T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

GELİŞTİRİLMİŞ Kİ-BİRLEŞTİRME ALGORİTMASI İLE AYRIKLAŞTIRILAN VERİNİN VERİ MADENCİLİĞİ YÖNTEMLERİ İLE SINIFLANDIRILMASI

DOKTORA TEZİ

Nuran PEKER

Enstitü Anabilim Dalı : ENDÜSTRİ MÜHENDİSLİĞİ

Tez Danışmanı : Prof. Dr. Cemalettin KUBAT

Ağustos 2021

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

GELİŞTİRİLMİŞ Kİ-BİRLEŞTİRME ALGORİTMASI İLE AYRIKLAŞTIRILAN VERİNİN VERİ MADENCİLİĞİ YÖNTEMLERİ İLE SINIFLANDIRILMASI

DOKTORA TEZİ

Nuran PEKER

Enstitü Anabilim Dalı : ENDÜSTRİ MÜHENDİSLİĞİ

Bu tez 27/08/2021 tarihinde aşağıdaki jüri tarafından oybirliği/oyçokluğu ile kabul edilmiştir.

Jüri Başkanı	**Üye**	**Üye**
Üye		**Üye**

BEYAN

Tez içindeki tüm verilerin akademik kurallar çerçevesinde tarafımdan elde edildiğini, görsel ve yazılı tüm bilgi ve sonuçların akademik ve etik kurallara uygun şekilde sunulduğunu, kullanılan verilerde herhangi bir tahrifat yapılmadığını, başkalarının eserlerinden yararlanılması durumunda bilimsel normlara uygun olarak atıfta bulunulduğunu, tezde yer alan verilerin bu üniversite veya başka bir üniversitede herhangi bir tez çalışmasında kullanılmadığını beyan ederim.

Nuran PEKER
06.06.2021

TEŞEKKÜR

Doktora eğitimim boyunca her konuda değerli bilgi ve deneyimlerinden yararlandığım, tezin tüm aşamalarında öğretici rehberliğini hissettiğim değerli danışman hocam Prof. Dr. Cemalettin KUBAT'a; sabrı, ilgisi ve nezaketi için sonsuz teşekkürlerimi sunarım.

Tez izleme süreci esnasında yapıcı ve yol gösterici fikirleriyle, tezde kullanılan yöntemlerin kapsamının genişletilmesinde katkıları olan tez izleme jürisindeki değerli hocalarım Doç.Dr. Safiye SENCER ve Dr. Öğr. Üyesi Seçkin ARI'ya teşekkürü borç bilirim.

Doktora eğitimi; uzun, yorucu ama hakkını verebilen insanlara çok şey katan önemli bir süreçtir. Ben de elimden geldiğince bu eğitimin hakkını vermeye çalışan bir insan olarak; bu sürecin bana kattıkları için, ister küçük ister büyük olsun payı olan herkese yürekten teşekkür ederim.

Son olarak sevgi, ilgi ve desteklerini hep hissettiğim ve bu süreçte bana duâlarıyla destek olan âileme içtenlikle teşekkür ederim.

İÇİNDEKİLER

SİMGELER VE KISALTMALAR LİSTESİ

dKiB	: Destek Ki-Birleştirme
DT	: Decision trees
EF	: Equal frequency
EW	: Equal width
FN	: False negative
FP	: False positive
KDD	: Knowledge Discovery from Database
KiB	: Ki-Birleştirme algoritması
kkKiB	: Karekök Ki-Birleştirme
KNN	: K-nearest neighbours
NB	: Naive Bayes
PGN	: Pyramidial Growing Network
sKiB	: Siluet Ki-Birleştirme
SVM	: Support Vector Machines
TN	: True negative
TP	: True positive
UKMD	: User Knowledge Modeling Dataset
wcss	: Within cluster sum of square

ŞEKİLLER LİSTESİ

TABLOLAR LİSTESİ

ÖZET

Anahtar kelimeler: Veri madenciliği, makine öğrenmesi, Ki-Birleştirme, ayrıklaştırma, sınıflandırma

Veri ayrıklaştırma, sürekli bir özniteliği mümkün olan en az bilgi kaybı ile sınırlı sayıdaki aralığa bölme işlemi olarak tanımlanabilir. Verinin bölündüğü her bir aralığa belirli bir değer atanır. Ayrıklaştırma, birçok veri madenciliği ve makine öğrenmesi algoritması açısından oldukça önem arz eden bir veri önişleme yaklaşımıdır. Çünkü bazı algoritmalar ya sürekli veri ile çalışamaz, ya da daha düşük performans ortaya koyar. Öte yandan ayrık verinin sürekli veriye göre anlaşılması ve yorumlanması daha kolaydır, ayrıca ayrık veri; tahmin, sınıflandırma ve birliktelik kuralları gibi farklı veri madenciliği problemlerinin çalışma süresini de kısaltmaktadır.

Bu çalışmada, Chi-kare istatistiğine dayalı ve literatürde oldukça sık kullanılan bir ayrıklaştırma yöntemi olan Ki-Birleştirme (KiB) algoritmasının performansını arttıran dört farklı ayrıklaştırma yöntemi önerilmektedir. Dirsek Ki-Birleştirme (dKiB), Siluet Ki-Birleştirme (sKiB), Karekök Ki-Birleştirme (kkKiB) ve 2-10'lu ayrıklaştırma olarak adlandırılan bu yöntemlerin, original KiB algoritması ile olan karşılaştırmalı sonuçları tartışılmaktadır. Bu yöntemlerden dKiB ve sKiB, k-ortalama algoritması kullanılarak, verinin bölüneceği en uygun küme sayısının bulunması esasına dayanmaktadır. Bu yöntemlerin uygulanmasında, dKiB için veri seti bütün olarak; sKiB için veri setinin herbir özniteliği ayrı ayrı ele alınmaktadır. Veri setinin herbir özniteliği için bulunan karekök değeri kkKiB algoritmasında, verinin bölüneceği küme sayısı olarak belirlenmektedir. 2-10'lu ayrıklaştırmada ise very, sırasıyla 2-10 arası kümeye bölünmekte ve sonuçlar KiB ile karşılaştırılmaktadır.

Yöntemlerin sınıflama başarısı; Karar Ağaçları (DT), Naive Bayes (NB), K-En yakın Komşular (KNN) ve Destek Vektör Makineleri (SVM) kullanılarak, 11 gerçek dünya veri seti üzerinde, katmanlı 10-kat çapraz doğrulama yöntemi ile ölçülmektedir. Elde edilen sonuçlar, önerilen dört yöntemin de orijinal KiB algoritması ile karşılaştırıldığında genelde daha iyi performans gösterdiğini ortaya koymaktadır.

CLASSIFICATION OF DATA THAT ARE DISCRETIZED WITH IMPROVED CHIMERGE ALGORITHM WITH DATA MINING METHODS

SUMMARY

Keywords: Data mining, machine learning, ChiMerge, discretization, classification

Data discretization can be defined as the process of dividing a continuous feature into a limited number of intervals with the least possible loss of information. Each interval in which the data is divided is assigned a specific value. Discretization is a very important data preprocessing approach for many data mining and machine learning algorithms. Because some algorithms either cannot work with continuous data or show lower performance. On the other hand, discrete data is easier to understand and interpret than continuous data, and it also decreases the running time of different data mining problems such as prediction, classification, and association rules.

