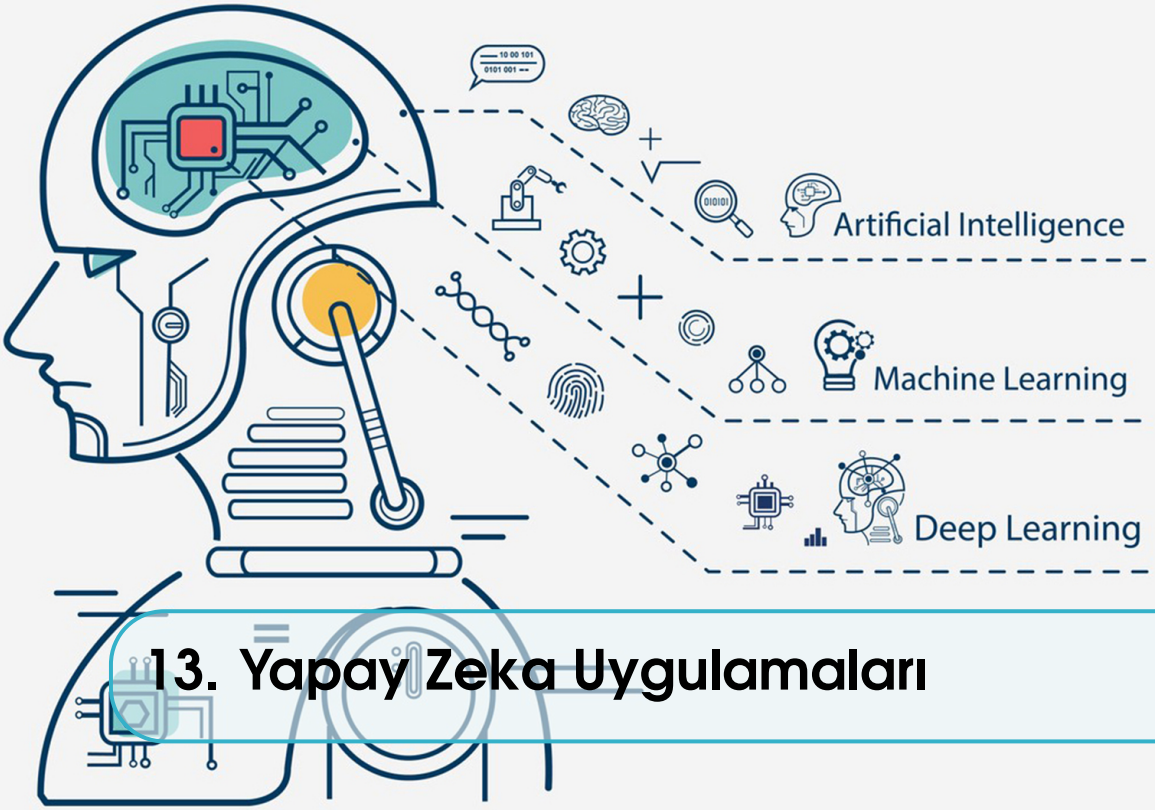
Silaparasetty, V., Deep Learning Projects Using TensorFlow 2, Neural Network Development with Python and Keras, Springer Science and Business Media LLC, 2020, ISBN: 978-1-4842-5802-6

Thimira A., T. "Deep Learning on Windows", Springer Science and Business Media LLC, (2021), ISBN: 978-1-4842-6431-7

Umberto M., Advanced Applied Deep Learning", Springer Science and Business Media LLC, 2019

Wilamowski, B.M., & Irwin, J.D. (Eds.). Intelligent Systems: The Industrial Electronics Handbook (2nd ed.). (2011). CRC Press. https://doi.org/10.1201/9781315218427

Zhou, M.L., Gao, W. and Liu, F.Hybrid control method for hysteresis non-linearity of magnetically controlled shape memory actuator, Published Online: July 27, (2012), pp 77-91



13. Yapay Zeka Uygulamaları

Yapay Zeka Uygulamaları - Akciğer Kanserinin Tespit ve Teşhisi için Örnek Bir Uygulama

Tijen ÖVER ÖZÇELİK*, Alpaslan KİBAR*, Mehmet Fatih AKCA*

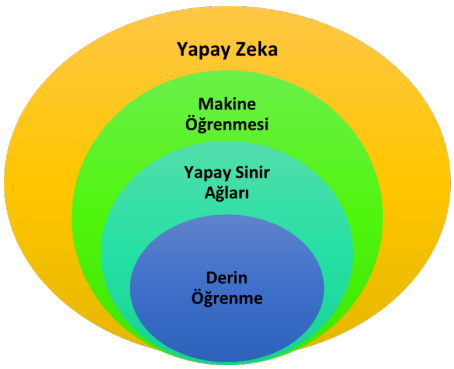
***Sakarya Üniversitesi**

13.1 Giriş

Yaklaşık 40 yıl önce literatüre giren 'Yapay Zeka' kavramı ve barındırdığı teknolojiler özellikle son 20 yılda gerçekleştirilen çalışmalarla, zor problemleri çözebilir, olaylara farklı açılardan bakabilir, zeki programlar ve zeki sistemler üretebilir konuma gelmiştir. Yapay zeka teknolojileri; tahminlemeyi mümkün kılarak, verileri modelleyebilmekte ve çok büyük ölçekli verileri anlamlandırabilmekte, aynı zamanda bunu her sektör ve işletme için sağlayabilmektedir. Kısacası yapay zeka teknolojileri; tıptan eğitime, ulaşımdan sanayiye her alanda kendisini etkin ve güçlü bir şekilde göstermektedir. Bu teknolojilerin sahip olduğu algoritmalar davranış kalıplarını modelleyerek kendi mantığını geliştirebilmektedir. Doğru ve tutarlı tahminleme de bulunabilmek için bu modellerin büyük miktarda ki veri ile eğitilmesi gerekmektedir. Bu teknolojilerle gelişen ve dijitalleşen sağlık uygulamaları da dünya genelinde bu değişimden etkilenmekte, makine öğrenimi ve yapay zeka ile doktorlar, hastaneler ve sağlıkla bağlantılı tüm alanlar daha verimli hale getirilerek toplum yararına sunulmaktadır. Bu bağlamda çalışmada öncelikle; makine öğrenmesi, yapay sinir ağları, derin öğrenme, evrişimsel sinir ağları teorisinden bahsedilmiş ve sağlık alanında derin öğrenmeyi temel alan bir uygulama gerçekleştirilmiştir. Uygulama kapsamında ilk olarak kişinin akciğer kanseri olup olmadığını tespite dayalı geliştirilen modeller ele alınmış ve eğer kişi kanser ise kanserin yerini tespite dayalı çalışmaları ve sonrasında da evresinin belirlenmesine dayalı çalışmalar gerçekleştirilmiştir.

13.2 Yapay Zeka

Yapay Zeka, canlıların problemlere çözüm olarak ürettikleri akıllı davranışları modelleyerek, kümeleme, sınıflandırma, tahmin vb. yöntemler ile bilişim problemlerine çözümler önerebilir. Otomatik olarak, yazım hatalarının düzeltilmesi için öneri oluşturan metin editörleri, insanların tam olarak ne istediklerini anlayan arama algoritmaları, sıkça sorulan soruları otomatik olarak cevap verebilen chatbotlar, sanal yardımcılar (dijital asistanlar), yüz algılama/tanıma uygulamaları, akıllı elektronik banka uygulamaları ve bunun gibi bir çok yerde yapay zeka kullanılmaktadır. Eskiden daha çok uzman sistemler olarak karşımıza çıkan yapay zeka günümüzde konuşma, tanıma, bulanık mantık, genetik algoritmalar, robotik uygulamalar, bilgisayarlı görme/görüntü tanıma gibi birçok alanda kullanılmaktadır. Günümüzde ise makine öğrenmesi, derin öğrenme, yapay sinir ağları vb. yöntemler sağlık, finans, eğitim, askeri vb. alanlarda daha çok duyulmaktadır (ATALAY & ÇELİK, 2017). Yapay zeka ile ilgili konu ve çalışmalarda yoğun olarak ele alınan makine öğrenmesi ve derin öğrenme de yapay zekanın kapsadığı alanlardır. Derin öğrenme yapay zekanın kapsadığı bir konu olmakla beraber makine öğrenmesinin de bir alt alanıdır ve bu alanlar, tahmine dayalı sistemler oluşturmayı amaçlayan yapay zeka algoritmalarını oluşturur. Yapay Zeka, Makine Öğrenmesi, Yapay Sinir Ağları ve Derin Öğrenme günümüzde popüler kavramlardır. Bu kavramlar yakından ilişkili olmakla birlikte aralarında bazı farklılıklarda mevcuttur. Şekil 1, bu farklılığı görsel olarak ifade etmektedir.



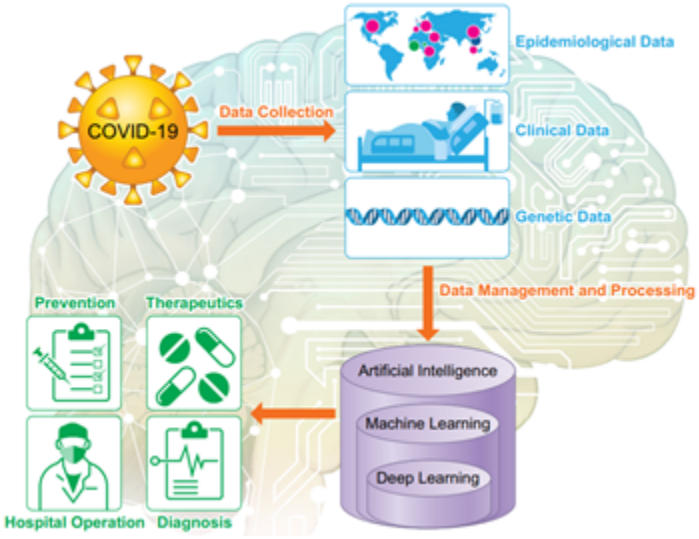
Şekil 13.1: Yapay Zeka, Makine Öğrenmesi, Yapay Sinir Ağları ve Derin Öğrenme arasındaki ilişki

Günümüzde COVID-19 pandemisi nedeniyle, sağlık alanında özellikle makine öğrenmesi teknikleriyle yapılan birçok çalışma göze çarpmaktadır. Şekil 2' de Ahmad Alimadadi ve arkadaşlarının COVID-19' un tespit/teşhis/tedavisi için önerdiği model (ALIMADAI, ve diğerleri, 2020) görülmektedir.

13.3 Makine Öğrenmesi

İnsanların öğrenme yeteneğini taklit edebilen ve veriler üzerinden çıkarımlar yapabilmek amacıyla geliştirilen algoritmaları ve çalışmaları inceleyen, gelen yeni verilere göre mevcut algoritmalar üzerinde güncelleme yapabilen bir yapay zeka teknolojisidir. Bu algoritmalar, basit program komutlarını bire bir uygulayan klasik yaklaşımın yerine, örnek verilerden tahminler üretmek, kararları oluşturabilmek için model oluşturma/eğitme/test etme adımlarını uygularlar.

Makine öğrenmesinde temel yaklaşım, sonucu bilinen verilerden hareketle, sonucu bilinmeyen verilerin sonuçları en doğru şekilde tahmin edecek algoritmayı geliştirmektir. Veri madenciliğinde



Şekil 13.2: Yapay Zeka tekniklerinden Makine Öğrenmesinin COVID-19 un tespit/teşhisi/tedavisi vb. için önerilen bir model

ise, veriler arasında daha önce bilinmeyen örüntüler, ilişkiler tespit edilerek, bilgiye ulaşılmaya çalışılır. Literatürde birçok makine öğrenmesi bulunmaktadır.

Denetimli: Bu teknik, etiketli veriler kullanılarak tahminler üretme prensibine dayanır. Bu tekniğin dezavantajı eğitim verisinin hazırlanmasının büyük zaman maliyete ihtiyaç duymasıdır.

Örneğin parça üretimi yapan bir fabrikada hatalı parçaların resimlerinden (etiketli) oluşan eğitim verileri ile sistem eğitilip sisteme hatalı parçalar öğretilir. Daha sonra içerisinde hem hatalı hem de hatasız parçalardan oluşan test veri seti için sistemden hangi resimlerin hatalı parça resmi olduğuna dair tahmin yapması istenir.

Denetimsiz: Bu teknikte veriler etiketli değildir. Girdi verilerinin daha önce belirlenmiş sınıfları yoktur. Eğitilmemiş bu veriler üzerinde çalışılarak, verilerin anlam/içerik/değer olarak kümelenmesi prensibine dayanır. Yeni veriler algoritma tarafından oluşturulan bu kümelere dağıtılır.

Eğitim verisi hazırlanmasının büyük yük olduğu durumlarda zamandan ve maliyetten tasarruf sağlar. Marketlerde bu satılırsa yanında şu da satılabilir şeklinde (apriori) algoritmalar kullanılabilir olsalar da müşteri sayısı çok olduğunda bu eğitilmiş verileri hazırlamak karmaşık olabilir, zaman alabilir, maliyetli olabilir. Bu gibi durumlarda benzer alışkanlıkları/özellikleri olan müşterilerin gruplandırılması nispeten daha kolay olabilir. Oluşan grupların özelliklerine göre raflar düzenlenebilir, stoklar planlanabilir.

Yarı Denetimli: Etiketlenmiş küçük miktardaki veriye karşın etiketlenmemiş büyük miktardaki veriden oluşan veri setlerinde kullanılabilir.

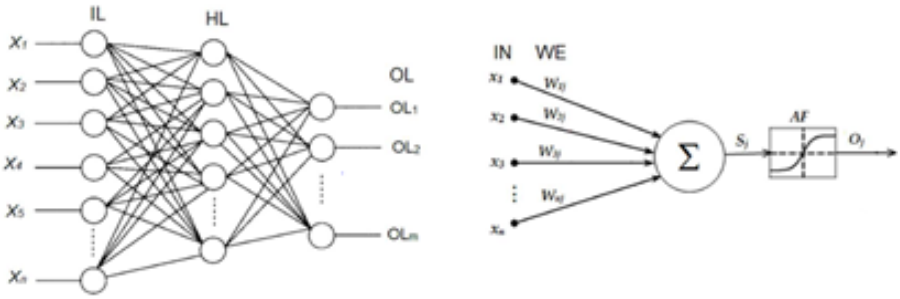
Takviyeli Öğrenme: Öğrenme, geri bildirimler yoluyla sağlanır. Olası durumlar vardır. Elde edilen geri bildirim istenen durum değil ise ceza yazılır ve o geri bildirimi tetikleyen hamle tekrar edilmez. Alınan geri bildirim istenen durum ise ödül yazılır ve bu deneyimden faydalanarak öğrenme işlemi devam eder. Amaç en fazla ödülü almaktır. Öğrenme işlemi süreklidir. Öğrenmeyi durdurma gibi bir işleme ihtiyaç duyulmaz.

Derin: GPU (Grafik İşlemci Ünitesi) kullanarak büyük miktardaki etiketlenmiş veri üzerinden, derin sinir ağları kullanarak yapılabilen öğrenmedir. Çok daha az güç ve altyapı ile çok daha

büyük hacimli veriler üzerinde çalışarak sınıflandırma ve tahmin yapabilir. Denetimsiz öğrenme de yapabilir.