In this study, four different discretization methods are proposed to increase the performance of the ChiMerge (KiB) algorithm, which is based on the Chi-square statistics and is a widely used discretization method in the literature. The comparative results of these methods, called Elbow ChiMerge (dKiB), Silhouette ChiMerge (sKiB), Square Root ChiMerge (kkKiB), and 2-10 discretization, with the original ChiMerge algorithm, are discussed. Among these methods, dKiB and sKiB are based on finding the most appropriate number of clusters into which the data will be divided by using the k-means algorithm. In the application of these methods, the data set for dKiB is considered as a whole; for sKiB, each attribute of the data set is handled separately. The square root value found for different values of each attribute of the data is determined in the kkKiB algorithm as the number of clusters into which the data will be divided. In 2-10 discretization, the data is divided into 2-10 clusters, respectively, and the results are compared with KiB.

Classification success of the methods is measured by stratified 10-fold cross-validation method on 11 real-world datasets using Decision Trees (DT), Naive Bayes (NB), K-Nearest Neighbors (KNN), and Support Vector Machines (SVM). The obtained results reveal that all four proposed methods generally perform better when compared to the original KiB algorithm.

BÖLÜM 1. GİRİŞ

İçinde yaşadığımız ve modern çağ olarak adlandırılan günümüz dünyasında, bilgi, sahip olunabilecek önemli bir güç kaynağı olarak görülmektedir. Bilginin ham maddesi olan veriyi saklamak, gerektiğinde kolayca erişmek ve bilgiye dönüştürmek maksadıyla işlemek, her geçen gün daha da önem kazanmaktadır. Verinin bilgiye dönüşüm sürecinde veri madenciliği ve makine öğrenmesi teknikleri sıklıkla kullanılır. Bu tekniklerden bazıları sürekli (continuous) verilerle çalışmadığı gibi, çalışanlardan bazıları da ayrık (discrete) veri ile daha iyi sonuçlar üretmektedir. Bu tez kapsamında Ki-Birleştirme (ChiMerge) (Kerber, 1992) algoritmasını temel alan, ancak bu algortimanın başarı oranını arttıran dört farklı yeni yöntem sunulmaktadır.

Ki-Birleştirme (KiB), ki-kare (chi-square) istatistiksel yaklaşımını kullanan bir veri ayrıklaştırma algoritmasıdır. Veri kümesindeki sınıf bilgisini kullanması bakımından denetimli, arama uzayını aşağıdan yukarıya taraması bakımından birleştirmeli bir algoritmadır. Algoritma, önceden belirlenen bir sonlandırma kriterine kadar, komşu aralıkların birleştirilmesi esasına dayanır. Algoritmada χ^2-eşik değeri (χ^2- threshold) kullanılır. Bu değer, istenen bir anlamlılık seviyesi ve serbestlik derecesine göre belirlenir. Anlamlılık seviyesinin doğru seçilmesi birleştirme işlemini doğrudan etkilediği için önemlidir. Gereğinden yüksek ya da düşük değerler, birleştirme sürecinin uzamasına; gerekenden daha fazla ya da daha az ayrıklaştırma aralığının oluşmasına yol açabilir. Serbestlik derecesi sınıf sayısının bir eksiğine karşılık gelir. Örneğin 3 sınıflı bir verinin serbestlik derecesi 2'dir (Kerber,1992).

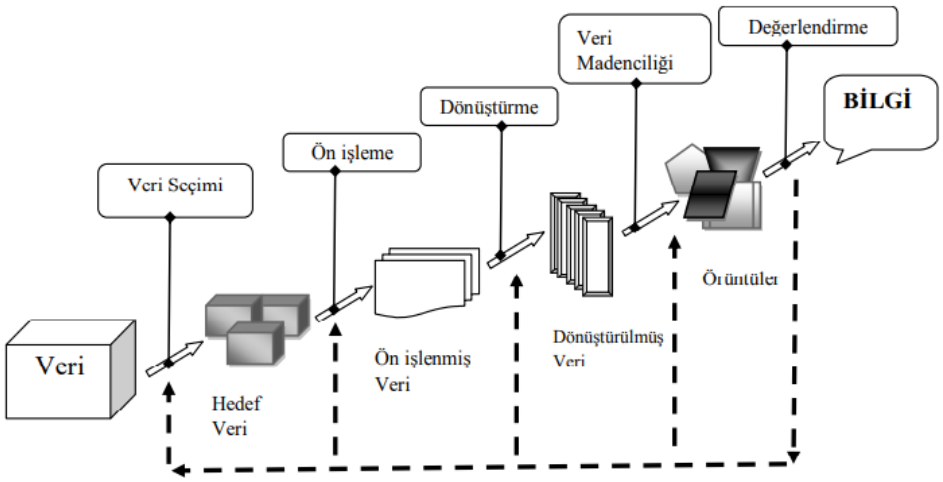
Tez kapsamında, KiB algoritması için aralık belirleme işleminde, anlamlılık seviyesi ve serbestlik derecesi yerine k-ortalama(k-means) algoritmasının yanısıra, karekök Ki-Birleştirme (kkKiB) ve 2-10'lu kümeleme yaklaşımları kullanılmaktadır. K-ortalama algoritması ile veri seti için uygun küme sayısı bulunduktan sonra, KiB algoritmasında birleştirme işlemi bu küme değerlerine göre yapılmaktadır. K-ortalama algoritmasında

bu k değerinin dışarıdan verilmesi gerekmektedir. K değeri, kümeleme başarımı açısından önemlidir. Bu değeri belirleme işleminde iki ayrı yöntem kullanılmaktadır. Bu yöntemler Dirsek (Elbow) ve Siluet (Silhouette) yöntemleridir. Bu yöntemler kullanılarak yapılan ayrıklaştırma işlemi, sırasıyla dirsek Ki-Birleştirme (dKiB) ve siluet Ki-Birleştirme (sKiB) olarak adlandırılmaktadır. Veri setlerinin 2 ilâ 10 arası kümeye ayrılacak şekilde ayrıklaştırılmasına dayanan çalışma, 2-10'lu kümeleme yaklaşımı olarak yer almaktadır.

1.1. Veri Madenciliği

Günümüzde gelişmiş teknolojiye bağlı olarak dijital aygıtların veri saklama kapasiteleri devasa boyutlara ulaşmıştır. Ayrıca sosyal ağlar ve internetin diğer amaçlar için yoğun bir şekilde kullanımı, giderek artan bir veri akışına yol açmıştır. İster kişisel ister kurumsal bazda olsun, internet üzerinden yapılan hemen her bağlantı, arkada bırakılan bir iz; diğer bir deyişle bir veridir. Veriyi bilgiye dönüştürmek amacıyla hareket eden karar vericiler açısından, ilk bakışta herhangi bir anlam ifade etmeyen bu ham verileri bilgiye dönüştürmek oldukça önem arz eder. Veri madenciliği teknikleri, bu ham verileri bilgiye dönüştürmek amacıyla sık kullanılan yöntemlerdendir.

Veri madenciliği, büyük veri yığınları arasından; anlamlı, işe yarar, ilginç örüntüleri elde etme süreci olarak tanımlanabilir. Veriler, veri tabanları, veri ambarları vb. ortamlarda tutulan yapılandırılmış veri olabileceği gibi; bilgi akışının sürekl, hızlı ve dinamik olarak gerçekleştiği internet ortamında bulunan ve 'büyük veri' olarak adlandırılan yapılandırılmamış veri de olabilir. Veri madenciliği veri tabanlarından bilgi keşif (knowledge discovery from databases-KDD) sürecinin aşamalarından biridir (Fayyad ve ark.,1996). Şekil 1.1., veri tabanlarından bilgi keşif sürecini ifade etmektedir.