13.4 Yapay Sinir Ağları (YSA)

Sinir ağları – ya da yapay sinir ağları (YSA) – bir dizi algoritma aracılığı ile insan beynini taklit etmeye dayalıdır. En temel düzeyde bir sinir ağı, dört ana bileşenden oluşur. Bunlar; girdiler, ağırlıklar, bir önyargı/eşik ve çıktıdır. YSA'lar, birbiri ile bağlantılı çok nöronlu (düğümlü) bir yapıdadır. Düğümler girdi verilerini alarak bu veriler üzerinde basit işlemler gerçekleştirebilir ve bu işlemlerin sonucu diğer nöronlara iletilir. Her düğümdeki çıktı, aktivasyon veya düğüm değeri olarak adlandırılır. Her girdi kendi ağırlığı ile çarpılır, tüm girdilere yapılan bu işlemin toplam fonksiyonu düğüm (nöron) için net girdiyi verir. Birçok YSA modelinde bu toplam fonksiyonu transfer fonksiyonu olarak kabul edilse de transfer fonksiyonunun kullanıldığı modeller de vardır. Bu transfer fonksiyonunda Bias gibi düzeltmeler de kullanılabilir. Transfer fonksiyonundaki değer aktivasyon fonksiyonuna gönderilir. Aktivasyon fonksiyonu sonucu elde edilen değer düğümün Y için (birden çok düğüme çıktı gönderebilir) çıktısını oluşturacaktır. YSA' da asıl amaç, düğümlere gelen girdi (X1, X2,... Xn) değerlerine göre, kabul edilebilecek doğrulukta çıktı değerlerini üretebilecek ağırlıkların (W1, W2,... Wn) belirlenebilmesidir ki buna öğrenme denir. Uyum gösterme (yeni türler vb.) ve sürekli öğrenme (yeni veriler vb.), eğitim verisinin kısıtlı olduğu ve yeni verilerin sürekli eklendiği durumlarda çok çok önemlidir. Ayrıca bir düğüm sadece tek bir düğüm tarafından etkilenmediğinde, etkileyici düğümlerden birindeki yetersiz veya bozuk veri sonucu çok fazla etkilemeyebilir. Aşağıda Şekil 3'de yapay sinir ağının yapısı görsel olarak ifade edilmiştir (CEBECİ, 2020).



Şekil 13.3: Yapay Sinir Ağının Yapısı

13.5 Derin Öğrenme

Derin öğrenme, insanların öğrenme şeklini taklit eden bir tür makine öğrenmesi ve yapay zeka algoritmasıdır. Derin öğrenme, istatistik ve tahmine dayalı modellemeyi mümkün kılarak veri bilimi için önemli bir unsur haline gelmiştir. Büyük miktarda veriyi toplama, analiz etme ve yorumlama sürecinden sorumlu olan veri bilimciler için son derece faydalı bir yöntemdir. Çünkü derin öğrenme bu süreci daha hızlı ve kolay hale getirmektedir. Derin öğrenme görüntü işleme-tanıma, konuşma tanıma ve doğal dil işlemedeki birçok soruna en iyi çözümleri sunmaktadır. Örneğin, evrişimsel sinir ağı olarak bilinen bir derin öğrenme modeli, atları içerenler çok sayıda (milyonlarca) görüntü kullanılarak eğitilebilir. Bu tür bir ağ, elde ettiği görüntülerdeki piksellerden öğrenir. Bir atın

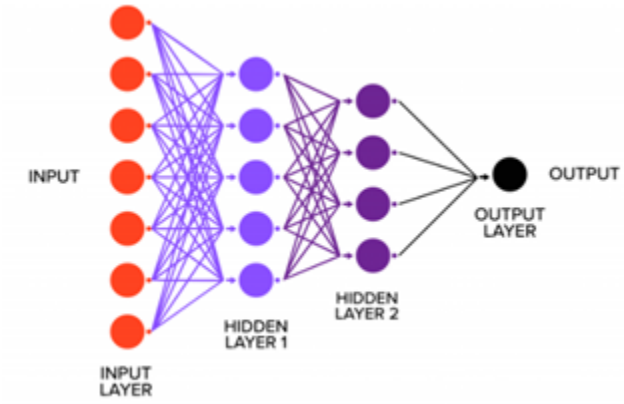
özelliklerini temsil eden piksel gruplarını, bir görüntüde bir atın tanımlayıcısı olan yeleler, kulaklar, toynaklar ve gözler gibi özellik gruplarıyla sınıflandırabilir. Derin öğrenme en klasik anlamda makine öğrenmesi içinde ele alınmakta, yukarıdaki örnekte görüldüğü gibi özellik çıkarma ve özellikleri; sınıflama, desenleme için birçok doğrusal olmayan katmanını kullanmaktadır.

Denetimli (sınıflandırma gibi) veya denetimsiz (desen analizi gibi) algoritmalar oluşturabilir (ŞEKER, DİRİ, & BALIK, 2017). Günümüzde büyük miktarlarda veriye ulaşma imkanı ve süper bilgisayar performansları sayesinde (özellikle GPU' daki gelişmeler) popülaritesi artan bir yöntemdir. Konuşmayı tanıma, konuşmacıyı tanıma, görüntü tanıma vb. alanlarda kullanılabilmektedir. Birden fazla katman ve çok fazla düğüm (bir milyardan fazla düğüme sahip bir derin öğrenme ağı için 16.000 bilgisayar kullanılmış – Google Brain) içerebilir. Google 2015 yılında konuşma tanıma uygulamasında CCT-LSTM tekrarlı yapay sinir ağı modelini kullanmıştır (AKPINAR, 2017).

Öznitelik seçimi en küçük kareler yöntemi ile yapılmakta, düğüm sayısı girdi katmanından çıktı katmanına doğru ilerlerken azaltılmaktadır (GLAUNER, 2015). Tekrarlı Yapay Sinir Ağı yöntemidir. Yolun derinliği gizli katmanların sayısının bir fazlasıdır (çıktı katmanı +1 olarak ekleniyor). Katman sayısı 2' den fazla olduğunda derin ağ, 10' dan fazla olduğunda çok derin ağ olarak tanımlanabilir.

GPU, vektör ve matris hesaplamaları ile grafik işlemlerinde klasik CPU' ya göre 10 ile 100 kat arası daha iyi performanslar gösterebilmektedir. GPUlar paralel işlem yapma yeteneğine sahiptirler. Açık kaynak kodlu OpenCL ile GPU kullanarak hesaplama yapmak çok daha kolaylaşmıştır.

Yapay Sinir ağları ve derin öğrenme algoritmaları ile Metin çeviri/tercüme, Doğal dil işleme problemleri çözülmeye çalışılabilir. Girdi katmanı, Gizli katman (derin öğrenmede gizli katman sayısı birden fazla) ve çıktı katmanından oluşan bu algoritmalarda hesaplama yapılırken paralel işlemler yapılabileceğinden (derin öğrenme GPU kullanabilir) diğer yöntemlerden farklıdır. (ATALAY & ÇELİK, 2017) Şekil 4'de Derin öğrenmenin gizli katman yapısı görsel olarak ifade edilmiştir.

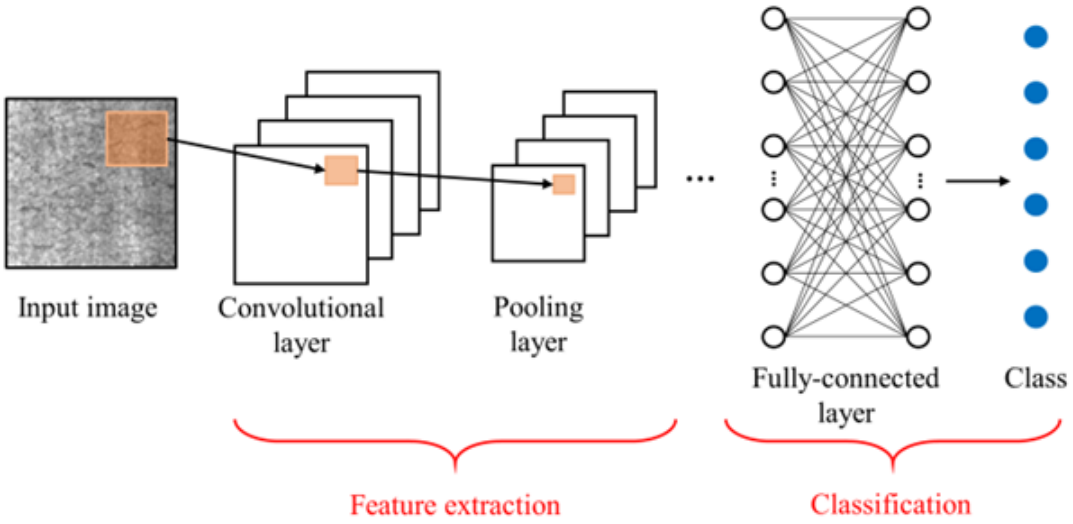


Şekil 13.4: Derin Öğrenmenin Gizli Katman Yapısı

13.6 Evrişimsel Sinir Ağları

Derin öğrenmeye dayalı bilgisayarlı bir görme modelidir. Girdi olarak görselleri kullanır. Evrişimsel Sinir Ağları (Convolutional Neural Network/CNN), yapay sinir ağlarındaki gibi güncellenerek eğitilebilen (öğrenen) yapısıyla görüntü işlemede rağbet görmektedir. Veriler girdi olarak sinir ağındaki nöronlar tarafından toplanır, işlenir, uygun algoritmalarla hesaplanan sonuçlar sonraki nöronlara gönderilirler. Yapay Sinir Ağının başında bulunan kaynak görüntünün özellikleri (feature)

işlenerek ağın sonundaki sınıf skor değeri hesaplanır. Elde edilen bu değer, tamamen bağlı (full connected) katmanda işlemlerden geçirilerek görüntünün hangi sınıfın bir üyesi olabileceği tahmini oluşturulur. Aslında yapılan işlem kısaca, Convolutional, Pooling ve Fully Connected katmanlarından geçirilen görüntülerin verilerinin derin öğrenme modelinde işlenerek tahminin oluşturulması olarak özetlenebilir. İlk olarak Yann LeCun ve arkadaşları tarafından oluşturulup denenen ve başarılı sonuçlar veren derin öğrenme modeli (ŞEKER, DİRİ, & BALIK, 2017), yapısal olmayan verileri kaynak olarak kullandığından çok fazla veriye ihtiyaç duymakta bu da diğer makine öğrenmesi algoritmalarına göre verilerin işlenmesi için çok daha fazla zaman gereksinimi oluşturmaktadır.

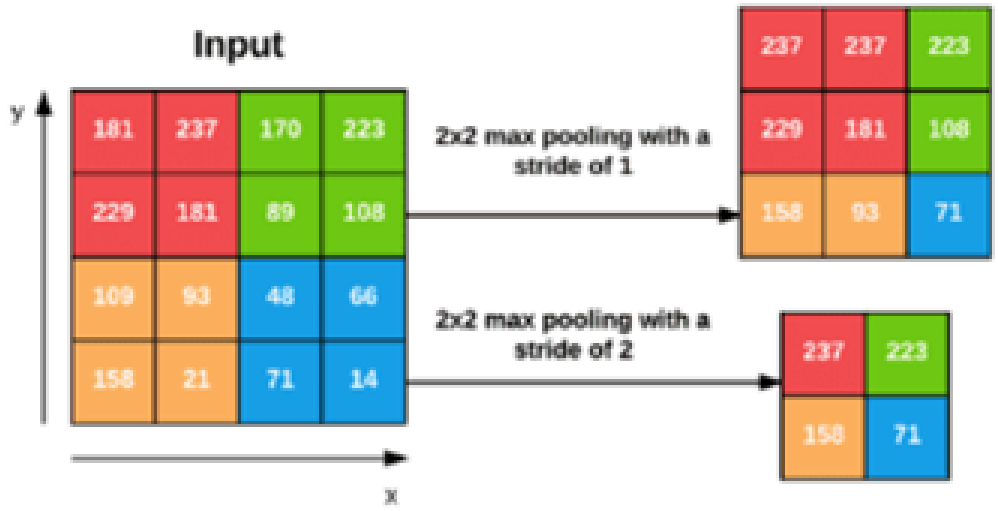


Şekil 13.5: Evrişimsel sinir ağlarının (CNN) katman yapısı

Convolution katmanında filtreler yardımı ile görüntünün özellikleri belirlenirken (özelliklerin değerleri tutulur) daha başarılı olunması için bazı filtrelerle kenar tespiti yapma, blur ekleme gibi işlemler de yapılabilir. Şekil 5'de Evrişimsel sinir ağlarının katman yapısı (Soo Young Lee, 2019) görselleştirilmiştir. Şekil 6'da ise Max Pooling işlemi grafiksel olarak açılmış ve ifade edilmiştir (Rosebrock, 2021).

13.7 Hızlı Bölgesel Tabanlı Evrişimsel Sinir Ağı (Faster R-CNN)

R-CNN'ler bazı eksiklikleri (yavaşlık, hata vb.) nedeniyle geliştirilme/iyileştirilme ihtiyacı hissettirmişlerdir. Hızlı R-CNN olarak adlandırılan yaklaşımlardan biri genel olarak R-CNN' e benzese de, evrişimli bir özellik haritası oluşturmak amacıyla kaynak görüntüsü ile CNN' i besler. Konvolüsyon özellik haritasından elde edilen öneri bölgeleri bulunur ve bunlar kareler ile işlenerek ve sonrasında bir ROI havuzlama katmanı kullanılarak tam olarak bağlı oldukları katmanı sabit bir boyutta yeniden şekillendirmek için beslerler. Önerilen bölgenin sınıfını ve sınırlayıcı kutuyu oluşturmak amacıyla, ofset değerlerini tahmin etmek için, ROI özellik vektöründen, softmax katmanı kullanılır (AALAMI, 2020).



Şekil 13.6: Max Pooling işlemi

13.8 YOLO (You Only Look Once)

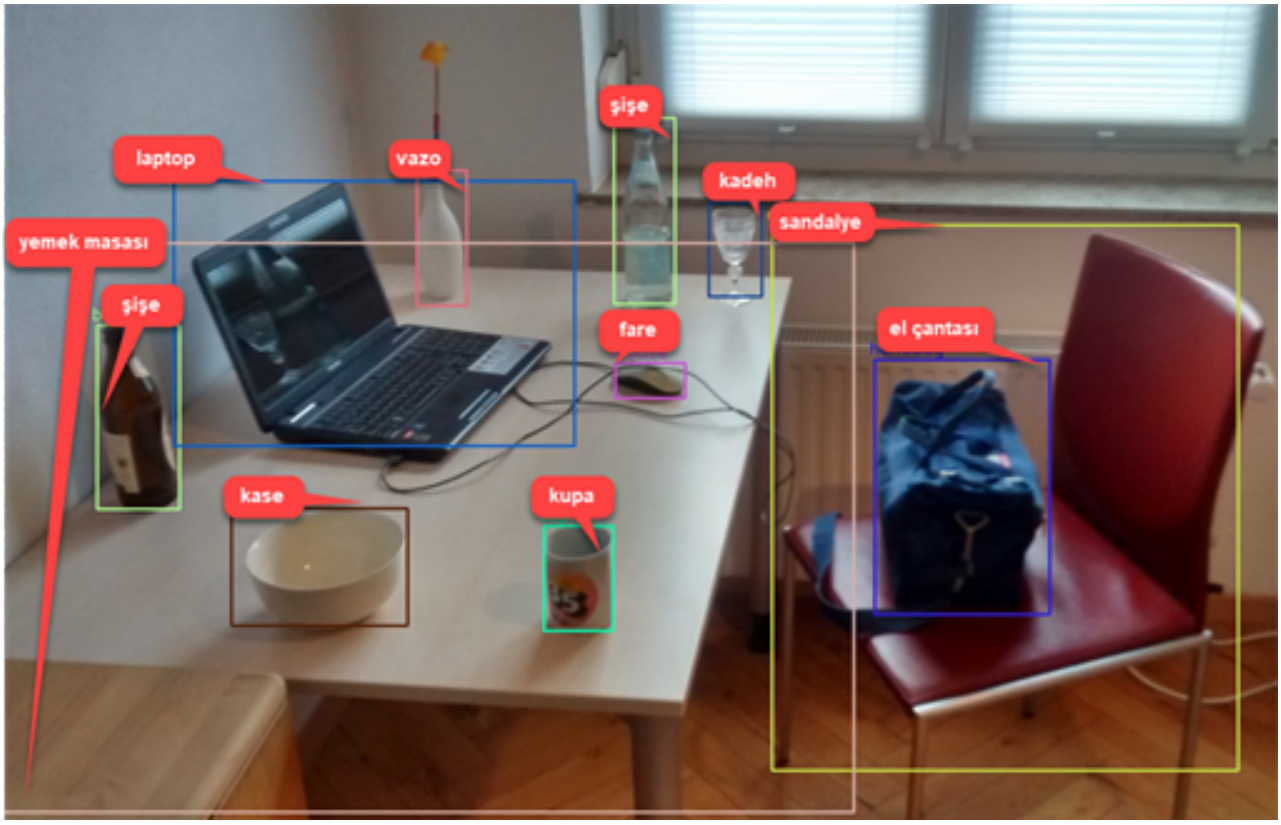
Yolo algoritmasının diğer algoritmalardan çok daha doğru tahminde bulunduğu iddia edilemez zira eskiden de günümüzde de YOLO' dan daha doğru tahminde bulunabilen algoritmalar mevcuttur. Fakat YOLO algoritmasının diğer algoritmalara olan üstünlüğü hızı sayesinde daha kısa sürede görüntüyü işlemesidir. Aynı zamanda gerçek zamanlı tahminler yapabilen az sayıdaki algoritmalardan biridir. YOLO' nun diğer algoritmalara göre hızlı olmasının sebebi resimlere sadece 1 kez bakıyor olmasından kaynaklanmaktadır. R-CNN gibi bölge bazlı nesne tespit eden algoritmalar ilk aşamada aranan nesnenin kaynak görüntü üzerindeki muhtemel koordinatlarını belirlerler daha sonraki aşamada ise CNN algoritmasını o kaynak görüntü üzerinde çalıştırırlar. Sonuç olarak bu algoritmalar aynı kaynak görüntü üzerinde 2 defa işlem yaparlarken YOLO algoritması kaynak görüntü üzerinde yaptığı tek bir işlem sonrası derin öğrenme ağından geçirmekte ve bu sayede daha kısa sürede sonuca ulaşabilmektedir. Aşağıda Şekil 7'de YOLO tekniği ile nesne tespiti şekilsel olarak verilmiştir.

13.9 Örnek Uygulama - Akciğer Kanserinin Tespiti ve Teşhisi

Örnek uygulama kapsamında, akciğer kanserinin tespiti ve mevcut kanserin evresinin belirlenmesi işlemi gerçekleştirilmiştir. Bu amaçla python kodlarından faydalanılmıştır. Bu amaçla; yapay zeka tekniklerinden makine öğrenmesi, makine öğrenmesi yöntemlerinden yapay sinir ağı, bir çeşit yapay sinir ağı olan derin öğrenme, derin öğrenme ile görüntü işleme yöntemlerinden biri olan YOLO kullanılmıştır. Uygulama Google.colab üzerinde gerçekleştirilmiş ve pyhton kütüphaneleri kullanılmıştır. Uygulama; hazırlık, veri ön işleme ve tahminleme olmak üzere üç kısımdan oluşmaktadır. Kullanılan imaj verileri `https://wiki.cancerimagingarchive.net/` sitesinden alınmıştır.

Birinci Kısım: Hazırlık

Uygulama Google.colab üzerinde oluşturulmuştur. Colab, makine öğrenmesi ve derin öğrenme çalışmalarında kullanılmak amacı ile tasarlanmış çevrimiçi ve bulut tabanlı bir platformdur. Bu sayede makine öğrenmesi için gerekli tüm ortak ve yüklü kütüphanelere ulaşmak mümkün olur.



Şekil 13.7: YOLO tekniğinde nesne tespiti (Kaynak; `https://tr.wikipedia.org/wiki/Nesne_tespiti`)

```
from google.colab import drive
drive.mount('/content/drive')
```

Öncelikli olarak, medikal görüntü işlemi formatı olan dicom için gerekli python kütüphanesi yüklenir.

```
!pip install pydicom
```

Bu aşamada kullanılması planlanan python kütüphaneleri çalışma kapsamına dahil edilmiş olur.

```
import pydicom
import numpy as np
import tensorflow as tf
import matplotlib.pyplot as plt
import cv2
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.resnet50 import ...
        preprocess_input, decode_predictions
```

Bu aşamada dicom dosyası okunarak, jpg formatında kaydedilir.

```
def read_dicom(image_path):
        img = pydicom.read_file(image_path)
        plt.imsave("/content/image.jpg", img.pixel_array, ...
                cmap = "gray")
```

Daha önce eğitimi gerçekleştirilmiş derin öğrenme modelinin çalışmaya alınması için kullanılan fonksiyon çağrılır.

```
def load_model(model_path):
        return tf.keras.models.load_model(model_path)
```

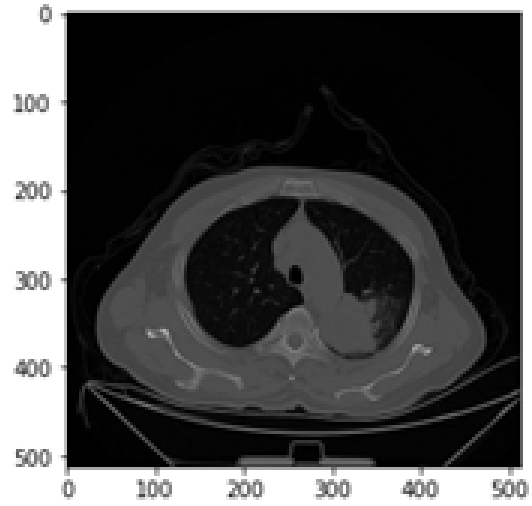
Dicom formatından jpg formatına getirilen görüntünün çalışmaya aktarılmış model ile tahmini gerçekleştirilir.

```
def predict(img_path, model):
        read_dicom(img_path)
        img = image.load_img('/content/image.jpg',
target_size=(512,512))
        plt.imshow(img)
        plt.show()
        img_array = image.img_to_array(img)
        img_batch = np.expand_dims(img, axis=0)
        img_preprocessed = preprocess_input(img_batch)
        return model.predict(img_preprocessed)
```

Eğitilmiş modelin kayıtlı olduğu yol verilir, bu sayede python o dosyaya gider ve modeli çalışmaya aktarır. Çalışmaya aktarılan modele dicom formatındaki görsel ve model değişkeni verilir. Çıktı olarak CNN modelinin tahmin sonucu alınır. Şekil 8' CNN ile oluşturulan modelin tahmin sonucu görülmektedir.

```
cnn_model_path = ...
        "/content/drive/MyDrive/sunum/model/h5file/model.h5"
```

```
model = load_model(cnn_model_path)
print(predict("/content/drive/MyDrive/sunum/1-19.dcm",model))
```



Şekil 13.8: CNN ile oluşturulan modelin tahmin sonucu

İkinci Kısım: Veri Ön İşleme

Bu aşamada kullanılacak kütüphaneler çalışmaya alınır.

```
import os
import pandas as pd
from skimage import measure, morphology
from scipy import misc
from sklearn.cluster import KMeans
```

Görsel için gerekli ön işleme süreçlerini yöneten fonksiyon oluşturulur. Gerçekleşen işlemler sırası aşağıda verilmiştir.

- Görselin Okunması,
- Blur atılması,
- Görselin normalize edilmesi,
- Kmeans algoritması kullanılarak görüntüdeki kemik ve hava gibi gereksiz noktaların ayrıştırılması,
- Görselin binary formatına getirilmesi için threshold atılması,
- Erozyon ve genişleme uygulanması,
- Farklı segmentleri daha rahat görebilmek için renklendirilmesi,
- Ciğeri doldurmak için son genişleme uygulanması ve
- Oluşan görüntünün ekrana yazdırılması.

Fonksiyon tanımlamanın ilk aşamasında fonksiyon adı verilerek istenen parametre tanımlanır.

```
def preprocess(img_path):
```

Fonksiyonun içerisindeki görsel okunur.

```
img_org = pydicom.read_file(img_path)
img = img_org.pixel_array
```

Blur atılır.

```
img = cv2.blur(img,(5,5))
row_size= img.shape[0]
col_size = img.shape[1]
mean = np.mean(img)
std = np.std(img)
img = img-mean
img = img/std
```

Görselin normalizasyonu yapılır.

```
middle = img[int(col_size/5):int(col_size/5*4), ...
         int(row_size/5):int(row_size/5*4)]
mean = np.mean(middle)
max = np.max(img)
min = np.min(img)
img[img==max]=mean
img[img==min]=mean
```

Kmeans algoritması kullanılarak görüntüdeki gereksiz noktaların ayrıştırılması (kemik hava vb.) yapılır.

```
kmeans = KMeans(n_clusters=2).fit(np.reshape(middle, ...
         [np.prod(middle.shape),1]))
centers = sorted(kmeans.cluster_centers_.flatten())
threshold = np.mean(centers)
```

Görselin binary formatına dönüştürülmesi için threshold işlemi yapılır.

```
thresh_img = np.where(img<threshold,1.0,0.0) *(-1)
```

Erozyon ve genişleme uygulanır.

```
eroded = morphology.erosion(thresh_img,np.ones([3,3]))
dilation = morphology.dilation(eroded,np.ones([8,8]))
```

Farklı segmentlerin renklendirilmesi amacıyla renk atanması yapılır.

```
labels = measure.label(dilation)
label_vals = np.unique(labels)
regions = measure.regionprops(labels)
good_labels = []
for prop in regions:
B = prop.bbox
if B[2]-B[0]<row_size/10*9 and B[3]-B[1]<col_size/10*9 and ...
        B[0]>row_size/5 and B[2]<col_size/5*4:
good_labels.append(prop.label)
mask = np.ndarray([row_size,col_size],dtype=np.int8)
mask[:] = 0
```

Ciğeri doldurmak için son bir genişletme yapılır.

```
for N in good_labels:
        mask = mask + np.where(labels==N,1,0)

mask=morphology.dilation(mask,np.ones([10,10]))
fig, ax = plt.subplots(3, 2, figsize=[12, 12])
ax[0, 0].set_title("Original")
ax[0, 0].imshow(img, cmap='gray')
ax[0, 0].axis('off')
ax[0, 1].set_title("Threshold")
ax[0, 1].imshow(thresh_img, cmap='gray')
ax[0, 1].axis('off')
ax[1, 0].set_title("After Erosion and Dilation")
ax[1, 0].imshow(dilation, cmap='gray')
ax[1, 0].axis('off')
ax[1, 1].set_title("Color Labels")
ax[1, 1].imshow(labels)
ax[1, 1].axis('off')
ax[2, 0].set_title("Final Mask")
ax[2, 0].imshow(mask, cmap='gray')
ax[2, 0].axis('off')
ax[2, 1].set_title("Apply Mask on Original")
ax[2, 1].imshow(mask*img, cmap='gray')
ax[2, 1].axis('off')
plt.show()
```

Fonksiyon çalıştırılır.

```
preprocess("/content/drive/MyDrive/sunum/1-19.dcm")
```

Aşağıda Şekil 9'da Görüntü üzerinde gerçekleştirilen işlemler sırası ile ve görselleştirilerek bir çıktı olarak üretilmiş ve ifade edilmiştir.

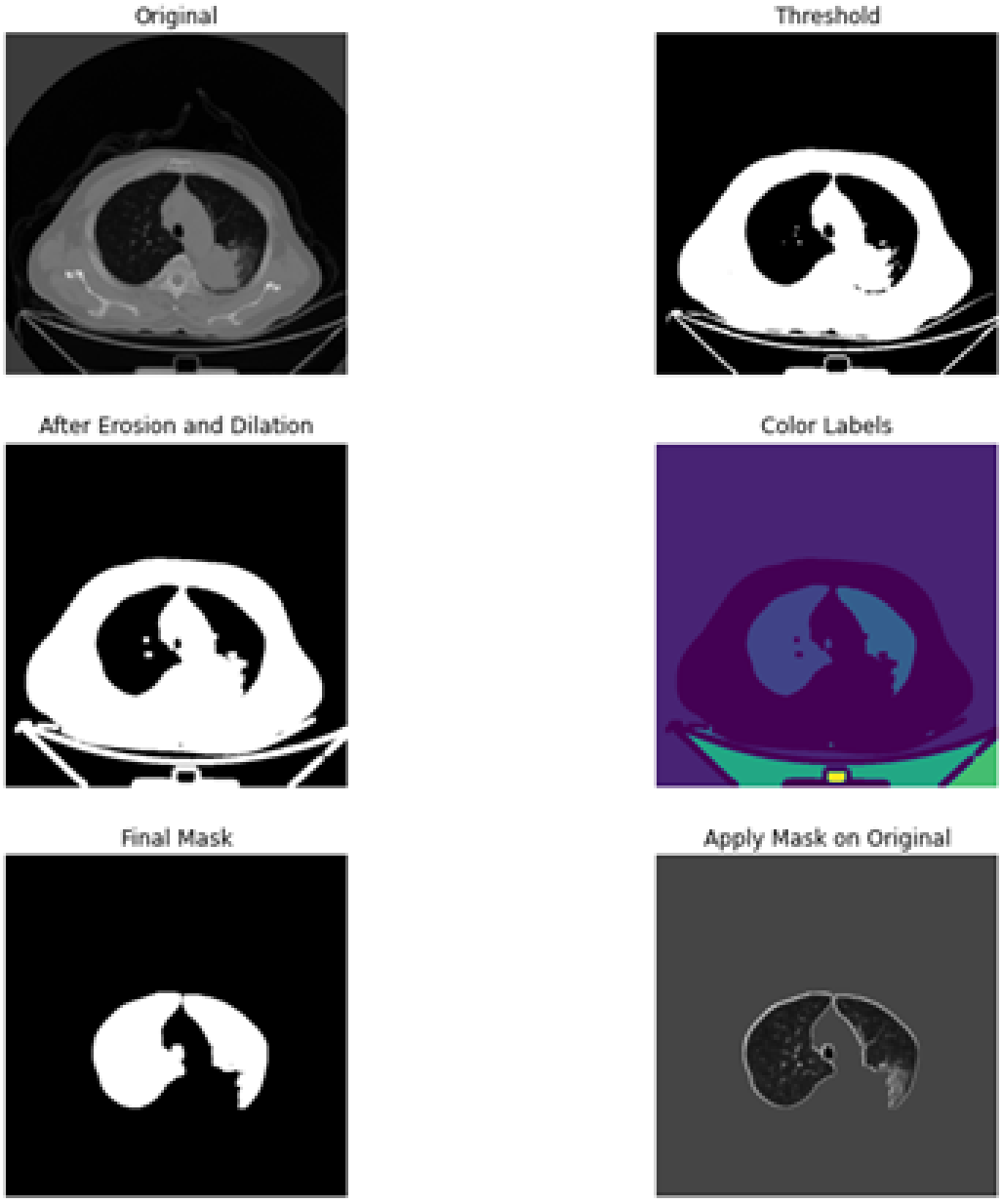
Üçüncü Kısım: Tahminleme

Bu aşamada, kişi eğer kanser ise hangi bölgede kanser olduğunu tespit etmek için eğitilen YOLO modeli ile görselin oluşturulması.

```
def segmente(image_path):
        cnn_model_path = ...
                "/content/drive/MyDrive/sunum/model/h5file/model.h5"
        model = load_model(cnn_model_path)
        pred_label = predict(image_path, model)[0][0]
```

Kanser ise;

```
if pred_label == 1:
!base64 /content/image.jpg | curl -d @-
"https://detect.roboflow.com/yolotransfer/
2?api_key=nG25HzY3eyJE4cE9kySt" > /content/output.txt
else: print('non-cancer')
```