Şekil 1.1. Veri tabanlarından bilgi keşfi (Savaş ve ark., 2012)

Veri tabanlarından bilgi keşif sürecindeki temel adımlar aşağıdaki gibi sıralanabilir (Han ve ark., 2011);

- Veri Seçimi: Bu adım, sorguya uygun örnek kümeyi elde etmek için veri kümelerinin birleştirildiği adımdır.
- Ön işleme: Birleşik veri kümesindeki hatalı verilerin çıkarıldığı, eksik ve gürültülü niteliklerin atıldığı adımdır.
- Dönüştürme: Birleşik veri kümesinden ilgisiz ve tekrarlı yapıların ayıklanıp verinin aynı forma sokulduğu adımdır. Bu aşama, uygulanacak veri madenciliği yönteminin, performansı açısından önemlidir.
- Veri Madenciliği: Veri madenciliği tekniklerinden biri kullanılarak veriden anlamlı bilginin çıkarıldığı aşamadır.
- Değerlendirme: Çıkarılan bilginin; yenilik, geçerlilik, basitlik ve yararlılık gibi kıstaslara göre değerlendirildiği aşamadır.
- Bilgi Sunumu: Elde edilen bilginin bilgi sunum ve görselleştirme araçlarıyla kullanıcıya sunulduğu aşamasıdır.

Veri madenciliği modellerini üç kategoriye ayırmak mümkündür: Sınıflama ve regresyon (classificaion and regression), kümeleme (clustering), birliktelik kuralları (association rules). Sınıflama ve regresyon, tahmin edici (predictive) model kapsamına girerken, kümeleme ve birliktelik kuralları tanımlayıcı (descriptive) model kapsamındadır (Özekes, 2003).

Sınıflama ve regresyon, veri sınıflarını tanımlamaya ve onları ayırt etmeye olanak sağlayacak birer model bulma işlemidir. Bu yöntemde, verilerin belirli bir kısmı eğitim amacıyla kullanılır. Yapılan analizler doğrultusunda, ortaya veriyi en iyi ifade edebilecek bir model çıkarılır. Oluşturulan bu model, hangi sınıfa ait olduğu belli olmayan verilerin, sınıf etiketini tayin etmek amacıyla kullanılır. Sınıflama, daha çok ayrık (discrete) yapılı değerler için; regresyon ise sürekli (continuous) yapılı değerler için kullanılır. Regresyon, istatistiksel bir yöntem olarak, mevcut veriye dayalı dağılımın tanımlanması için de kullanılır (Han ve ark., 2011). Sınıflama ve regresyon için, k-en yakın komşu (k-nearest neighbor), karar ağaçları (decision trees), naïve-bayes, yapay sinir ağları (artificial neural networks) ve lojistik regresyon (logistic regression) teknikleri sıklıkla kullanılır.

Kümeleme (Clustering): Kümeleme, bir veri topluluğundaki kayıtların benzer niteliklerine göre ayrı gruplara yeleştirilmesi işlemidir (Ramkumar ve Swami, 1998). Sınıflamada, gelen yeni veriler etiketi önceden belli olan gruplara yerleştirilir. Kümelemede ise başlangıçta mevcut verilerin herhangi bir sınıf etiketi yoktur. Veriler, sınıf içi benzerliğin en yüksek seviyeye çıkarıldığı; sınıflar arası benzerliğin ise en düşük seviyeye indirildiği bir ilke dikkate alınarak kümelere ayrılır. Yani, bir küme içindeki veriler olabildiğince birbirine benzerken, diğer kümelerdeki verilerle olabildiğince farklılık gösterir. Kümeleşme benzer olayları birlikte gruplayan sınıflar içinde gözlemlerin yapılmasını kolaylaştırır (Ramkumar ve Swami, 1998). Hiyerarşik kümeleme, k-en yakın komşu algoritması (k-NN), k-ortalama (k-means) algoritmaları, kümeleme için kullanılan önemli bazı yöntemlerdendir.

Birliktelik Kuralları (Association Rules): Bir veri topluluğu içindeki kayıtların birlikte gerçekleşme durumunu inceleyen bir veri madenciliği yaklaşımıdır. Eğitim tıp, e-ticaret, mühendislik, finans gibi birçok alanda kullanılır. Müşterilerin satın alma eğilimlerini ölçen ve birçok alanda uygulanan pazar sepeti analizi en çok bilinen birliktelik kural yaklaşımlarından biridir. Bu yaklaşımda, birlikte satın aldıkları ürünler üzerinden müşteri davranışları analiz edilir. Bu durum, pazarı konsolide eden

firmalara, ürünlerini daha etkin bir şekilde pazarlama olanağı tanır. Apriori, FP-büyüme (FP-growth), AIS, SETM, birliktelik kuralları için kullanılan bazı veri madenciliği yaklaşımlarındandır.