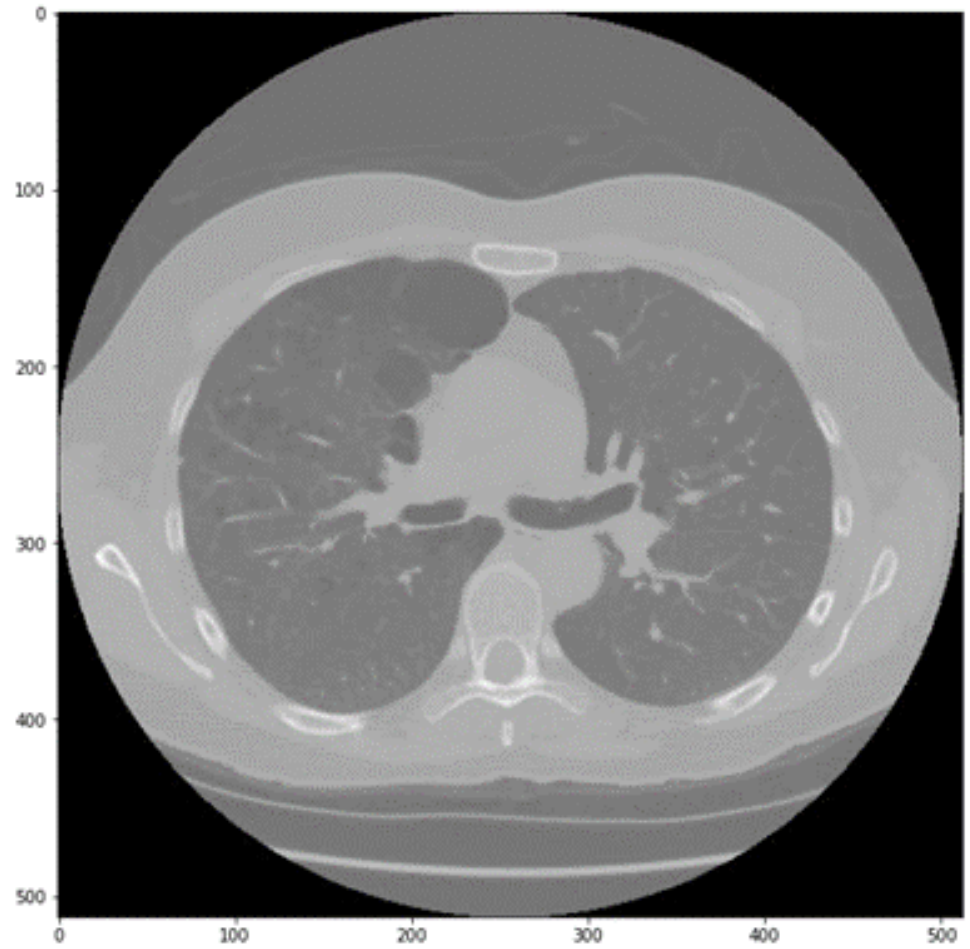


Şekil 13.9: Görüntü üzerinde yapılan işlemler

Görsel

```
segmente("/content/drive/MyDrive/sunum/kansersiz2.dcm")
```

Aşağıda Şekil 10'da Tahminin görselleştirilmiş hali sunulmuştur.




```
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 27483  100    19  100 27464      2   3747  0:00:09  0:00:07  0:00:02     4
```

Şekil 13.10: Tahminin görselleştirilmiş hali

Çıktı için oluşturulan txt dosyasını uygun formata dönüştürmek için kullanılan fonksiyon aşağıda görüldüğü şekilde oluşturulur.

```
def get_pred():
        try:
        data_dict = {}
        f = open("/content/output.txt","r")
        data = str(f.read()).strip().lstrip('{"predictions":[{'). ...
                        rstrip("}]}").split(",")
        for i in data:
```

```
        data_dict[i.split(":")[0]] = i.split(":")[1].strip(). ...
            rstrip('"')
    except: print("non cancer")
    return data_dict
    data_ = get_pred()
    data_
    w, h = int(float(data_[' "width"'])), ...
            int(float(data_[' "height"']))
    x, y = int(float(data_['x"'])), int(float(data_[' "y"']))
```

Kanserli bölge ekrana çizdirilir ve dörtgen içine alınır. Şekil 11'de Kanserli bölgenin dikdörtgen ile gösterilmesini ifade eden görsel paylaşılmıştır.

```
import matplotlib.pyplot as plt
import matplotlib.patches as patches
from PIL import Image
import numpy as np
img_ = np.array(Image.open('/content/image.jpg'),dtype = np.uint8)
plt.rcParams['figure.figsize'] = (20, 10)
fig, ax = plt.subplots(1)
ax.imshow(img_)
rect = patches.Rectangle((x - 27, y-27), w, h-10, linewidth = ...
    2,edgecolor = 'r', facecolor = "none", label = 'cancer')
ax.add_patch(rect)
plt.legend()
plt.show()
```

Şekil 13.11: Kanserli bölgenin dikdörtgen ile gösterilmesi

Yükseklik genişlik gibi değeler yardımı ile kanser hücresinin evrelendirilmesi işlemi.

```
    import numpy as np
    def evre():
    hypo_cm = (np.sqrt((((x+w)-x) * 512)^2 + ((y-(y-h)) * ...
            512)^2))/10
    label =""
    if hypo_cm <1:
    label = "T1a"
```

```
    elif hypo_cm<2 and hypo_cm>1:
    label = "T1b"
    elif hypo_cm<3 and hypo_cm>2:
    label = "T1c"
    elif hypo_cm<4 and hypo_cm>3:
    label = "T2a"
    elif hypo_cm<5 and hypo_cm>4:
    label = "T2b"
    elif hypo_cm<7 and hypo_cm>5:
    label = "T3"
    else:
    label = "T4"
    return label
```

Fonksiyonun çalıştırılması.

```
print(evre())
```

Elde edilen evre bilgisi ekranda; T4 olarak görülmüştür.

Çalışma kapsamında gerçekleştirilen üç aşamalı uygulamada ilk olarak model ve veri ile ilgili hazırlıklar bir sonraki adımda ise veri ön işleme yapılmış ve son olarak tahminleme işlemi gerçekleştirilmiştir. Aynı zamanda kişinin akciğer kanseri olup olmadığının tespite dayalı geliştirilen modelin tutarlılığı irdelenmiş doğruluk değeri %91 olarak belirlenmiştir. Sonuç olarak; yapay zeka temelli bu değerlendirme, akciğer kanseri teşhis, tespit ve evre değerlendirme doğruluğu ve verimliliğini artırmada gerçekleştirilecek diğer denemelerinde yolunu açabilir.

13.10 Kaynaklar

AKPINAR, H. (2017). DATA Veri Madenciliği Veri Analizi. İstanbul: Papatya Yayıncılık Eğitim A.Ş.

ALIMADAI, A., ARYAL, S., MANANDHAR, I., MUNREO, P. B., JOE, B., & CHENG, X. (2020). Artificial intelligence and machine learning to fight COVID-19. AI and Machine Learning for Understanding Biological Processes, 200-202.

ATALAY, M., & ÇELİK, E. (2017). BÜYÜK VERİ ANALİZİNDE YAPAY ZEKÂ VE MAKİNE ÖĞRENMESİ. Mehmet Akif Ersoy Üniversitesi Sosyal Bilimler Enstitüsü Dergisi, 155-172.

CEBECİ, H. İ. (2020, Mart 25). Kestirimci Modelleme Teknikleri. İş Zekası Ders Notları. Sakarya, Adapazarı, Türkiye: Sakarya Üniversitesi.

GLAUNER, P. O. (2015). Deep Convolutional Neural Networks for Smile Recognition. Deep Convolutional Neural Networks for Smile Recognition.

Rosebrock, A. (2021, Ekim 15). Convolutional Neural Networks (CNNs) and Layer Types. pyimagesearch: `https://www.pyimagesearch.com/2021/05/14/convolutional-neural-networks-cnns-and-layer-types/` adresinden alındı

Soo Young Lee, B. A. (2019). Steel Surface Defect Diagnostics Using Deep. ICMR 2019. Penang: Applied Sciences.

ŞEKER, A., DİRİ, B., & BALIK, H. H. (2017). Derin Öğrenme Yöntemleri ve Uygulamaları Hakkında Bir İnceleme. Gazi Mühendislik Bilimleri Dergisi, 47-64.

14. Sağlıkta Yapay Zeka

Sağlıkta Yapay Zeka ve Romatoloji Alanındaki Uygulanmaları

Duygu Temiz KARADAĞ*, Melih İNAL†, Furkan ZAMAN‡

***Dr. Öğr. Üyesi, Kocaeli Üniversitesi, Tıp Fakültesi, Romatoloji Anabilim Dalı, Kocaeli, Türkiye,** dr_dtemiz@hotmail.com

†**Prof. Dr., Kocaeli Üniversitesi, Enformatik Bölümü, Kocaeli, Türkiye,** melih.inal@gmail.com

‡**Uz. Dr., T.C. Sağlık Bakanlığı, İzmit Seka Devlet Hastanesi, İlk ve Acil Yardım Kliniği, Kocaeli, Türkiye,** furkan.zaman@gmail.com

14.1 Giriş

Günümüzde dijitalleşme ile birlikte başlayan artık Endüstri 4.0'dan Endüstri 5.0'a evrilen süreç; kısaca toplum odaklı insansız teknolojiler olarak özetlenmektedir. Elbette bu sürecin toplum odaklı olması sağlık alanı açısından tartışmasız üstlenilen bir olgu iken günümüzde sağlıkta aslında insansız teknolojiler yerine yardımcı hatta destekleyici süreçlerden söz edilebilir. Dijital dönüşüm süreçleri; büyük veri, nesnelerin interneti, robotik, biyosensörler ile giyilebilir teknolojiler, bulut bilişim gibi alanlar özellikle yapay zeka destekli uygulamalar ile ivmelenmektedir. Böylelikle bu ivmelenme sağlık alanında zaman, kaynak yönetimi ve insan kaynaklı hataların azalmasına önemli katkılar sağlamaktadır.

Elektronik Sağlık Kayıtları (ESK) için sağlık alanındaki büyük veriyi işleyebilmek; veriyi bilgiye dönüştürmek amacıyla kullanılan optimizasyon algoritmaları, veri madenciliği, veri görselleştirme, doğal dil işleme, istatistiksel analiz, matematiksel modeller ve yapay zeka gibi araçlar yer almaktadır. Dolayısı ile tıp bilişiminin somut neticesi sağlıkta dijital dönüşüm gerçek yaşam verisinden, büyük veriye ve makine öğrenmesinden yapay zeka platformuna doğru ilerlerken felsefî temellendirmeye de ihtiyaç duyulmaktadır (Bozbuğa, 2021). Bu çalışmada sağlık alanında kullanılan yapay zeka teknikleri özelinde makine öğrenmesi algoritmaları ve özellikle romatoloji alanındaki uygulamaları

ele alınacaktır.

14.2 Yapay Zeka ve Makine Öğrenmesi

Yapay zeka günümüzde neredeyse her alana uygulanmaktadır. Bunun yanında Yapay zeka yöntemlerinin uygulandığı alanlar yanında içerdikleri teknolojiler ile anılmaktadır. Bu teknolojiler ve alanlar aşağıdaki gibi özetlenebilir (Gülseçen, 2021):

- Uzman sistemler
- Karar Destek Sistemleri
- Özelleştirme / kişiselleştirme
- İtme / çekme teknolojileri
- Tavsiye sistemleri
- Görselleştirme
- Bilgi haritaları
- Akıllı ajanlar
- Otomatik taksonomi sistemleri
- Metin çözümlemesi – özetleme

Bu alanlarda kullanılan veriler üzerinde yapay zeka teknolojileri, makine öğrenmesi ve derin öğrenme algoritmalarını kullanarak çok daha isabetli kararlar veren, tıbbi klinik ve yönetimsel süreçleri kolaylaştıran ve insan hatalarını minimize eden sistemleri geliştirmek mümkün olmaktadır.

Makine öğrenmesi, eğitildiği verideki anlamlı örüntüleri yakalayarak, örüntünün bir veriden tekrar elde edilmesi noktasında eksik kalan kısımlarını da tamamlayarak öngörüde bulunmak veya belirsizlik durumunda en isabetli kararı üreten algoritmalar olarak tanımlanabilir.

Makine öğrenmesi sürecini en genel anlamda beş adımda özetlemek mümkündür (Kartal, 2021):

1. makineye verilecek görevin iyi şekilde tanımlanması
2. verinin makine öğrenmesi analizlerine hazırlanması
3. makine öğrenmesi algoritmalarının veri setine uygulanarak makine öğrenmesi analizleri yapılarak verinin modellenmesi
4. elde edilen performansın geliştirilmesi/iyileştirilmesi için makine öğrenme algoritmalarına ait parametrelerin ayarlanması (bagging, boosting ve blending gibi yöntemlerden faydalanılması)
5. makine öğrenmesi analiz sonuçlarının performans kriterlerine uygun şekilde yorumlanması

Makine öğrenmesinde kullanılan öğrenme yöntemleri eğiticili (supervised), eğiticisiz (unsupervised) ve pekiştirmeli (reinforcement) öğrenme olmak üzere üç başlıkta toplanabilir. Eğiticili öğrenmede eğitim öncesinde belirlenen hedef değere göre eğitim sürecinin sağlanmasıdır. Eğiticisiz öğrenmede ise böyle bir hedef değer olmadan modelin kendi yaklaşımlarını kendi belirleyerek genellemesi temeline dayanır. Pekiştirmeli öğrenmede ise verinin belirleyici anahtar özellikleri bir arama tablosu (lookup table) ile hedeflenen değerlerin eğiticili öğrenmede olduğu gibi eğitim sürecine katılması özelliğini ifade etmektedir (İnal, 2001).

En çok kullanılan makine öğrenmesi algoritmaları öğrenme yöntemlerine göre aşağıdaki gibi sıralanabilir:

- Eğiticili olanlar; doğrusal, logistik ve benzeri regresyonlar (linear, logistic regressions), karar ağaçları (decision trees), k-En Yakın Komşuluk Algoritması (k-Nearest Neighbour Algorithm), Naive Bayes Sınıflandırıcı (Naive Bayes Classifier), Yapay Sinir Ağları (Artificial Neural Networks ANN), destek vektör makinaları (support vector machines SVM), Convolutional neural network.

- Eğiticisiz olanlar; apriori algoritması, k-Ortalamalar Algoritması (k-Means Algorithm), kendi kendini organize eden temel içerenler analizi (Principal Component Analysis PCA).
- Benzerliği arttıran pekiştirmeli olanlar; Bagging yöntemi ile Random Forests Ağacı ve boosting ile AdaBoost (Uyarlamalı Arttırma Adaptive Boosting).

Bagging (torbalama) Bootstrap Örnekleme yöntemini kullanılarak oluşturulan veri kümeleriyle birden çok model oluşturmaktır. Bootstrap örneklemede, oluşturulan her eğitim seti, orijinal veri setinden rastgele alt örneklerden seçilir. Bagging, paralel bir topluluktur çünkü her model bağımsız olarak inşa edilmiştir. Öte yandan, boosting, her modelin bir önceki modelin yanlış sınıflandırmalarını düzeltmeye dayalı olarak oluşturulduğu sıralı bir topluluktur. Bagging çoğunlukla, her sınıflandırıcının paralel modellerin çoğunluğu tarafından belirlenen nihai bir sonucu elde etmek için oy kullandığı "basit oylamayı basic voting" içerir; boosting her sınıflandırıcının çoğunluk tarafından belirlenen nihai bir sonucu elde etmek için oy kullandığı "ağırlıklı oylamayı weighted voting" içerir. Ancak sıralı modeller, önceki modellerin yanlış sınıflandırılmış örneklerine daha fazla ağırlık verilerek oluşturulur (DATAQUEST, 2019).

14.2.1 Makine Öğrenmesinde Performans

Eğitim süreci tamamlanan bir algoritmanın performansının değerlendirilmesi için çeşitli performans değerlendirme yöntemleri ve ölçütleri kullanılmaktadır. Doğruluk (accuracy) ve hata oranı (error rate), duyarlılık (sensitivity), belirleyicilik (specificity) gibi kriterler performansın değerlendirilmesinde tercih edilmektedir. Tüm bu kriterler test verisinin tahmin ve gerçek değerleri üzerinden karışıklık (confusion) matrisine göre değerlendirilir. Tablo 1'de karışıklık (confusion) matris örneği bulunmaktadır.

Örneğin hastalık = var durumu Pozitif (hasta olma), hastalık = yok durumu ise Negatif (sağlıklı olma) olarak ele alındığında Tablo 1'de gösterilen karışıklık matrisi aşağıdaki şekilde ifade edilebilir:

- Gerçekte hasta olup, algoritmanın da hastalığı "var" şeklinde doğru tahmin ettiği örnekler doğru pozitif (true positive - TP).
- Gerçekte hasta olmayıp, algoritmanın hastalığı "var" şeklinde yanlış tahmin ettiği örnekler yanlış pozitif (false positive - FP).
- Gerçekte hasta olup, algoritmanın hastalığı "yok" şeklinde yanlış tahmin ettiği örnekler yanlış negatif (false negative - FN).
- Gerçekte hasta olmayıp, algoritmanın da hastalığı "yok" şeklinde doğru tahmin ettiği örnekler doğru negatif (true negative - TN).