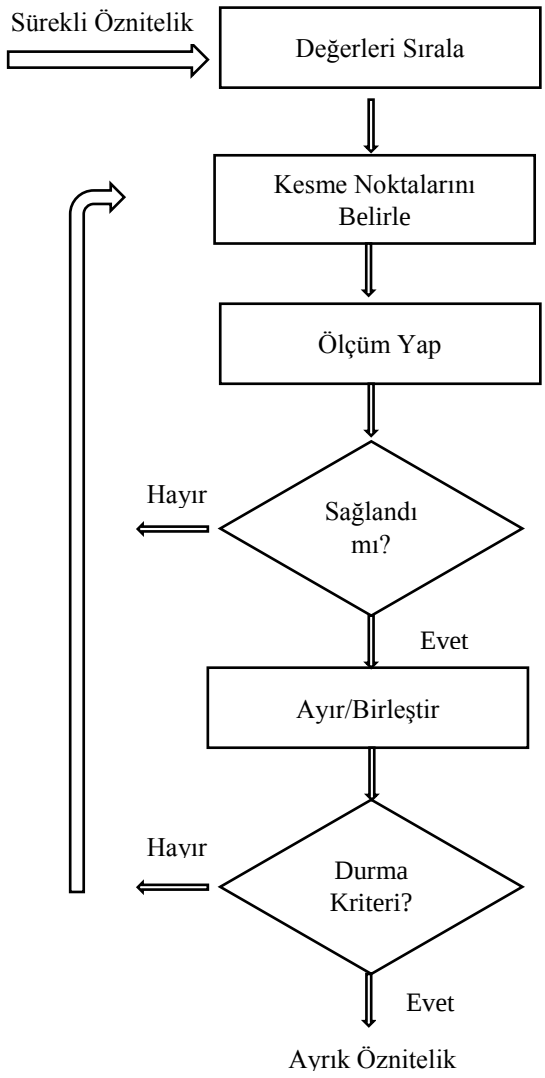
1.2. Veri Ayrıklaştırma

Veri ön işleme adımlarından biri olan veri ayrıklaştırma, sürekli sayısal değerleri belirli koşullar altında ayrık değerlere dönüştürme işlemidir. Örneğin ayrıklaştırılacak öznitelik kişilerin yaşına tekabül ediyorsa bu özniteliği, [0...12]: çocuk, [13...17]: ergen, [18...45]: yetişkin, [46...69]: orta yaş, [70...∞]: yaşlı şeklinde; ya da [0...12]: 0, [13...17]: 1, [18...45]: 2, [46...69]: 3, [70...∞]: 4 şeklinde ayrıklaştırmak mümkündür. Birçok veri madenciliği algoritmasının, sürekli nitelikler ayrıklaştırıldığında daha kullanışlı modeller ürettiği bilinmektedir. Örneğin Naive Bayes sınıflandırıcısı, olasılık tahmini üzerine kuruludur ve sürekli niteliklerin kullanımı genellikle kolay değildir. Çünkü çoğu zaman bir frekans aralığı için çok fazla farklı değer alır. Aynı durum karar ağaçları gibi kural çıkarımına dayalı sınıflama yöntemleri için de geçerlidir. Karar ağaçları nominal niteliklerin seçim işlemini gerçekleştirir ve sürekli olan değerleri doğrudan kullanamaz. Temel olarak, çok sayıda veri madenciliği yöntemi ve istatistiksel teknik sadece nominal değişkenlerden oluşan veri setlerine uygulanabilir. Bununla birlikte, gerçek veri setlerinin çok büyük bir kısmı sürekli değişkenler içerir. Bu sorunun bir çözümü, sürekli değişkenleri bir dizi alt aralığa ayırmak ve bu alt aralıkları birer kategori olarak ele almaktır.

Genel bir ayrıklaştırma süreci aşağıda sıralanan dört adımdan oluşur (Liu ve ark., 2002). Bu adımlara ait diyagram Şekil 1.2.'de görülmektedir.

- Sürekli bir özniteliğin değerlerini sıralama
- Önceden belirlenmiş bir değerlendirme ölçütüne dayalı bir kesme noktası seçme
- Yukarıdan aşağı (top-down) yöntemini kullanarak veriyi aralıklara bölme veya aşağıdan yukarıya (bottom-up) yöntemini kullanarak aralıkları birleştirme

- Ayrıklaştırma görevini ve kesme noktalarını bulma sürecini sona erdirmek için durma kriteri belirleme



Şekil 1.2. Veri ayrıklaştırma akış diyagramı

Literatürdeki mevcut ayrıklaştırma yöntemleri kullandıkları teknikler açısından birkaç sınıfa ayrılır: denetimli (supervised) veya denetimsiz (unsupervised), dinamik (dynamic) veya statik (static), küresel (global) veya yerel (local), parametrik (parametric) veya parametrik olmayan (non-parametric), bölmeli (splitting) veya birleştirmeli (merging). Bölmeli yöntemler, aynı zamanda yukarıdan aşağıya(top-down); birleştirmeli yöntemler ise aşağıdan yukarıya (bottom-up) olarak da adlandırılmaktadır.

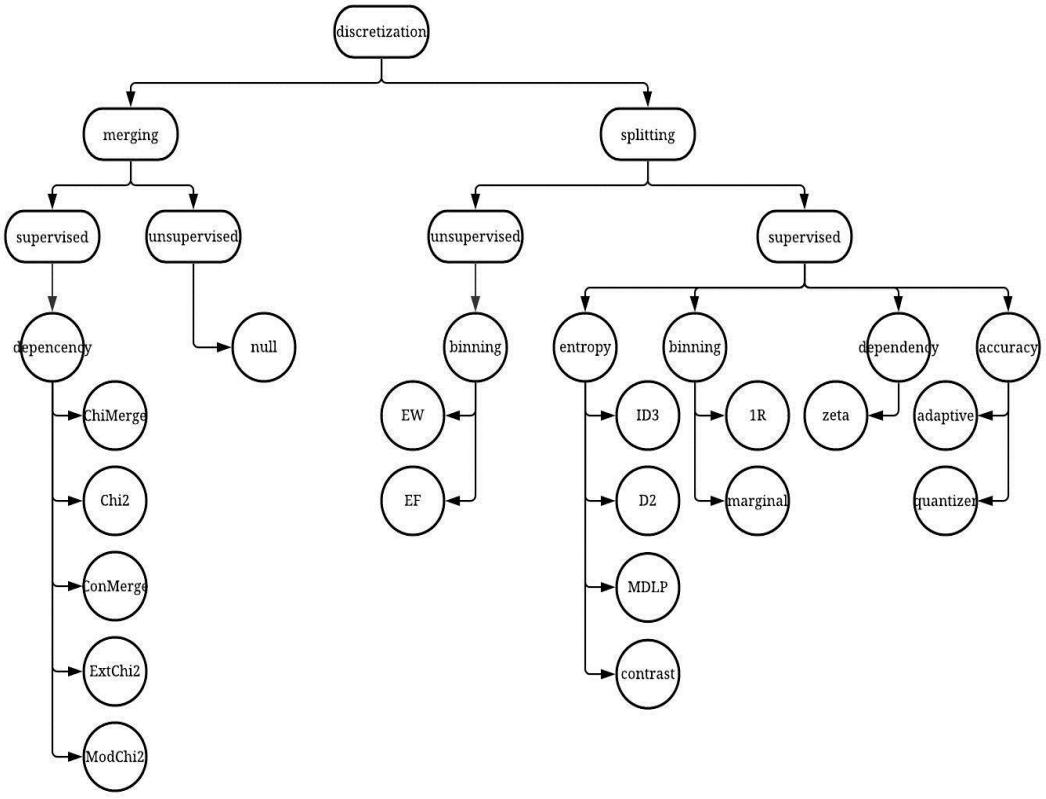
Veriler, sınıf bilgisine sahip olup olmamalarına bağlı olarak denetimli veya denetimsiz olarak kategoriz edilebilir. Denetimli ayrıklaştırmada verinin sahip olduğu sınıf bilgisi dikkate alınarak ayrıklaştırma yapılırken; denetimsiz ayrıklaştırmada sınıf bilgisi kullanılmaz. Denetimsiz ayrıklaştırmaya, eşit genişlikli (equal-width-EW) ve eşit frekanslı (equal-frequency-EF) yaklaşımları örnek olarak verilebilir. Denetimli yaklaşıma ise ChiMerge, Chi2 gibi istatistik tabanlı ve ID3 ve MDLP gibi entropi tabanlı birçok yöntem örnek olarak verilebilir. Denetimsiz yöntemlerde, sürekli aralıklar, kullanıcı tanımlı genişlik (değer aralığı) veya sıklığa (her aralıktaki örnek sayısı) göre alt aralıklara bölünür. Sürekli değerlerin dağılımının uniform olmadığı durumlarda bu durum iyi sonuçlar vermeyebilir. Ayrıca, aralıkları önemli ölçüde etkilediği için aykırı değerlere karşı duyarlıdır (Catlett, 1991). Bu eksikliğin üstesinden gelmek için, denetimli ayrıklaştırma yöntemleri, sınıf bilgisini kullanarak, daha uygun kesim noktaları bulmayı hedeflemiştir.

Ayrıklaştırma yöntemlerinin dinamik veya statik olması öğrenme modeli ile nasıl kullanıldıklarına göre değişir. Dinamik bir yöntem, bir sınıflandırıcı oluşturulurken aynı zamanda bütün sürekli değerleri eşzamanlı olarak ayrıklaştırır. Statik yaklaşımda ise ayrıklaştırma, sınıflandırma görevinden önce yapılır ve sürekli değerler birbirlerinden bağımsız olarak ayrıklaştırılır.

Diğer bir taksonomi olan yerel ve küresel yöntemler ise, verinin ne kadarının kullanıldığı ile ilişkilidir. Yerel bir yöntem, örnek uzayının yerelleştirilmiş bir bölgesinde (yani örneklerin bir alt kümesinde) ayrıklaştırma yaparken, küresel bir ayrıklaştırma yöntemi, tüm örnek alanda ayrıklaştırma yapar (Chmielewski ve Grzymala-Busse, 1994). Dolayısıyla, yerel bir yöntem, genellikle ayrıklaştırma için yalnızca bir örnek alanı bölgesinin kullanıldığı dinamik bir ayrıklaştırma yöntemidir.

Parametrik veya parametrik olmayan ayrıklaştırma yöntemleri, kullanıcı tanımlı parametre ihtiyacına göre sınıflandırılır. Parametrik yöntemler, ayrıklaştırma esnasında aralıkların belirlenebilmesi için kullanıcıdan bir ya da daha fazla parametre isterken, parametrik olamayan yöntemler, ayrıklaştırma işlemi için herhangi bir parametreye ihtiyaç duymaz.

Ayrıklaştırma yöntemleri ayrıca veri uzayını ele alış biçimi açısından yukarıdan aşağıya veya aşağıdan yukarıya olarak gruplandırılabilir. Yukarıdan aşağıya yöntemler, boş bir kesme noktası (cut-points) listesiyle başlar ve ayrıklaştırma ilerledikçe veriyi ayırarak (splitting) aralıkları listeye ekler. Aşağıdan yukarıya yöntemler de ise, kesme noktaları, bir özniteliğin tüm sürekli değerlerinin tam listesiyle başlar ve ayrıklaştırma ilerledikçe birleştirme (merging) yapılarak aralıkların bazıları kaldırılır. Şekil 1.3.'te ayrıklaştırma yöntemlerinin hiyerarşik şeması görülmektedir[1].



Şekil 1.3. Ayrıklaştırma algoritmalarının hiyerarşik şeması (Liu ve ark., 2002)

Sürekli bir değişkeni ayrıklaştırmanın çok sayıda yolu mevcuttur. Bu tez kapsamında KiB algoritmasının geliştirilmiş versiyonları olan dKiB, sKiB, kkKiB ve 2-10'lu kümeleme yöntemleri ele alınmaktadır.

[1] ExtChi2 ve ModChi2 yazar tarafından eklenmiştir

BÖLÜM 2. KAYNAK ARAŞTIRMASI

Birçok makine öğrenmesi ve veri madenciliği yaklaşımı için, veriyi ayrık veriye dönüştüştürürken, bunu en az bilgi kaybı ile yapmak, kullanılan yöntemlerin başarısı ve analiz sonuçlarının doğruluğu açısından önemlidir. Ayrıca sürekli veriye nispeten, ayrık verinin anlaşılmasının ve yorumlanmasının da daha kolay olduğu söylenebilir. Literatürde rastlanan çok sayıda ayrıklaştırma tekniğinden ve bu teknikler ile yapılmış farklı sınıflandırma yöntemlerinden söz etmek mümkündür. Tez kapsamında KiB (Kerber, 1992) yöntemini baz alan çalışmalardan söz edilmektedir.

Richeldi ve Rosotto (Richeldi ve Rosotto, 1995) StatDisc adını verdikleri ayrıklaştırma algoritmasını, EF, EW, D2 ve KiB algoritmaları ile karşılaştırmakta ve daha iyi performans elde etmektedirler.

Ropero ve arkadaşları (Ropero ve ark., 2018), KiB yöntemini, Bayes ağlarına uygulayıp, sonuçları eşit frekanslı, eşit genişlik ayrıklaştırma yöntemleri ile karşılaştırmıştır. Elde edilen sonuçlarda, KiB yöntemi daha iyi performans göstermiştir.

Minnie ve Srinivasan (Minnie ve Srinivasan, 2013), tam kan sayımına ait on iki bin adet kayıttan oluşan veriye, veri ön işleme adımında KiB algoritmasını uygulayarak veriyi ayrıklaştırmışlardır.

Mitov ve arkadaşları (Mitov ve ark., 2009) KiB, EW, EF ve MDL'den oluşan dört ayrıklaştırma yöntemini, Pyramidal Growing Network (PGN) sınıflandırma algoritmasına uygulamış ve KiB'in en başarılı sonucu elde ettiğini rapor etmişlerdir.

Rosati ve arkadaşları (Rosati ve ark., 2015), KiB algoritması ile prostat kanseri teşhisine yönelik bilgisayar destekli tanı sisteminden aldıkları verileri ayrıklaştırmış, ayrıklaştırılan veri ile tanıda önemli ölçüde başarı artışı elde etmişlerdir.

Cebeci ve Yıldız (Cebeci ve Yıldız, 2017), tavuk yumurtası kalite özelliğini değerlendirmek için veri üzerine Chi2, modified Chi2 ve KiB algoritmaları ile ayrıklaştırma yapmış ve en iyi sonucu KiB algoritmasının verdiğini bulmuşlardır.

Koçoğlu (Koçoğlu, 2012), aralarında KiB algoritmasının da bulunduğu sekiz farklı ayrıklaştırma yöntemi ile sürekli sayısal bir veriyi ayrıklaştırmış; çalışmada, KiB, CAIM, Chi2 algoritmaları aynı sonucu vermiştir.

Sriwanna ve arkadaşları (Sriwanna ve ark., 2017) GraphS ve GraphM olarak adlandırdıkları çizge tabanlı iki ayrıklaştırma yöntemini, aralarında KiB'in de bulunduğu 11 farklı ayrıklaştırma yöntemi ile aralık sayısı, zaman karmaşıklığı ve sınıflama doğruluğu açısından karşılaştırmakta ve daha iyi sonuçlar elde etmektedirler.

Lavangnananda, K., Chattanachot (Lavangnananda ve Chattanachot, 2017) KiB algoritması da dâhil sekiz farklı ayrıklaştırma algoritmasını, değişik sınıflama yöntemleri ile denemiş, en iyi sonucu veren yöntemleri Chi2, MDLP ve CACC olarak bulmuştur.

Ali ve Shahzad (Ali ve Shahzad, 2016), Ameva, Chi2, CADD, 1R, Bayes, CACC, DIBD, Ki-birleştirme algoritmalarıyla veri ayrıklaştırma yapmış, verileri birliktelik sınıflayıcıları olan CBA ve CBA2 üzerinde test etmiştir. Elde edilen bulgular sınıflama başarısının ortalama en iyi çıktığı durumun KiB yöntemine ait olduğunu ortaya koymuştur.

Thaiphan ve Phetkaew (Thaiphan ve Phetkaew, 2018), aralarında MDLP, CACC, KiB algoritmalarının bulunduğu on iki farklı ayrıklaştırma yöntemini karar ağaçlarına yönelik sınıflandırma için kullanmış, bulgular MDLP ve CACC'ın daha iyi performans gösterdiğine yönelik çıkmıştır.

Salama ve Hassanien (Salama ve Hassanien, 2011), Fuzzy C-Ortalama yönteminde Euclidean mesafesini hesaplamada KiB algoritmasını uygulamış ve geliştirilen yöntemin daha iyi sonuçlar elde ettiğini gözlemişlerdir.

Lehtinen ve arkadaşları (Lehtinen ve ark., 2012) KiB'in çevrimiçi ortamdaki verilere uygulandığı bir çalışma yapmışlardır. Çalışmada, KiB'in çevrimiçi gürültüye karşı dayanıklı olduğu ve uygulamada verimli sonuçlar ürettiğini gözlemlemişlerdir.