Tablo 14.1: Karışıklık (Confusion) Matrisi

Karışıklık (Confusion)Matrisi		**Tahmin Değerleri**	
		Negatif	**Pozitif**
Gerçek (Hedef)Değerler	**Negatif**	Doğru Negatif (TN)	Yanlış Pozitif (FP)
	Pozitif	Yanlış Negatif (FN)	Doğru Pozitif (TP)

Bilgilerden yararlanılarak çeşitli performans kriterleri aşağıdaki gibi hesaplanabilmektedir:

$$Dogruluk(Accuracy)Accuracy = (TP+TN)/(TP+TN+FP+FN) \tag{14.1}$$

$$Belirleyici(Specificity) Specificity = TN/(TN+FP) \tag{14.2}$$

$$Kesinlik(Precision) Precision = TP/(TP+FP) \tag{14.3}$$

$$Hatirlama(Recall) Recall = TP/(TP+FN) \tag{14.4}$$

$$HataOrani = 1 - Dogruluk \tag{14.5}$$

Ayrıca model performansı hesaplanırken; ROC eğrisi (Receiver Operating Characteristics - ROC Curve), ROC Eğrisi altında kalan alan (Area Under the ROC Curve - AUC), ortalama karesel hata (mean squared error -mse) ya da kök ortalama karesel hata (root mean squared error-rmse) gibi ölçütlerden de faydalanılmaktadır.

14.3 Romatizmal Hastalıklarda Makine Öğrenmesi

Makine öğrenmesi (Machine Learning-ML) son yıllarda sağlık araştırmalarında sıklıkla başvurulan bir yöntemdir. Makine öğrenmesi ile büyük veri yığınlarını, çok boyutlu ve değişkenli analizlerle yorumlayabilmek mümkün görünmektedir. Makine öğrenmesi hasta verilerinde gizli olan karmaşık örüntüleri saptayarak geleneksel yöntemlerden daha etkin ve kesin öngörücü modeller oluşturabilmektedir.

Romatizmal hastalıklar oto-immün aracılı mekanizmalarla ortaya çıkan kronik ve ilerleyici hastalıkları içerir. Günümüzde Romatizmal hastalıklarla ilgili cevaplanmamış ya da yeterince aydınlatılamamış pek çok soru bulunmaktadır. Öncelikle hastalıkların sebepleri ve altında yatan mekanizmalar tam olarak anlaşılamamıştır. Romatizmal hastalıkların çok erken dönemlerinden itibaren tanınmasını ve tanı koyulmasını sağlayacak iyi tanımlanmış tanı kriterleri hala eksiktir. Hastalıkların tanısı bazı durumlarda kesin kriterlerden çok uzman görüşüne dayanmaktadır. Klinik bulguları hastalar arasında oldukça değişkenlik gösterdiği gibi hastalığa bağlı gelişen komplikasyonlar da oldukça heterojendir. Romatizmal hastalıkların kimlerde ortaya çıkacağı veya daha ağır seyredeceğine dair ipuçları olsa da tümüyle aydınlatılmış değildir. Hastaların tedavi hedeflerine ulaşması, hastalığın seyrinin belirlenmesi ve sonuçların erken ön görülmesindeki parametreler yetersizliğini korumaktadır. Romatizmal hastalıklarla ilgili klinik pratikten, bilimsel araştırmalardan, elektronik hasta kayıtlarından, görüntüleme ve laboratuvar arşivlerinden günümüze dek elde edilmiş oldukça büyük bir veri bulunmaktadır ancak bu verinin tamamı yeterince işlenerek bir sonuca ulaşmamıştır.

Son yıllarda romatizmal hastalıklarda makine öğrenmesi yöntemi ile yapılmış çalışmaların sayısı bu konudaki bilgiyi artırmıştır. Şimdiye kadar Romatoid artrit (RA), Sistemik lupus eritematozus (SLE), osteoartrit (OA), Juvenil idiopatik artrit (JIA), Spondiloartritler, Sistemik Sklerozis, miyozitlerle ilgili çalışmaların sonuçları yayınlanmıştır. Bu çalışmaların bazılarında makine öğrenmesi yöntemleri hastalığın alt tiplerini tanımlamak ve yeni tanı araçları geliştirmek için kullanılmıştır. Bir kısmında ise, geleneksel ve yeni nesil tedavilerde (biyolojik ilaçlar) yanıtı öngörmek amaçlanmıştır.

Tedaviye yanıtı öngörmede yeni belirteçler aramak için serum belirteçlerini, hastalık özelliklerini, hastaların demografik verilerini ve doku örneklerinin histopatolojisini makine öğrenmesi yöntemlerinde kullanan çalışmalar da vardır. Bu alanda araştırılan bir diğer konu da hastalık aktivitesini, mortaliteyi, morbiditeyi (hasta olma durumu), kronik hasarı ve hastalık progresyonunu öngörmek için etkin yapay zeka modelleri oluşturmaktır. Sağlık sisteminin önemli bir sorunu olan hastane yatışlarının iş ve maliyet yükünü azaltabilmek için hastanede yatış süresi ve tekrar yatışları öngörmeyi hedefleyen çalışmalar da diğer bir önemli araştırma konusudur.

14.3.1 Romatizmal Hastalıklarda Makine Öğrenmesi Uygulamaları

Romatoid artrit (RA)

Romatoid artrit eklem ve eklem dışı organlarda tutulumlar ile seyreden kronik, inflamatuvar, sistemik bir hastalıktır. Eklem tutulumu özellikle el ve ayak küçük eklemlerinde kalıcı hasar ve engellilikle sonuçlanan artrite yol açar. Kadınlarda erkeklerden daha sıktır. Sıklığı coğrafi dağılıma göre değişmekle birlikte Avrupa'da en sık romatizmal hastalık; ülkemizde ikinci en sık romatizmal hastalıktır. Son yıllarda yapılmış RA makine öğrenmesi çalışmaları özetlenmiştir (Jiang, 2021).

Tablo 14.2: Romatoid Artritte makine öğrenmesi çalışmaları

Amacı	ML Metodu	Örneklem	Giriş değişkenleri	Sonuçlar	Referans/yıl
Eklem hasarını tahmin	Convolutional neural network	108 RA hastası (klinik kohort)	Radyolojik görüntüleme	Accuracy: 49.3–65.4% (JSN), 70.6–74.1% (erozyon)	(Hirano, 2019)
Persistan ağrıyı tahmin	Unsupervised ML, supervised ML (random forests)	288 RA (klinik kohort)	Demografik ve klinik özellikler	Accuracy: 70%	(Lotsch,2020)
Hastalığı tanımlama	Decision tree, random forest	2588 RA (Health Informatics Research database)	Kodlar ve narrative concepts	PPV: 85.6%, specificity: 94.6%, sensitivity: 86.2%, accuracy: 92.29%	(Zhou, 2016)
Hastalığı tanımlama	Artificial neuralnetworks, decision tree	100 RA vs 100 sağlıklı kontrol (klinik kohort)	Serum sitokinler	Spesifite: 93%, sensitivite: 93%	(Heard, 2014)
Sinovyal örneklere göre RA sınıflaması	Support vector machine	Sinovyal doku örnekleri 123 RA ve 6 OA (klinik kohort)	Patolojik değişkenler	üç subtip için sırasıyla AUC: 0.71, 0.59, 0.88	(Orange, 2018)
Hastalık aktivitesi tahmini	Convolutional neural network	40 RA (klinik kohort)	Doppler ultrasound görüntüler	Accuracy: 86.9%, sensitivity: 87.5%, specificity: 86.4%	(Andersen, 2019)
MTX'e yanıt tahmini	L2-regularized logistic regression	85 RA (klinik kohort)	Genetik ve klinik değişkenler	AUC: 0.78	(Plant, 2019)
anti-TNF ilaca yanıt tahmini	Gaussian process regression	2706 RA (klinik kohort)	Genetik ve klinik değişkenler	AUC: 0.66, accuracy: 78%	(Guan, 2019)
Sinovyal imzaya göre tedavi yanıtı	Support vector machine	256 RA, 41 OA ve 36 sağlıklı control sinovyal doku (public datasets)	Genetik değişkenler	AUC: 0.92	(Kim, 2019)
Mortalite tahmini	Random forest	1741 RA (klinik kohort)	Demografik ve klinik özellikler	Accuracy: 76.7%	(Lezcano-Valverde, 2017)
Sonuç tahmini	Deep learning	820 RA (health care systems database)	Narrative concepts	AUC: 0.91	(Norgeot, 2019)

Romatoid artritte eklem hasarı hastalık ciddiyetini belirlemek ve tedaviye karar vermekte önemli bir etkendir. Günümüzde klinisyenler konvansiyonel grafilerde (X-Ray) eklemleri değerlendirerek ve bazı skorlamalar (sharp skoru gibi) yaparak eklem hasarını saptamaktadır. Skorlama yöntemlerinin konvansiyonel grafileri okumada oldukça fazla deneyim gerektirmesi ve uygulayıcı açısından zaman alıcı olması nedeniyle günlük pratikte sıklıkla kullanılmamaktadır. Daha çok bilimsel araştırmalar sırasında uygulanmaktadır. Hirano vd. (2019) yaptıkları bir çalışmada RA'da eklem destrüksiyonunu saptayabilmek için X-Ray grafilerini kullanarak derin öğrenme metodu ile bir model geliştirmeyi amaçlamışlardır (Hirano, 2019). Uyguladıkları model önce eklemi saptamak ve sonra eklemi değerlendirmek üzere iki aşamadan oluşmuştur. Eğitim/Doğrulama (validasyon) veri setinde proksimal interfalangeal, interfalangeal ve metakarpofalangeal eklemlerde eklem aralığının daralması (joint space narrowing (JSN)) ve erozyonu önce klinisyen tarafından skorlanmıştır. RA hastalarının 216 X-Ray görüntüsünden 186 tanesi eğitim/validasyonda ve 30 tanesi test için kullanılmıştır. 11160 imaj

eğitim ve validasyon için (3720 imaj eklemi saptamadaki eğitim için) değerlendirilmiştir. Çalışmanın sonunda makine öğrenmesi skorları ve klinisyenlerin skorları karşılaştırılmıştır. Model tüm eklemleri %95,3 sensitivite ile doğru saptadı. JSN için sensitivite %88,0–94.2 ve spesifite %52,0– 74,8; erozyon için sensitivite %34,8–42,4 ve spesifite %88,2–89,4 (JSN için yüksek tanı; erozyon için düşük tanı) Model ve klinisyenler arasında the percentage of exact agreement (PEA-accuracy) JSN için %49,3–65,4; erozyon için %70.6–74.1 arasında Model ve klinisyenler arasında the percentage of close agreement (PCA) eklem aralığında daralma (JSN) için %64,0–85,3; erozyon için % 84,3 arasında saptanmış.

Romatizmal hastalıklarda persistan (yerleşik) ağrı gelişme riski olan hastaları erken dönemde saptamak, tedaviye hızlı karar vermekte önemlidir. Lotsch vd., RA ile ilgili çok fazla miktardaki klinik veriyi makine öğrenmesi metoduyla kullanarak persiste eden ağrıyı öngörecek erken fonksiyonel parametreleri araştırmışlardır (Lotsch,2020). Çalışmanın amaçları; RA hastalarında tanıdan sonraki 60 aya kadar olan ağrının grup etkisini görmek, hastaları ağrı ilişkili gruplardan birine dahil etmeyi sağlayan klinik, laboratuvar ve demografik parametreleri tanımlamak (yaş, cinsiyet, ESH, CRP, şiş eklem, hassas eklem, hasta global değerlendirme, HAQ, DAS28-CRP, DAS28-ESH, antiCCP pozitifliği), tanı sırasında elde edilen parametrelerin bu açıdan ne kadar yeterli olduğunu görmek ve tanıdan 3 ay sonra hastaları persistan ağrı açısından değerlendirmenin gerekliliğini araştırmaktır. Çalışmada hastalar ağrının süre ve şiddetine göre ağrı ilişkili fenotipe dayanan 3 subgruba (düşük, orta, yüksek) ayrılmış. Çalışma sonucunda tanıdan sonra 3. aydan 5 yıla kadar persiste eden ağrısı diğerlerinden fazla olan subgrup (yüksek) ile ilgili 4 öngörücü parametre (hasta global değerlendirme, sağlık değerlendirme anketi, şiş eklem, hassas eklem) tanımlanmıştır.

Sağlık sisteminin birinci basamak elektronik kayıtları kullanılarak ikinci basamak elektronik kayıtlarındaki medikal duruma ilişkin RA tanısı konulabilir. Zhou vd. ikinci basamak lokal romatoloji klinik sistemi ile bağlantılı genel sağlık sisteminin tanı, tedavi ve işlemlerle ilgili 5 basamaklı kodlarını kullanarak birinci basamak kayıtlarından RA için hastalık fenotipleme algoritması geliştirmişlerdir (Zhou, 2016). Birinci basamak kayıtlarındaki 2.238.360 hastanın verilerinden 20.667'sinin ikinci basamak bir romatoloji kliniğine geçişte kullanıldığı görülmüş. 43.100 kod arasından 900 tanesinin RA tanısı konulan hastalarda, konulmayanlara göre daha fazla kullanıldığı keşfedilmiştir. Makine öğrenmesi modeli geliştirmek için ilgili klinik kodlar 37 gruba indirilmiştir. Sonuçta, birinci basamak sağlık sistemi kayıtlarından, RA tanısı için yüksek tahmin gücüne sahip olan 8 kod elde edilmiştir. Kod verilerine dayalı algoritma sonucu daha önce tanımlanmış 2 klinik algoritma [Quality and Outcomes Framework (QOF) ve Thomas vd.] ile kıyaslanabilir sonuçlar vermiştir.

Sistemik Lupus Eritematozus (SLE)

Sistemik Lupus Eritematozus (SLE), pek çok organı etkileyebilen kronik oto-immün bir hastalıktır. Hücre içi hedeflere karşı oto-antikor üretimi karakteristiktir. Özellikle anti-nükleer antikor (ANA) hastaların %95'inde pozitiftir. Kadınlarda erkeklere göre daha sıktır (10:1). Kadınlarda en sık doğurganlık çağında (15-45 yaş) görülmekle birlikte her yaşta ortaya çıkabilir. Son yıllarda yapılmış SLE makine öğrenmesi çalışmaları özetlenmiştir (Jiang, 2021).

Gürültülü Etiketleme (Noisy labeling) SLE tanımlamasını makine öğrenmesi ile otomatikleştirmek için bir "gümüş standart" oluşturmak üzere pozitif ve negatif kontrollerin otomatik olarak "gürültülü etiketlenmesini" sağlayan yöntemlerin uyarlanmış şeklini ifade etmektedir.

SLE'de hastalığa bağlı olarak organlarda kronik hasar hastaların yaklaşık %50'sinde görülebilir ve önlenmesi en önemli amaçlardan biridir. Kronik hasar geri dönüşümsüz ve en az 6 ay kalıcı olan bozukluk olarak tanımlanır. Ceccarelli vd. çalışmalarında SLE'ye bağlı gelişecek hasarı önceden öngörmek için makine öğrenmesi yöntemini kullanarak bir algoritma geliştirmeyi amaçlamışlardır.

Tablo 14.3: Sistemik Lupus Eritematozusta makine öğrenmesi çalışmaları

Amacı	**ML Metodu**	**Örneklem**	**Giriş değişkenleri**	**Sonuçlar**	**Referans/yıl**
Kronik hasarı tahmin	Recurrent Neural Networks	413 SLE (Sapienza Lupus Cohort)	Demografik ve klinik değişkenler	AUC: 0.77, sensitivite: 74%, sensitivite: 76%	(Ceccarelli, 2017)
Eroziv artriti tahmin etme	Forward wrapper, decision tree	120 SLE (Sapienza Lupus Cohort)	Klinik değişkenler ve görüntüleme	AUC: 0.806	(Ceccarelli, 2018)
Hastalık tanısı	Natural language processing, penalized logistic regression	400 SLE (Partners HealthCare Biobank)	Kodlar ve narrative concepts	PPV: 90%, spesifite: 97%, sensitivite: 64%	(Jorge, 2019)
Hastalık tanısı	Noisy labeling	1166 SLE (UCSF health system database)	Kodlar ve narrative concepts	AUC: 0.97, accuracy: 92%, precision: 85%	(Murray, 2019)
Subtipleri tanımlama	Unsupervised ML (clustering analysis)	161 SLE, 57 healthy controls (four public datasets)	Genetik ve klinik değişkenler	-	(Figgett, 2019)
Hastalık aktivitesi tahmini	K-nearest neighbors, random forest	156 SLE (Gene Expression Omnibus	Genetik ve klinik değişkenler	Accuracy: 70%	(Kegerreis, 2019)
Sonuç tahmini	Random forest	140 biopsy-proven lupus nephritis (clinical cohort)	Biyomarkırlar ve klinik değişkenler	sensitivite: > 70%, spesifite: > 70%	(Wolf, 2016)
Mortalite tahmini	Random forest	3839 SLE (OSHPD database)	Demografik ve klinik değişkenler	Accuracy: 88.1%	(Ward, 2006)
Hastane tekrar yatışlarını tahmin	Deep learning, artificial neural networks	9457 SLE (Cerner Health Facts EMR database)	Kodlar	Derin öğrenme için AUC: 0.70, yapay sinir ağları için 0.66	(Reddy, 2018)

Hasarı belirlemek için Systemic Lupus International Collaborating Clinics/American College of Rheumatology Damage Indexini (SDI) kullanmışlardır. Yapay sinir ağları için başlangıçta hasar olmayıp takipte hasar gelişen 38 hasta ve gelişmeyen 94 hasta 2 yıl boyunca en az 5 vizitte takip edilmiştir. Kronik hasarı öngörmede eğri altında kalan alanı (AUC) 0,77 olan bir matematiksel model geliştirmişler. 0,35 olan bir eşik değeri ile riskli hastaları en iyi sensitivite (0,74) ve spesifite (0,76) ile tanımlayabilmişlerdir.

Makine öğrenmesi modelleri SLE'de eklemlerde hasar bırakan eroziv artritin gelişimini öngörmede kullanılabilir. Ceccarelli vd. bir başka çalışmalarında ultrason ile tespit ettikleri kemik erozyonunun oluşumuna yol açan faktörleri hasar gelişmeden önce tespit edebilecek bir algoritma geliştirmeye çalışmıştır (Ceccarelli, 2018). Eklem tutulumu olan (artrit/artralji) ve olmayan SLE hastalarının ayrımı eklemlerin ultrason ile incelenmesi ile yapılmış. Algoritmaya klinik parametreler, laboratuvar parametreleri ve ilaçlar dahil edilmiş, Bunlar arasından anti-CarP, ACPA, artralji, Jaccoud's artropati, anti-Sm ve nörolojik bulguların yüksek tahmin edici değeri olduğu saptanmıştır.

Spondiloartritler (SpA)

Klasik olarak aksiyel iskeletin (sakroiliak eklem ve vertebra) inflamasyonu sonucu ortaya şiddetli bel ağrısı ile karakterize bir grup hastalıktır. Aksiyel iskelet dışında periferik eklemlerde artrit (diz, ayak bileklerinde ağrılı şişlik), kasların kemiklere yapışma yerlerinde ağrı ve şişlik (entezit), eklem dışı buğular (üveit) ve ortak genetik mutasyon (HLA B27) da saptanabilir. Günümüzde aksiyel tutulan ekleme göre aksiyel ve periferik olmak üzere iki SpA tanımlanmıştır. Aksiyel SpA ise, sakroiliak eklemlerin konvansiyonel grafide saptanabilen tutulumuna göre radyografik (r-axSpA) ve non-radyografik aksiyel SpA (nr-axSpA) olarak isimlendirilmiştir. Erkek/kadın oranı eski verilere göre 3:1 ve 9:1 arasında bildirilmiştir. Son yıllarda yapılmış SpA makine öğrenmesi çalışmaları özetlenmiştir (Jiang, 2021).

Rutin sağlık hizmetleri sırasında aksiyel spondiloartrit (AxSpA) hastalarına tanı konulmasında gecikme nedeniyle, daha az hastanın tedavi olduğu bilinmektedir. Walsh vd. çalışmalarında birinci basamak elektronik tıbbi kayıtlarındaki serbest metin dokümanlarını kullanarak ileri tetkik aşamasında AxSpA tanısı konulan hastaları tanımlayacak bir algoritma geliştirmeyi amaçlamıştır. Çalışmaları, AxSpA'yı tanımlamak için dil araştırması, kavramı temsil eden anlamlı terimlerin seçilmesi, seçilen sözcükleri içeren (snippet) metin bölümlerinin alınması ve bu metinlerin sınıflandırılması aşamalarını içermektedir. Algoritmanın eğitilmesinde doğal dil işleme (natural language processing-NLP) araçlarını kullanarak uzmanların sınıflamalarını tekrarlamışlardır. Eğitimde kullanılmayan bağımsız bir validasyon verisi anlamlı terimlerle bildirdikleri doğruluk oranları: "sacroiliitis" için %92; "spond*" için %91; "HLA-B27+" için %99 bulunmuştur. PPV (positive predictive value = Precision): "sacroiliitis" için %92.4; "spond*" için %90.5; "HLA-B27+" için %100 bulunmuştur (Walsh, 2017).

AxSpA'da radyolojik progresyon engelliliğe yol açan bir durumdur. Tedavi hedefi progresyonun önlenmesi ve kronik hasarın önüne geçilmesi olmalıdır. Joo vd. çalışmalarında makine öğrenmesi yöntemleri ile AxSpA'da radyolojik progresyonu öngörecek bir algoritma geliştirmeyi hedeflemiştir. İki bağımsız AxSpA grubundan elde edilen klinik ve laboratuvar veriler eğitim ve test için kullanılmıştır. 2 yıllık radyolojik progresyon modified Stoke Ankylosing Spondylitis Spine Score (mSASSS) ile hesaplanmış ve mSASSS'ta ≥ 2 unite artış radyolojik progresyon olarak tanımlanmıştır. Çalışmada yedi farklı makine öğrenmesi modeli ve algoritmalar uygulanarak test verisi için sonuçlar ROC ve precision recall (PR) eğirisi ile değerlendirilmiştir. Model için cinsiyet, tanı sırasındaki yaş, hastalık süresi, vücut kitle indeksi, HLA-B27, sigara, periferik artrit, entezit, üveit, psöriasis, inflamatuvar bağırsak hastalığı, eritrosit sedimentasyon hızı, C-reaktif protein (CRP),

Tablo 14.4: Spondiloartritlerde makine öğrenmesi çalışmaları

Amacı	**ML Metodu**	**Örneklem**	**Giriş değişkenleri**	**Sonuçlar**	**Referans/yıl**
Hastalığı tanımlamak	Natural language processing, supervised ML	Gaziler 2005-2015 arası (Veteran Health database)	Açıklamalı snippetler	PPV: 92.4% "sacroiliitis", 92.8% "spond*", ve 88.6% HLA-B27+	(Walsh, 2017)
Eklem erozyonunu tahmin	K-nearest neighbors, random forest	53 AS (clinical cohort)	BT - görüntüleri	AUC: 0.97, accuracy: 96%	(Joo, 2020)
anti-TNF ilaca yanıtı tahmin	Random forest, support vector machine, naive bayes, neural network	92 AS patients (clinical cohort)	Genetik değişkenler ve serum belirteçleri	Accuracy: 100% yanıtsızlar, 98% yanıtlılar	(Liu, 2019)
Psöriasislerde PsA gelişimini tahmin	Random forest, shrinkage discriminant analysis	3674 PsA ve 3566 PsC (5 GWAS datasets ve 1 Immunochip dataset)	Genetik değişkenler	AUC: 0.82, precision: > 90%, spesifite: 100%	(Patrick, 2018)
PsA'da kardiyovasküler risk tahmini	Support vector machine, k-nearest neighbors and random forest	155 PsA (klinik kohort)	Demografik ve klinik değişkenler	AUC: 0.8468	(Navarini vd, 2020)]

Ankylosing Spondylitis Disease Activity Score-CRP (ASDAS-CRP), mSASSS, sindezmofit, sakroiliak eklem tutulum derecesi, kalça eklemi tutulum derecesi, Z-skor (femur boyun, kalça, vertebra) ve anti-tümör nekrotizan faktör (anti-TNF) kullanımı gibi özellikler seçilmiştir. Eğitim veri setinde modeller %70-100 arasında bir dengelenmiş doğruluk göstermiştir. Test veri setinde dengelenmiş doğruluk tüm modeller için >%65 ve en yüksek random forest (RF) ile % 69,3 saptanmıştır. Test veri setinde ROC eğrisinde eğri altında kalan alan (AUC-ROC) > 0,78 olan iki model, generalized linear model (GLM) ve support vector machines (SVM) en iyi performansı göstermiştir (Joo, 2020).

Psöriatik artrit (PsA) AxSpA grubunun geleneksel bir alt tipi olarak bilinir. PsA hastalarında kardiyovasküler riskin arttığı bilinmektedir. Kardiyovasküler riski yüksek hastaları belirlemek önleyici stratejiler geliştirmek için önemlidir. Geleneksel kardiyovasküler risk algortimaları [Framingham Risk Score (FRS), Progetto CUORE ve Systematic Coronary Risk Evaluation (SCORE) risk skorları] PsA'daki heterojen değişkenlerin karmaşık ilişkileri nedeniyle kardiyovasküler riski tahmin etmede yetersiz kalmaktadır. Klasik risk öngörücüleri kardiyovasküler hastalık ile risk faktörünün lineer ilişki gösterdiği varsayımına dayanır. Navarini vd. çalışmalarında PsA'da geleneksel kardiyovasküler risk algoritmalarının kısıtlılıklarını aşarak daha iyi tahminlerde bulunacak makine öğrenmesi metodları geliştirmeyi amaçlamıştır (Reddy, 2018). Support Vector Machine, Random Forest (RF) ve K-Nearest Neighbor modellerini kullanarak yaş, cinsiyet, sistolik kan basıncı, total kolesterol, sigara ve hipertansiyon gibi risk faktörlerini değerlendirmişlerdir. Çalışmanın sonucunda PsA ile ilişkili CRP and Psoriatic Area Severity Index'in (PASI) kardiyovasküler risk açısından öngörücü faktörler arasında yer aldığını saptamışlardır.

14.4 Sonuç ve Öneriler

Son yıllarda Yapay Zeka tekniklerinin yaygın etkisi sonucu, sağlık alanında da yapay zekanın bir alt grubu olan makine öğrenmesi ile ilgili araştırmalar oldukça hız kazanmıştır. Makine öğrenmesi hastaların klinik, laboratuvar, tedavi ve takipleri sırasında elde edilmiş olan çok büyük sağlık verilerinin daha etkin şekilde analiz edilmesinde kullanıldığı görülmektedir. Özellikle Ankilozan spondilit ve Romatoid artrit gibi eklem tutulumları için veya Sistemik lupus eritematozus gibi iç organ tutulumları için görüntüleme yöntemlerinin çok sık kullanıldığı ve tanıdan tedavi takibine kadar henüz cevap bulunamamış pek çok soru bulunan romatolojik hastalıklarda makine öğrenmesi modelleri ile çözüm aranmaya başlandığı görülmektedir. Önümüzdeki yıllarda bu alandaki çalışmaların daha fazla hız kazanacağı kesindir.

14.5 Kaynaklar

1. Bozbuğa N, Gülseçen S, (2021) Tıp Bilişimi / Medical Informatics İstanbul Üniv. Yayını, E-ISBN: 978-605-07-0773-1 DOI: 10.26650/B/ET07.2021.003

Gülseçen S, Çevik E, (2021) Büyük Sağlık Verisi Ve Bilgi Yönetimi, Tıp Bilişimi, İstanbul Üniv. Yayını, DOI: 10.26650/B/ET07.2021.003.05

Kartal E, (2021) Makine Öğrenmesi ve Yapay Zeka, Tıp Bilişimi, İstanbul Üniv. Yayını, DOI: 10.26650/B/ET07.2021.003.18

İnal M, (2001) Yapay Sinir Ağları Tabanlı Konuşmacı Tanıma, Kocaeli Üniversitesi Fen Bilimleri Enstitüsü, Doktora Tezi

DATAQUEST, (2019) The 10 Best Machine Learning Algorithms for Data Science Beginners 26 Haziran 2019 tarihinde https://www.dataquest.io/blog/top-10-machine-learning-algorithms-for-beginners/ adresinden erişildi

Jiang M, Li Y, Jiang C, Zhao L, Zhang X, Lipsky PE. Machine Learning in Rheumatic Diseases. Clin Rev Allergy Immunol. 2021 Feb;60(1):96-110. doi: 10.1007/s12016-020-08805-6. PMID: 32681407

Hirano T, Nishide M, Nonaka N, Seita J, Ebina K, Sakurada K et al (2019) Development and validation of a deep-learning model for scoring of radiographic finger joint destruction in rheumatoid arthritis. Rheumatol Adv Pract 3(2):rkz047

Lotsch J, Alfredsson L, Lampa J (2020) Machine-learning-based knowledge discovery in rheumatoid arthritis-related registry data to identify predictors of persistent pain. Pain. 161(1):114–126

Zhou SM, Fernandez-Gutierrez F, Kennedy J, Cooksey R, Atkinson M, Denaxas S, Siebert S, Dixon WG, O'Neill TW, Choy E, Sudlow C, UK Biobank Follow-up and Outcomes Group, Brophy S (2016) Defining disease phenotypes in primary care electronic health records by a machine learning approach: a case study in identifying rheumatoid arthritis. PLoS One 11(5): e0154515

Heard BJ, Rosvold JM, Fritzler MJ, El-Gabalawy H, Wiley JP, Krawetz RJ (2014) A computational method to differentiate normal individuals, osteoarthritis and rheumatoid arthritis patients using serum biomarkers. J R Soc Interface 11(97):20140428

Orange DE, Agius P, DiCarlo EF, Robine N, Geiger H, Szymonifka J et al (2018) Identification of three rheumatoid arthritis disease subtypes by machine learning integration of synovial histologic features and RNA sequencing data. Arthritis Rheumatol. 70(5):690–701

Andersen JKH, Pedersen JS, Laursen MS, Holtz K, Grauslund J, Savarimuthu TR, Just SA (2019) Neural networks for automatic scoring of arthritis disease activity on ultrasound images. RMD Open 5(1):e000891

Plant D, Maciejewski M, Smith S, Nair N, Hyrich K, Ziemek D et al (2019) Profiling of gene expression biomarkers as a classifier of methotrexate nonresponse in patients with rheumatoid arthritis. Arthritis Rheumatol. 71(5):678–684

Guan Y, Zhang H, Quang D, Wang Z, Parker SCJ, Pappas DA, Kremer JM, Zhu F (2019) Machine learning to predict anti-tumor necrosis factor drug responses of rheumatoid arthritis patients by integrating clinical and genetic markers. Arthritis Rheumatol. 71(12):1987–1996

Kim KJ, Kim M, Adamopoulos IE, Tagkopoulos I (2019) Compendium of synovial signatures identifies pathologic characteristics for predicting treatment response in rheumatoid arthritis patients. Clin Immunol 202:1–10

Lezcano-Valverde JM, Salazar F, Leon L, Toledano E, Jover JA, Fernandez-Gutierrez B et al (2017) Development and validation of a multivariate predictive model for rheumatoid arthritis mortality using a machine learning approach. Sci Rep 7(1):10189

Norgeot B, Glicksberg BS, Trupin L, Lituiev D, Gianfrancesco M, Oskotsky B, Schmajuk G, Yazdany J, Butte AJ (2019) Assessment of a deep learning model based on electronic health record data to forecast clinical outcomes in patients with rheumatoid arthritis. JAMA Netw Open 2(3):e190606

Ceccarelli F, Sciandrone M, Perricone C, Galvan G, Morelli F, Vicente LN, Leccese I, Massaro L, Cipriano E, Spinelli FR, Alessandri C, Valesini G, Conti F (2017) Prediction of chronic damage in systemic lupus erythematosus by using machine learning models. PLoS One 12(3):e0174200