Tahraoui ve arkadaşları (Tahraoui ve ark., 2017), bitemporal ve multispektral uzaktan algılanan görüntüler kullanarak arazi değişiminin tespiti için denetimsiz bir yaklaşım sunmuş, eşikleme için KiB yöntemini kullanmışlardır. Bulgular yöntemin başarılı sonuç ürettiğini göstermiştir.

Bettinger (Bettinger, 2011), KiB ve Chi2 algortimalarına dayalı olarak geliştirdiği ChiD algortimasını 8 ayrı veri seti üzerinde denemiş, sonuçlar bu algortmanın 5 veri seti üzerinde daha iyi sonuçlar ürettiğini ortaya koymuştur.

Kalpana ve Mani (Kalpana ve Mani, 2017) yaptıkları çalışmada, Naive Bayes Sınıflandırıcısının performansının, bilgi kazanımı kullanılarak seçilen özenitelikler için, medyan temelli ayrıklaştırma ile KiB ayrıklaştırmayı karşılaştırmaktadırlar. Deneysel sonuçlar, medyan temelli ayrıklaştırmanın daha iyi sonuç verdiğini ortaya koymaktadır.

Wójciak ve Łupińska-Dubicka (Wójciak ve Łupińska-Dubicka, 2018), OneR, EF, EW ve KiB algoritmalarının sınıflama performansı üzerindeki etkilerini inceledikleri çalışmada denetimli ayrıklaştırma algortimaları olan OneR ve KiB'in daha iyi performans gösterdiklerini bulmaktadırlar.

Dai (Dai, 2004) çalışmasında, genetik algoritmalar ile yapılan ayrıklaştırma yöntemini, KiB algoritması ile karşılaştırmalı olarak sunmuş, sonuçlar genetik algoritma yönteminin daha iyi performans ortaya koyduğunu göstermiştir.

Tahan ve Asadi (Tahan ve Asadi, 2018), önerdikleri MEMOD adlı ayrıklaştırma yöntemini içlerinde KiB algoritmasının da bulunduğu 8 ayrı ayrılaştırma yöntemi ile, 20 veri seti üzerinde karşılaştırmaktadırlar. Önerdikleri yöntem, 2 veri seti dışındaki diğer veri setleri üzerinde KiB'den daha başarılı olmaktadır.

Drias ve arkadaşları (Drias ve ark., 2018), KiB algoritmasının hesaplama karmaşıklığını azaltmayı amaçlayan LR-SDiscr adlı ayrıklaştırma yöntemini literatürdeki yöntemlerle karşılaştırmakta ve algoritmanın daha iyi sonuçlar elde ettiğini rapor etmektedirler.

Vejkanchana ve Kucharoen (Vejkanchana ve Kucharoen, 2019), ikili sınıflandırma problemleri için, genetik algoritma kullanan GAbin adlı yöntemi KiB'i de içeren 11 farklı ayrıklaştırma yöntemi ile karşılaştırmaktadırlar. Tahmine dayalı değişkenleri analiz etmek için kullandıkları ana istatistiklerden biri olan bilgi değeri (information value) açısından en başarılı sonucu KiB üretmektedir.

Li ve arkadaşları (Li ve ark., 2020) berrak hücreli böbrek karsinmu hastalığına ait verilerin kümeleme (binning) işleminde KiB algortimasını kullanmaktadırlar. Sonuçlar yöntemin tahminleme başarısını arttırdığını göstermektedir.

Chen ve arkadaşları (Chen ve ark., 2020) takviyeli öğrenme tabanlı genetik algoritmaya dayalı olarak geliştirdikleri ayrıklaştırma algoritmasını KİB'i de içeren 7 ayrı yöntemle, kesme noktalarının sayısı ve hata payı açısından karşılaştırmaktadırlar. Sonuçlar, önerdikleri yöntemin daha başarılı olduğunu ortaya koymaktadır.

Literatürde ki-kare istatistiğini kullanan ve bazıları KiB algoritmasının geliştirilmiş versiyonu olan algoritmalar da bulunmaktadır. ChiSplit (Bertier ve Bouroche, 1981), bölme noktasına bitişik iki alt aralığa uygulanan ki-kare ölçütünü maksimize ederek bir aralığın en iyi bölünmesini hedefler, eğer iki alt aralık istatistiksel olarak önemli ölçüde farklıysa aralık bölünür. ChiSplit durma kuralı, iki alt aralığın çok benzer olması durumunda bölünmeyi reddetmek için kullanıcı tanımlı ki-kare eşiğine dayanır.

Chi2 algoritması (Liu ve Setiono, 1995), tanımlı bir tutarsızlık oranını durdurma kriteri olarak tanımlayıp, ayrıklaştırma sürecini otomatikleştiren bir KiB uzantısıdır. Algortimada istatistiksel anlamlılık seviyesi, tutarsızlık kriteri karşılandığı sürece gittikçe daha yakın aralıkları birleştirmek için değişmeye devam eder.

ConMerge (Wang ve Liu, 1998), Chi2'ye çok benzeyen bir algoritmadır. Her seferinde bir özniteliği dikkate almak yerine, tüm sürekli özniteliklerin aralıkları arasından en düşük χ^2 değerini seçerek ayrıklaştırma işlemi yapar. ModifiedChi2 (Tay ve Shen, 2002), Chi2'nin bir modifikasyonudur. Tutarsızlığı kontrol etmek için kaba küme teorisine dayalı bir tutarlılık oranı kullanır. ExtendedChi2 (Su ve Hsu, 2005), Chi2'deki tutarsızlık kontrolünün en düşük üst sınırla değiştirildiği bir Chi2 sürümüdür. Rectified Chi2 algoritması (Qu ve ark.,2008), ModifiedChi2 ve ExtendedChi2 algoritmalarındaki sorunları gidermeyi amaçlamaktadır. ChiD algortiması (Bettinger, 2011), KiB ve Chi2'ye dayalı başka bir algoritmadır. ChiD algoritması ayrıklaştırılan sürekli değişkenin bitişik aralıklarından hesaplanan bir χ^2 istatistiğinin önem seviyesinin *logworth*'unu maksimize etmeye dayalıdır.

Khiops (Boulle, 2004), ki-kare kriterini tüm örnek uzayında global bir şekilde optimize eder ve herhangi bir durdurma kriteri gerektirmez. Khiops yöntemi ayrıklaştırmayı temel tek değer aralıklarından başlatır. Bitişik aralıklar arasındaki tüm birleştirmeleri değerlendirir ve tüm aralıklara uygulanan ki-kare kriterine göre en iyi olanı seçer. Yöntem, ayrıklaştırılmış öznitelik ile sınıf özniteliği arasındaki ki-kare bağımsızlık testi ile ilgili güven düzeyi artık azalmadığında, birleştirme işlemini otomatik olarak durdurur.

Ameva (Gonzalez-Abril ve ark., 2009), Ki-kare istatistiklerine dayalı bir acil durum katsayısını en üst düzeye çıkaran ve potansiyel olarak minimum sayıda kesikli aralık oluşturan bir ayrıklaştırma yöntemidir.

Zou ve arkadaşları (Zou ve ark., 2013) Chi2 algoritmasına dayalı, aralık benzerliği tekniğini kullanan yeni bir algoritma önermişlerdir. Algoritmada *koşul* ve *küçük hareket* olarak adlandırdıkları iki parametre kullanarak ayrıklaştırma ve bitişik iki

aralığın tutarsızlığı sürecinde dengeyi sağlamayı hedeflemişlerdir. Sonuçlar, algoritmanın, gerçek değer özniteliklerini makul bir şekilde ayırabildiğini ve aynı zamanda Chi2 algoritmasının korelasyon sorununa çözüm getirdiğini ortaya koymaktadır. Denetimli alan tabanlı ki-kare ayrıklaştırma algoritması (supervised area-based chi-square discretization algorithm) (Sang ve ark., 2014), yüksek boyutlu verileri ayrıklaştırmak için önerilen bir yöntemdir.

BÖLÜM 3. MATERYAL VE YÖNTEM

3.1. Veri Setleri

Tez kapsamında kullanılan bütün veri setleri UCI Machine Learning Repository (Asuncion ve Newman, 2007)'den alınmaktadır. Bu veri setleri, 150 ilâ 9752 arası örnekten oluşan Iris, Bupa, Heart, WholeSale, Ecoli,Vertebral, Yeast, User Knowledge Modeling Dataset(UKMD), Occupancy, Wilt ve Wifi (Wifi Localization) veri setleridir. Böylece küçük, orta ve büyük ölçekli ve çeşitli sayıda sınıf etiketine sahip 11 farklı alandan alınan veri seti üzerinde ayrıklaştırma yönteminin sınıflama başarım sonucunu incelemek, mümkün olmaktadır. Ecoli, Yeast ve Occupancy veri setlerinde *id* ve *date* olarak ifade edilen öznitelikler, ayrıklaştırma işlemine uygun olmadığı için, adı geçen veri setlerinden çıkarılmaktadır. Nominal öznitelikler zaten ayrık değerleri ifade ettiği için ayrıklaştırmaya tabi olmamaktadır.

Tablo 3.1. Kullanılan veri setleri

Veri seti	#Öznitelik	#Örnek	#Sürekli öznitelik	#Nominal öznitelik	#Sınıf
Iris	4	150	4	0	3
Bupa	6	345	5	1	2
Heart(Statlog)	13	270	5	8	2
WholeSale	7	440	6	1	2
Ecoli	8	336	7	1(id)	8
Vertebral	6	310	6	0	2
Yeast	9	1484	8	1(id)	10
UKMD	5	403	5	0	4
Occupancy	6	9752	5	1(date)	2
Wilt	5	4839	5	0	2
Wifi	7	2000	7	0	4

3.2. Ki-Birleştirme Algoritması

Ki-kare (chi-square), bir özniteliğin değerleri ile sınıf arasındaki ilişki üzerine anlamlılık testi yapan istatistiksel bir ölçüdür. Bir özniteliğin iki bitişik aralığının sınıftan bağımsız olup olmadığını test eder. Eğer bağımsız iseler, birleştirilmeleri; aksi takdirde ayrı bırakılmaları gerekir.

KiB, ki-kare (chi-square) istatisitksel yaklaşımını kullanan bir veri ayrıklaştırma algoritmasıdır. Veri kümesindeki sınıf bilgisini kullanması bakımından denetimli, arama uzayını aşağıdan yukarıya taraması bakımından birleştirmeli bir algoritmadır. Algoritma, önceden belirlenen bir sonlandırma kriterine kadar, komşu aralıkların birleştirilmesi esasına dayanır. Algoritmada χ^2-eşik değeri (χ^2- threshold) kullanılır. Bu değer, istenen bir anlamlılık seviyesi ve serbestlik derecesine göre belirlenir. Serbestlik derecesi sınıf sayısının bir eksiğine karşılık gelir (Kerber, 1992). Örneğin 3 sınıflı bir verinin serbestlik derecesi 2 ve %95 düzeyinde (α=0,05) χ^2-eşik değeri 5,991'dir. Bu eşiğin anlamı, sınıf ve özniteliğin bağımsız olduğu durumlarda, hesaplanan χ^2 değerinin 5,991'den az olma olasılığının %95 olduğudur; bu nedenle, eşiği aşan χ^2 değerleri, öznitelik ve sınıfın bağımsız olmadığı anlamına gelir. Anlamlılık seviyesinin doğru seçilmesi birleştirme işlemini doğrudan etkilediği için önemlidir. Gereğinden yüksek ya da düşük değerler, birleştirme sürecinin uzamasına; gerekenden daha fazla ya da daha az ayrıklaştırma aralığının oluşmasına yol açabilir. Aralıkları birleştirmede kullanılan χ^2 değeri, bir formül kullanılarak belirlenir. Algoritmanın çalışma mantığı temel olarak aşağıdaki gibi ifade edilebilir:

- İstenen nitelik için veriler artan düzende sıralanır
- Bu veriler için sınıfları dikkate alınarak aralıklar belirlenir
- Belirlenen aralık çiftleri için χ^2 değeri hesaplanır
- Her aralık çifti için hesaplanan χ^2 değeri, belirlenen eşik değerinin altında ise bu aralık çifti birleştirilir
- Birleştirme sonrası oluşan yeni aralıklar için yeniden hesaplanan χ^2 değerleri eşik değerinin üzerine çıkıncaya kadar, birleştirme işlemine devam edilir.

Yukarıda bahsi geçen χ^2 değeri aşağıdaki formüle göre hesaplanır:

$$\chi 2 = \sum_{i=1}^{m} \sum_{j=1}^{k} \frac{\left(A_{ij} - E_{ij}\right)^2}{E_{ij}} \tag{3.1}$$

Burada m = 2 (karşılaştırılan iki aralık), k ise sınıfların sayısını ifade etmektedir.

$$R_i = \sum_{j=1}^{k} A_{ij} \tag{3.2}$$

$$C_j = \sum_{i=1}^{m} A_{ij} \tag{3.3}$$

$$E_{ij} = \frac{R_i C_j}{N} \tag{3.4}$$

$$N = \sum_{j=1}^{k} C_j \tag{3.5}$$

(3.1) ilâ (3.5) arasındaki formüllerde ifade edilen değerlerin açıklaması aşağıda sıralanmaktadır:

R_i: i. aralıktaki örnek sayısı
A_{ij}: j. sınıfın i. aralığındaki örnek sayısı
C_j: j. sınıfın örnek sayısı
E_{ij}: A_{ij}'nin beklenen frekansı
N: Örneklerin toplam sayısı

Literatürde KiB algoritmasının çalışma mantığını anlatmak için Tablo 3.2.'nin kullanıldığı örnekten sıklıkla bahsedilir.

Tablo 3.2. KiB örnek tablosu

Örnek	Öznitelik Değeri	Sınıf
1	1	1
2	3	2
3	7	1
4	8	1
5	9	1
6	11	2
7	23	2
8	37	1
9	39	2
10	45	1
11	46	1
12	59	1

Yukarıdaki tablo için aralıklar (0-2), (2-5), (5-7,5), (7,5-8,5), (8,5-10), (10-17), (17-30), (30-38), (38-42), (42-45,5), (45,5-52), (52-60) şeklinde belirlenmiş olsun.

Tablo 3.3. Birinci örnek aralık için değerler

Aralık	Sınıf=1	Sınıf=2	Toplam
7,5-8,5	$A_{11}=1$	$A_{12}=0$	$R_1=1$
8,5-10	$A_{21}=1$	$A_{22}=0$	$R_2=1$
Toplam	$C_1=2$	$C_2=0$	N=2

Tablo 3.3. Birinci örnek aralık için değerleri göstermektedir. Tablodaki değerlere göre aşağıdaki hesaplamalar yapılır.