Ceccarelli F, Sciandrone M, Perricone C, Galvan G, Cipriano E, Galligari A, Levato T, Colasanti T, Massaro L, Natalucci F, Spinelli FR, Alessandri C, Valesini G, Conti F (2018) Biomarkers of erosive arthritis in systemic lupus erythematosus: application of machine learning models. PLoS One 13(12): e0207926

Jorge A, Castro VM, Barnado A, Gainer V, Hong C, Cai T, Cai T, Carroll R, Denny JC, Crofford L, Costenbader KH, Liao KP, Karlson EW, Feldman CH (2019) Identifying lupus patients in electronic

health records: development and validation of machine learning algorithms and application of rule-based algorithms. Semin Arthritis Rheumatol 49(1):84–90

Murray SG, Avati A, Schmajuk G, Yazdany J (2019) Automated and flexible identification of complex disease: building a model for systemic lupus erythematosus using noisy labeling. J Am Med Inform Assoc 26(1):61–65

Figgett WA, Monaghan K, Ng M, Alhamdoosh M, Maraskovsky E, Wilson NJ, Hoi AY, Morand EF, Mackay F (2019) Machine learning applied to whole-blood RNA-sequencing data uncovers distinct subsets of patients with systemic lupus erythematosus. Clin Transl Immunology 8(12):e01093

Kegerreis B, Catalina MD, Bachali P, Geraci NS, Labonte AC, Zeng C, Stearrett N, Crandall KA, Lipsky PE, Grammer AC (2019) Machine learning approaches to predict lupus disease activity from gene expression data. Sci Rep 9(1):9617

Wolf BJ, Spainhour JC, Arthur JM, JanechMG, PetriM, Oates JC (2016) Development of biomarker models to predict outcomes in lupus nephritis. Arthritis Rheumatol. 68(8):1955–1963

Ward MM, Pajevic S, Dreyfuss J, Malley JD (2006) Short-term prediction of mortality in patients with systemic lupus erythematosus: classification of outcomes using random forests. Arthritis Rheum 55(1):74–80

Reddy BK, Delen D (2018) Predicting hospital readmission for lupus patients: anRNN-LSTM-based deep-learningmethodology. Comput Biol Med 101:199–209

Walsh JA, Shao Y, Leng J, He T, Teng CC, Redd D, Treitler Zeng Q, Burningham Z, Clegg DO, Sauer BC (2017) Identifying axial spondyloarthritis in electronic medical records of US veterans. Arthritis Care Res (Hoboken) 69(9):1414–1420

Joo YB, Baek IW, Park YJ, Park KS, Kim KJ (2020) Machine learning-based prediction of radiographic progression in patients with axial spondyloarthritis. Clin Rheumatol 39(4):983–991

Liu J, Zhu Q, Han J, Zhang H, Li Y, Ma Y, He D, Gu J, Zhou X, Reveille JD, Jin L, Zou H, Ren S, Wang J (2019) IgG Galactosylation status combined with MYOM2-rs2294066 precisely predicts anti-TNF response in ankylosing spondylitis. Mol. Med. 25(1):25

Patrick MT, Stuart PE, Raja K, Gudjonsson JE, Tejasvi T, Yang J, Chandran V, Das S, Callis-Duffin K, Ellinghaus E, Enerbäck C, Esko T, Franke A, Kang HM, Krueger GG, Lim HW, Rahman P, Rosen CF, Weidinger S, Weichenthal M, Wen X, Voorhees JJ, Abecasis GR, Gladman DD, Nair RP, Elder JT, Tsoi LC (2018) Genetic signature to provide robust risk assessment of psoriatic arthritis development in psoriasis patients. Nat. Commun. 9(1): 4178

Navarini L, Sperti M, Currado D, Costa L, Deriu MA, Margiotta DPE et al (2020) A machine-learning approach to cardiovascular risk prediction in psoriatic arthritis. Rheumatology 1;59(7):1767-1769

15. Yazarlar Hakkında

Deniz DEMİRCİOĞLU DİREN, 2007 yılında lisans derecesini Sakarya Üniversitesi Endüstri Mühendisliği bölümünde, 2011 yılında yüksek lisans derecesini ve 2020 yılında doktora derecesini Sakarya Üniversitesi Fen Bilimler Enstitüsü Endüstri Mühendisliği Anabilim dalında tamamlamıştır. 2009 yılından bu yana Sakarya Üniversitesi Uzaktan Eğitim Araştırma ve Uygulama Merkezinde öğretim görevlisi olarak görevini sürdürmektedir. Yapay zeka, veri madenciliği, makine öğrenme, uzaktan eğitim, karar destek sistemleri, çok kriterli karar verme, kalite sistemleri konularında ulusal ve uluslararası birçok makale, bildiri, kitap bölümü çalışmaları ve hakemlikleri bulunmaktadır.

İletişim: ddemircioglu@sakarya.edu.tr

Mehmet Barış HORZUM, lisans derecesini Sakarya Üniversitesi, Eğitim Fakültesi, yüksek lisans derecesini Sakarya Üniversitesi Sosyal Bilimler Enstitüsü Bilgisayar ve Öğretim Teknolojileri Anabilim Dalında ve doktora derecesini de 2007 yılında Ankara Üniversitesi Eğitim Bilimleri Enstitüsünde Eğitim Teknolojisi alanında tamamlamıştır. 2007 yılından bu yana Sakarya Üniversitesi Eğitim Fakültesi Bilgisayar ve Öğretim Teknolojileri Eğitimi Bölümünde öğretim üyesidir. Uzaktan eğitim, eğitimde teknoloji kullanımı, öğretim tasarımı konularında ulusal ve uluslararası birçok proje, makale, bildiri, kitap çalışmaları ile hakemlik ve editörlükleri bulunmaktadır. Aynı zamanda 2019 yılından beri Sakarya Üniversitesi Uzaktan Eğitim Araştırma ve Uygulama Merkezi müdürlüğünü yürütmektedir.

İletişim: mhorzum@sakarya.edu.tr

Zerrin AYVAZ REİS, lisans derecesini İstanbul Üniversitesi, Astronomi ve Uzay Bilimleri, yüksek lisans derecesini İstanbul Üniversitesi Kantitatif Yöntemler ve doktora derecesini de İstanbul Üniversitesi Bilgisayar Bilimleri Mühendisliği alanlarında almıştır. Yazılım mühendisliği, veritabanı, bilgisayar destekli eğitim, uzaktan eğitim, engellilerin eğitimi, etik ve yapay zeka konularında ulusal ve uluslararası birçok makalesinin yanında, bildiri, kitap, kitap içi bölüm çalışmaları mevcuttur. Akademide Etik Derneği kurucu üyelerindendir. Halen İstanbul Üniversitesi-Cerrahpaşa Hasan Ali Yücel Eğitim Fakültesi öğretim üyesidir. Aynı zamanda İstanbul Üniversitesi Enformatik Bölümünde görev yapmaktadır.

İstanbul Üniversitesi-Cerrahpaşa, Hasan Ali Yücel Eğitim Fakültesi-BÖTE (Bilgisayar ve Öğretim Teknolojileri Eğitimi Bölümü)

İletişim: ayvazzer@iuc.edu.tr / zerrinareis@yahoo.com

Selçuk Kıran 1991'de Alman Lisesi'ni, 1998'de Boğaziçi Üniversitesi Elektrik – Elektronik Mühendisliği Bölümünü bitirmiştir. Aynı üniversitede Biyomedikal Mühendisliği bölümünde yüksek lisans tez aşamasındayken 2001 yılında Accenture isimli çok uluslu firmada Bilişim Danışmanlığı yapmak için yurt dışına gitmiş, bu firma adına 6 yıla yakın Avrupa ve ABD'de çeşitli projelerde çalışmıştır. Türkiye'ye döndükten bir süre sonra akademik kariyere geri dönmüş ve 2011 yılında Boğaziçi Üniversitesi Yönetim Bilişim Sistemleri Bölümü'nde yüksek lisansını, 2015'te de İstanbul Üniversitesi Sayısal Yöntemler'de doktorasını tamamlamıştır.

Boğaziçi Üniversitesi Yönetim Bilişim Sistemleri Bölümü'nde 3 yıl araştırma görevliliği ve Namık Kemal Üniversitesi'nde 5 yıl öğretim görevliliği tecrübelerinden sonra 2016 Ocak ayından beri Marmara Üniversitesi İşletme Fakültesi Yönetim Bilişim Sistemleri bölümünde öğretim üyesi olarak çalışmaktadır. Bir yandan bu fakültenin Dekan Yardımcılığı görevini yürütmekte diğer yandan da hem kendi bölümünde hem de Özyeğin, Bilgi, Yeditepe, Kültür, Beykoz gibi üniversitelerde bilişim üzerine dersler vermektedir.

İlkim Ecem EMRE, lise eğitimini Feyziye Mektepleri Vakfı Işık Lisesinde, lisans eğitimini ise 2015 yılında Marmara Üniversitesi İşletme Enformatiği Bölümü'nde tamamlamıştır. Yüksek lisans eğitimini 2017 yılında İstanbul Üniversitesi Enformatik Bölümü'nde "Veri Madenciliği ile Çocukluk Çağındaki Akut Romatizmal Ateşin Kalp Hastalığına Etkilerinin Analizi" başlıklı tez çalışması ile tamamlamıştır. Doktora eğitimini de yine İstanbul Üniversitesi Enformatik Bölümü'nde "Psikiyatrik Hastalıkların Makine Öğrenmesi Yöntemleri ile Ayrıştırılması" başlıklı tez çalışması ile 2021 yılında tamamlamıştır. 2017 yılından beri Marmara Üniversitesi İşletme Fakültesi Yönetim Bilişim Sistemleri Bölümü'nde araştırma görevlisi olarak çalışmaktadır. Veri madenciliği, makine öğrenmesi ve yapay zekâ uygulamaları ile ilgilenmektedir.

Kadir Burak OLGUN, İstanbul Ticaret Üniversitesi Bilgisayar Programcılığı önlisans programını tamamladıktan sonra lisans derecesini Yeditepe Üniversitesi Bilgisayar ve Öğretim Teknolojileri Eğitimi, yüksek lisans ve doktora derecesini ise İstanbul Üniversitesi Enformatik alanlarından almıştır. İlgi alanları erken yaşta programlama eğitimi, çevrimiçi öğrenme, eğitsel veri madenciliği ve makine öğrenmesidir. Akademik iş yaşamı öncesi öğrencilik hayatında yazılım sektöründe bir firmada yazılım geliştirme süreçleri üzerine çalışmıştır. Yeditepe Üniversitesi Eğitim Fakültesi'nde araştırma görevlisi ve öğretim görevlisi olarak bulunduktan sonra şu an Bursa Uludağ Üniversitesi Orhangazi Yeniköy Asil Çelik Meslek Yüksekokulu'nda öğretim görevlisi olarak görev yapmaktadır. Ayrıca yaklaşık 4 yıldır özel bir eğitim kurumunda lise öğrencileri için robotik ve kodlama eğitimleri düzenleyerek yarışmalar için danışmanlık yapmaktadır. 2014 yılında evlenmiş ve 4 yaşında bir kız babasıdır.

Dr. Sevinç GÜLSEÇEN, Lisans eğitimini 1984 yılında İstanbul Üniversitesi Fen Fakültesi'nde tamamladıktan sonra, İşletme Fakültesi Sayısal Yöntemler Anabilim Dalı'nda Yüksek Lisans ve "Yapay Sinir Ağları, İşletme Alanında Uygulanması ve Bir Örnek Çalışma" isimli tezle Doktora eğitimini yine burada tamamlamıştır. 2006 yılında Yönetim Bilişim Sistemleri alanında Doçent olmuş, 2012 yılında Profesör unvanı almıştır. ERASMUS ve ikili anlaşmalar kapsamında Litvanya, Fransa, İskoçya, Slovakya, Bulgaristan, İtalya ve Polonya'da araştırma ve ders verme faaliyetlerini gerçekleştirmiş, projelerde yer almıştır. Misafir araştırmacı olarak 1987-88 yıllarında YÖK-Dünya Bankası Projesi kapsamında DellTECH (Delaware, ABD), 2000 yılında NOAO (Arizona, ABD) ve 2008 ile 2011 yıllarında Ball State University'de (Indiana, ABD) bulunmuştur. 2016 yılında açılan İÜ İnsan-Bilgisayar Etkileşimi (İBE) Laboratuvarı'nın kurucusudur. SCI, SSCI ve alan indekslerine giren dergilerde yayınlanan 30 adet makalesi; 60 adet hakemli kongre/sempozyum yayını; kitap, kitap bölümleri ve kitap editörlüğü yaptığı 20 adet çalışması bulunmaktadır. İlgi alanları arasında Bilgisayar Programlama, Sistem Analizi ve Tasarımı, Bilgi Yönetimi, İnsan Bilgisayar Etkileşimi ile e-Öğrenme yer almakta; bu konularda lisans, yüksek lisans ve doktora düzeyinde dersler vermekte olup çok sayıda yüksek lisans ve doktora tezi yönetmiştir. Istanbul University Press kapsamında bilimsel hakemli dergi olarak yayınlanan ActaINFOLOGICA'nın baş editörüdür. Türkiye Bilişim Derneği ve Türkiye Mantık Derneği üyesidir. Dr. Sevinç GÜLSEÇEN evli olup İngilizce, Bulgarca ve Rusça dillerini bilmektedir. Enformatik Bölüm Başkanı ve FBE Enformatik Anabilim Dalı Başkanı olarak idari görevlerini İstanbul Üniversitesi'nde sürdürmektedir.

Hülya ÇİVAK, 2017 yılında Lisans derecesini, Karabük Üniversitesi Endüstri Mühendisliği bölümünü 2. olarak tamamlamıştır. 2017 yılında Bulanık Mantık yöntemi ile stok seviyeleri üzerine hazırladığı lisans tez bildirisi ISITES -2017 'de sunulmuştur. Aynı yıl Kardemir A.Ş.'de Malzeme ve Stok Yönetimi biriminde çalışmıştır. 2019 yılında Karabük Üniversitesi – Sakarya Üniversitesi ortak programı ile Fen Bilimleri Enstitüsü Endüstri Mühendisliği Bölümü'ne başlamıştır. Yüksek lisans döneminde bir demir çelik firmasında imalat ve stok alanlarının optimizasyonu üzerine simülasyon projesi yapmıştır. 2021 yılında Panel Data Analiz yöntemi ile COVID-19 verileri üzerine yaptığı çalışma Social Medicine and Health Management dergisinde yayımlanmıştır. 2021 yılında Doğuş Teknoloji'de RPA Developer olarak çalışmaya başlamıştır ve halen devam etmektedir.

İletişim: hulya.civak@d-teknoloji.com.tr, hulyacivak@hotmail.com

Duygu TEMİZ KARADAĞ, lisans derecesini Çukurova Üniversitesi Tıp Fakültesi, tıpta uzmanlık derecesini Kocaeli Üniversitesi Tıp Fakültesi İç Hastalıkları ve yan dal uzmanlık derecesini de Kocaeli Üniversitesi Tıp Fakültesi İç Hastalıkları -Romatoloji alanlarında almıştır.

Başlıca araştırma alanı "Sistemik Sklerozis" hastalığı olmakla birlikte hastalığın klinik bulguları ve tanı yöntemleri konularında ulusal-uluslararası makaleleri ve ulusal-uluslararası bildirileri mevcuttur. Türkiye Romatoloji Derneği ve European Scleroderma Trials and Research group (EUSTAR) üyesidir. Halen Kocaeli Üniversitesi Tıp Fakültesi İç Hastalıkları ABD, Romatoloji BD'da öğretim üyesi olarak görev yapmaktadır.

İletişim: duygu.temiz@kocaeli.edu.tr / dr_dtemiz@hotmail.com

Melih İNAL, lisans derecesini 1993'te Marmara Üniversitesi Teknik Eğitim Fakültesi Bilgisayar Öğretmenliği, yüksek lisans derecesini 1996'da Kocaeli Üniversitesi Elektronik ve Bilgisayar Eğitimi ve doktora derecesini 2001'de Kocaeli Üniversitesi Elektrik Eğitimi alanlarında almıştır. 2009'da Bilgisayar ve Kontrol Sistemleri alanında Doçent unvanını almış, 2014'te Enformatik Bölümünde Profesör unvanı verilmiştir. Programlama, veri yapıları, örüntü tanıma, kontrol sistemleri, bilgisayar destekli eğitim, uzaktan eğitim ve yapay zeka konularında ulusal ve uluslararası alanda birçok makalesi ve çalışmaları bulunmaktadır. Halen Kocaeli Üniversitesi Enformatik Bölüm Başkanlığı görevini yürütmektedir.

İletişim: melih.inal@kocaeli.edu.tr / melih.inal@gmail.com

Furkan ZAMAN, İstanbul Üniversitesi İstanbul Tıp Fakültesi mezunudur. Fakülteden sonra Kocaeli Derince Eğitim ve Araştırma Hastanesi Acil Tıp Kliniğinde uzmanlık eğitimini tamamlamıştır. Kocaeli Üniversitesi Mekatronik Mühendisliğine bir dönem öğrenci olarak devam ettikten sonra tıpta uzmanlık tezini yapay zeka üzerine yapmış ve bu alanda ilerlemektedir. Evli, bir çocuk babası ve Gebze Fatih Devlet Hastanesinde Acil Tıp Uzmanı olarak çalışmaktadır. Bisiklet sporu, amatör fotoğrafçılık, kampçılık, elektrik, yapay zeka ilgi alanlarıdır.

İletişim: furkan.zaman@sbu.edu.tr / furkan.zaman@gmail.com

Dr. Öğretim Üyesi Tijen Över Özçelik, İstanbul Teknik Üniversitesi Mühendislik Fakültesi Metalurji Mühendisliği Bölümünden 1987 yılında mezun olmuş ve Lucas Elektrik A. Ş.'de Kalite Kontrol Mühendisi olarak görev yapmıştır. 1999 yılı itibari ile Sakarya Üniversitesi'nde çalışmaya başlamış ve Metalurji Malzeme Mühendisliği'nde Yüksek Lisansını tamamlamıştır. "İhtiyaç Belirlemede Endüktif ROC Temelli bir Model" başlıklı doktora tezini 2006 yılında tamamlayarak Doktora derecesini Sakarya Üniversitesi Endüstri Mühendisliği'de alan Dr. Tijen Över Özçelik, Sakarya Üniversitesi Endüstri Mühendisliği'nde görev yapmaktadır. Mevcut araştırma alanları arasında; Yalın Üretim, Sistem Geliştirme, İleri İmalat Teknolojileri, Yenilik ve Teknoloji Yönetimi, İstatistik, Elektronik Devlet, Endüstri 4.0, Kadın ve Teknoloji sayılabilir. Över Özçelik, 2011'den bu yana Sakarya Üniversitesi Kadın Araştırma Merkezi üyesi olarak çalışmaktadır.

İletişim; tover@sakarya.edu.tr

1975 Sakarya doğumlu olan **Alpaslan Kibar**, lisans öğrenimini Çevre mühendisliği dalında 1996 – 2001 yılları arasında tamamladıktan sonra yüksek lisans derecesini Sakarya Üniversitesi Sosyal Bilimler Enstitüsü bünyesinde Üretim Yönetimi ve Pazarlama anabilim dalında 2004 yılında almıştır. Takip eden süreçte doktora derecesini yine aynı kurumda "Tedarik Zinciri Yönetimi Yazılımlarının semantikleştirilmesi" 2012 yılında elde etmiştir. 2001 – 2012 yılları arasından Enformatik Bölümü ve Uzaktan Eğitim Uygulama ve Araştırma Merkezlerinde araştırma görevlisi olarak çalışmıştır. Çalıştığı sürece bahse konu kurumlarda "Veri Tabanı Yöneticisi" sorumluluğunu da üstlene yazar, araştırma görevlisi olarak girdiği Sakarya Üniversitesi İşletme Fakültesinde hala Doktor öğretim üyesi görevini ifa etmektedir. Akademik hayatı boyunca uzaktan eğitim ve bilişim ile alakalı birçok projede aktif rol alan yazarın temel çalışma alanları veri tabanı yönetimi, semantik, doğal dil işleme ve makine öğrenmesidir.

İletişim; kibar@sakarya.edu.tr

Dr. Öğr. Üyesi Yunus ÖZEN, lisans ve yüksek lisans derecelerini Kocaeli Üniversitesi, doktora derecesini ise Sakarya Üniversitesi Bilgisayar Mühendisliği alanında almıştır.

Bilgisayar Ağları, Ağ Protokolleri, Nesnelerin İnterneti, İşletim Sistemleri, Yapay Zeka ve Derin Öğrenme konularında ulusal ve uluslararası makale ve bildiri çalışmaları mevcuttur.

Halen Yalova Üniversitesi Bilgisayar Mühendisliği Bölümü'nde öğretim üyesi olarak görev yapmaktadır.

Doç. Dr. Safiye TURGAY lisans derecesini, İstanbul Teknik Üniversitesi Endüstri Mühendisliği, yüksek lisans ve doktora derecesini Sakarya Üniversitesi Fen Bilimleri Enstitüsü Endüstri Ana Bilim dalından almı¸stır. Abant İzzet Baysal Üniversitesi Bilgisayar Programcılığı, Bilgisayar ve Öğretim Teknolojileri ve Eğitimi,˙I ¸sletme, Sakarya Üniversitesi Yönetim Bili¸sim Sistemleri bölümlerinde öğretim üyeliği yapan Safiye Turgay halen Sakarya Üniversitesi Mühendislik Fakültesi Endüstri Mühendisliği bölümünde öğretim üyesi olarak görev yapmaktadır. Çok etmenli sistemler, bulanık mantık, karar destek sistemleri, üretim sistemleri, çok kriterli karar verme teknikleri ve kaba kümeleme konularında çok sayıda yayını bulunmaktadır.

İletişim: safiyeturgay@yahoo.com, sencer@sakarya.edu.tr

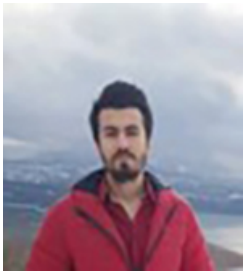
Suat ERDOĞAN, 2007 yılında başladığı İstanbul Kültür Üniversitesi Bilgisayar Teknolojisi ve Programlamadanmezun oldu. Sonrasında 2010 yılında başladığı Anadolu Üniversitesi İşletme Fakültesi'nden mezun oldu, ayrıca 2013 yılında başladığı Karabük Üniversitesi Bilgisayar Mühendisliği Bölümü'nü 2016 yılında bitirerek lisans eğitimini tamamladı. Yükseklisans eğitimini Sakarya Üniversitesi Mühendislik YönetimiBölümü'nde 2020yılında tamamlamıştır. Çalışma hayatında yazılım ve özellikle Java altyapısında olan birçok uygulama geliştirdi. 2019 yılından beri AgeSA Emeklilik ve Hayat şirketinde seniorsoftware developer olarak görev almaktadır.

Ramise Koçak, lisans derecesini Bilkent Üniversitesi – CTIS (Bilgisayar Tkn. Ve Bilişim Sistemleri), yüksek lisans derecesini Sakarya Üniversitesi – Yönetim Bilişim Sistemlerinden almıştır. 23 yıldır Otokar CAD Sistemlerinde çalışmaktadır.

Orhan Torkul, Lisans derecesini, Sakarya Üniversitesi Sakarya Mühendislik Fakültesi Endüstri Mühendisliği bölümünden 1982 yılında, Yüksek Lisans derecesini Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı'ndan 1987 yılında, doktora derecesini İngiltere Cranfield Teknoloji Enstitüsü'nden 1993 yılında almıştır. Sakarya Üniversitesi Mühendislik Fakültesi Endüstri Mühendisliği bölümünde 1993-1998 yılları arasında Yardımcı Doçent, 1998-2003 yılları arasında Doçent, 2003 yılından itibaren ise Profesör olarak görev yapmıştır. 1993-1995 yılları arasında Mühendislik Fakültesi Dekan Yardımcılığı, 1997-2011 yılları arasında Enformatik Bölüm Başkanlığı, 2005-2011 yılları arasında Uzaktan Eğitim Araştırma ve Uygulama Merkezi Müdürlüğü, 2011-2014 tarihleri arasında Endüstri Mühendisliği Bölüm Başkanı ve 2014-2018 yılları arasında ise Mühendislik Fakültesi Dekanı olarak görev yapmıştır. 2018-2020 yılları arasında Yalova Üniversitesi'nde Rektör Yardımcılığı görevi yapmıştır. İmalat Planlama ve Kontrol, Yönetim Bilişim Sistemleri, Uzaktan Eğitim, Yapay Zeka, Dijital Dönüşüm ve Endüstri 4.0 alanlarında araştırmalar ve çalışmalar yapmaktadır. Çok sayıda uluslararası hakemli dergide makale, ulusal ve uluslararası konferanslarda bildirileri ve kitap bölümleri bulunmaktadır. Birçok ulusal ve uluslar arası konferans ve sempozyum düzenleme komitelerinde yer almış ve davetli konuşmacı olarak çok sayıda konferans, bilimsel toplantı ve çalıştaylara katılmıştır. Prof. Dr. Orhan Torkul, Yönetim Bilişim Sistemleri, Bilişim Sistemleri Güvenlik ve Kontrolü ve İleri İmalat Planlama ve Kontrol Sistemleri konularında çeşitli dersler vermektedir.

Mehmet Emin BAL Sakarya Üniversitesi Bilişim Sistemleri Mühendisliği 4. sınıf öğrencisi. Data Science, Data Mining alanlarında çalışmalar yapmakta. Veri bilimi, web scraping, sql ve daha birçok teknolojik konuyla ilgili youtube kanalında ve github hesabında paylaşımlar yapmakta.

İletişim
`08muhammedeminbal@gmail.com`
`https://www.youtube.com/c/MuhammedEminBal`
`https://github.com/EminMuhammed`



Mehmet Fatih AKÇA 2021 yılında Sakarya Üniversitesi Yönetim Bilişim Sistemleri ve Anadolu Üniversitesi Web Tasarım ve Kodlama bölümünden mezun oldu. Veri analisti olarak çalışmakta. Son yıllarda makine öğrenmesi, yapay zeka, veri madenciliği konularıyla ilgilenmekte. Bu konularda çeşitli proje ve akademik çalışmalara dahil oldu.

İletişim; mehmetfatihakca0@gmail.com

2021 yılında "Mühendislikte Yapay Zeka ve Uygulamaları 4" kitabı ile bir seriye devam etmek istiyoruz. Umarız ki bu tür hizmetler yetiştirdiğimiz öğrencilerimiz için faydalı olur ve her yıl bu kitabın devamını çıkarabiliriz.

Yapay Zeka Yaz Okulu (YAZSUM) ilk olarak 2017 yılında yüz yüze 88 farklı üniversiteden 550'den fazla katılımcı ile Sakarya Üniversitesi ev sahipliğinde gerçekleştirilmiştir. 2018 yılında detaylı içeriklerle bir kez daha hizmet etme fırsatı bulduk. 2020 yılında ise COVID-19 sebebiyle çevrimiçi platformları kullanarak 3500'den fazla katılımcı ile gerçekleştirdik.

Eğitim kapsamında 96 saat eğitim verilmiştir. Bu rakam eğitmenlerimizi ve bizleri ziyadesiyle memnun etmiştir. 2021 yılında 6 yurtiçi 2 yurtdışı üniversite ortaklığıyla 3 günde 42 eğitimci ile 67 saatlik eğitim ile YAZSUM gerçekleştirildi.

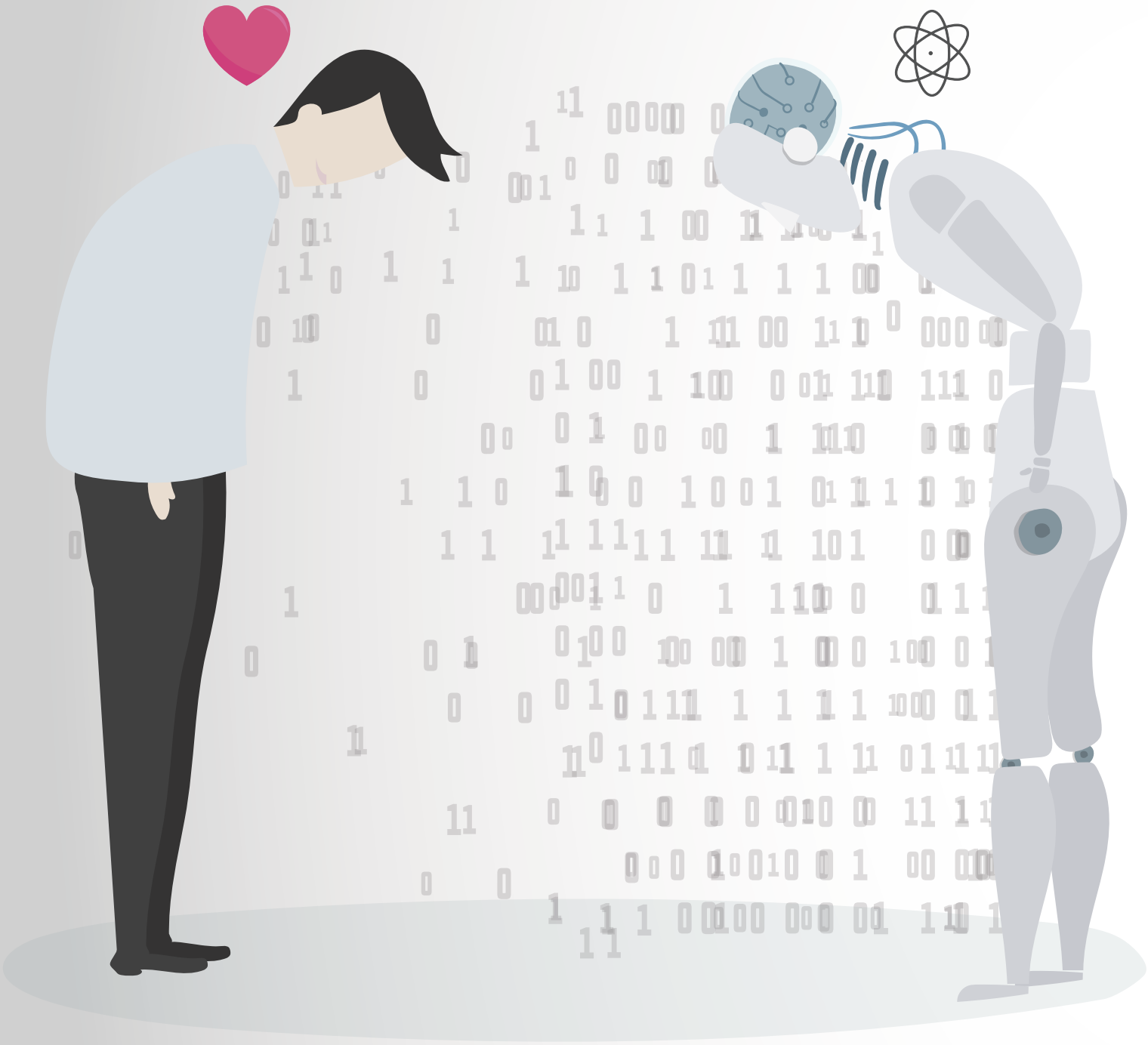
Pandemi sürecinde teknolojik alt yapılarının önemi bir kez daha ortaya çıkmıştır. Bu süre zarfında sürece hazırlıklı olan kurum ve devletler ilerleyişini hız kesmeden devam ettirmektedir.

Ülkemize ve kendimize ilim bakımından yatırım yapmak hayatımızın en önemli adımları olacaktır.

Elimizdeki bu kitap gerek teorik gerekse pratik uygulamalarla size yeni bir yol gösterici olmasını umuyoruz. Yapay zeka oldukça geniş bir konudur.

Zifiri karanlıkta her tarafı aydınlatamasakta önümüzü görecek kadar kendimize ve çevremize ışık tutmayı umuyoruz.

Işığınızın hiç kaybolmaması dileğiyle.



SAKARYA
ÜNİVERSİTESİ
YAYINLARI