$E_{11} = 2/2 = 1$, $E_{12} = 0/2 \approx 0{,}1$; $E_{21} = 2/2 = 1$, $E_{22} = 0/2 \approx 0{,}1$

$\chi^2 = (1 - 1)^2/\ 1 + (0 - 0{,}1)^2/\ 0{,}1 + (1 - 1)^2/\ 1 + (0 - 0{,}1)^2/\ 0{,}1 = 0{,}2$ şeklinde bulunur. Serbestlik değeri $d = 1$ ve $\chi^2 = 0{,}2 < 2{,}706$ olduğundan bu iki aralık için birleştirme yapılır. Tablo 3.4. ikinci örnek aralık için değerleri göstermektedir.

Tablo 3.4. İkinci örnek aralık için değerler

Aralık	Sınıf=1	Sınıf=2	Toplam
0-10	A_{11}=4	A_{12}=1	R_1=5
10-42	A_{21}=1	A_{22}=3	R_2=4
Toplam	C_1=5	C_2=4	N=9

$E_{11} = 2,78$, $E_{12} = 2,22$; $E_{21} = 2,22$, $E_{22} = 1,78$

$\chi^2 = 2,72 > 2,706$ olduğu için birleştirme yapılmaz. En son ayrıklaştırma işleminden sonra veri, (0-10), (10-42), (42-60) aralıklarına bölünmüş olur.

3.3. K-ortalama (K-means)

Kümeleme, bir veri setinde benzer özellikler gösteren verilerin gruplanması olarak tanımlanabilir. Kümeleme, yapılarına göre bölümlemeli, hiyerarşik, yoğunluk-tabanlı, grid-tabanlı gibi farklı dallara ayrılır. K-ortalama kümeleme algortiması, bölümlemeli yöntemlerden biridir. Aynı küme içinde benzerlikler fazla, kümeler arası benzerlikler az olacak şekilde verinin, uzaklık hesabına dayalı olarak bölündüğü bir algoritmadır.

K-ortalama (MacQueen, 1967) algoritmasında öncelikle her kümenin merkez noktasını veya ortalamasını temsil etmek üzere k adet nokta (veri) rastgele seçilir. Kalan diğer noktalar, kümelerin ortalama değerlerine olan uzaklıkları dikkate alınarak en benzer oldukları kümelere dahil edilir. Daha sonra, her bir kümenin ortalama değeri hesaplanarak yeni küme merkezleri belirlenir ve tekrar noktaların merkeze uzaklıklarına bakılır. Herhangi bir değişim olmayıncaya kadar algoritma iteratif bir şekilde devam eder. Durum kararlı hale geldiğinde herbir veri noktası, en benzer olduğu kümeye atanmış olur. Algoritma temel olarak aşağıdaki 4 aşamadan oluşur:

- Küme merkezlerinin belirlenmesi
- Merkez dışındaki verilerin mesafelerine göre kümelendirilmesi
- Yapılan kümelendirmeye göre yeni merkezlerin belirlenmesi
- Veri kararlı hale gelene kadar 2. ve 3. adımların tekrarlanması

Çalışma kapsamında k sayısının belirlenmesi aşamasında k-ortalama algoritması üzerinde dirsek ve siluet yöntemleri kullanılmıştır. Dirsek yönteminde sınıf etiketi çıkarılan veri setlerinin bütün sayısal nitelikleri kullanılarak veri seti kümelenmiş ve optimal küme sayısı Ki-birleştirme algoritmasına parametre olarak verilmiştir. Siluet yönteminde ise veri setlerinin her bir özniteliği ayrı ayrı birer sütun vektörü olarak ele alınmış ve kümelenmiştir. Her bir öznitelik için bulunan optimum küme değeri o öznitelik için Ki-birleştirme algoritmasına parametre olarak verilmiştir.

3.3.1. Dirsek (Elbow) yöntemi

Dirsek yöntemi (Thorndike, 1953), bir veri kümesinde uygun sayıda kümenin bulunmasına yardımcı olmak için tasarlanan sezgisel bir yöntemdir. Yöntem küme analizindeki tutarlılığın yorumlanması ve doğrulanması amacıyla kullanılır. Bölümlemeli kümeleme yöntemlerinin arkasındaki temel fikir kümelerin toplam küme içi değişiminin en aza indirgenmesi olarak tanımlanabilir. Bu amaçla dirsek yönteminde, küme-içi kareler toplamının(within cluster sum of square-wcss) mümkün olduğunca küçük olması hedeflenir.

$S = (S_1,...,S_k)$: Küme sayısı

$\mu = (\mu_1,..., \mu_k)$: Küme merkezleri

n : Eleman sayısı

$||z||^2 = \sum z_i^2$: $z = (z_1,..., z_n)$ için Öklit mesafesi olmak üzere; wcss değeri aşağıdaki eşitlikler kullanılarak hesaplanmaktadır.

$$wcss = \sum_{i=1}^{k} \sum_{x \in S_i} ||x - \mu_i||^2 \tag{3.6}$$

$$\mu = \frac{1}{n} \sum_{i=1}^{k} x_i \tag{3.7}$$

Yukarıda ||z|| notasyonuyla gösterilen ifadede z yerine; a = (a_1,...a_n) ve b = (b_1,...,b_n) gibi iki vektör kullanılarak, Öklit uzaklığı hesaplanmak istensin. Bunun için aşağıdaki eşitlik kullanılır.

$$||a - b|| = d(a, b) = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2} \tag{3.8}$$

Algoritmanın çalışma mantığı aşağıdaki gibidir:

- Farklı k değerleri için kümeleme algoritması (örneğin k-ortalama) çalıştırılır
- Her bir k için, küme içi kareler toplamı (wcss) hesaplanır
- Kümelerin sayısına göre wcss eğrisi çizdirilir
- Eğride dirsek oluşan yer genellikle uygun küme sayısı olarak kabul edilir

Örneğin Şekil 3.1.'de k-ortalama algoritması, Iris veri seti üzerinde iteratif olarak 1'den 10'a kadar olan k değerleri için çalıştırılmıştır. Grafikten bu veri seti için uygun küme sayısının 5 olduğu sonucuna varılabilir.

Şekil 3.1. Dirsek yöntemi

3.3.2. Siluet (Silhouette) yöntemi

Siluet (Rousseeuw, 1987) yöntemi de dirsek yöntemi gibi, veri kümeleri içindeki tutarlılığın yorumlanması ve doğrulanması için kullanılır. Siluet analizi, elde edilen kümeler arasındaki ayrılma mesafesini incelemek için kullanılabilir. Siluet grafiği, bir kümedeki her bir noktanın komşu kümelerdeki noktalara ne kadar yakın olduğunun bir ölçüsünü gösterir ve böylece küme sayısı gibi parametreleri görsel olarak değerlendirmenin bir yolunu sunar.

Siluet değeri, bir nesnenin, diğer kümelere kıyasla kendi kümesine ne kadar benzer olduğunun bir ölçüsüdür. Bu değer [-1,+1] arasındadır. Burada +1'e yaklaşan yüksek bir değer o nesnenin kendi kümesine iyi uyduğunu ve komşu kümelerle iyi eşleşmediğini gösterir. 0 değeri, nesnenin komşu iki küme arasındaki karar sınırına çok yakın olduğunu ve negatif değerler, bu nesnelerin yanlış kümeye atanmış olabileceğini gösterir.

a(i) : i'nin A kümesi içindeki diğer gözlemlere olan ortalama benzemezliği (dissimilarity)

d(i,C) : i'nin, A'dan farklı bir C kümesi içindeki diğer gözlemlere olan ortalama benzemezliği

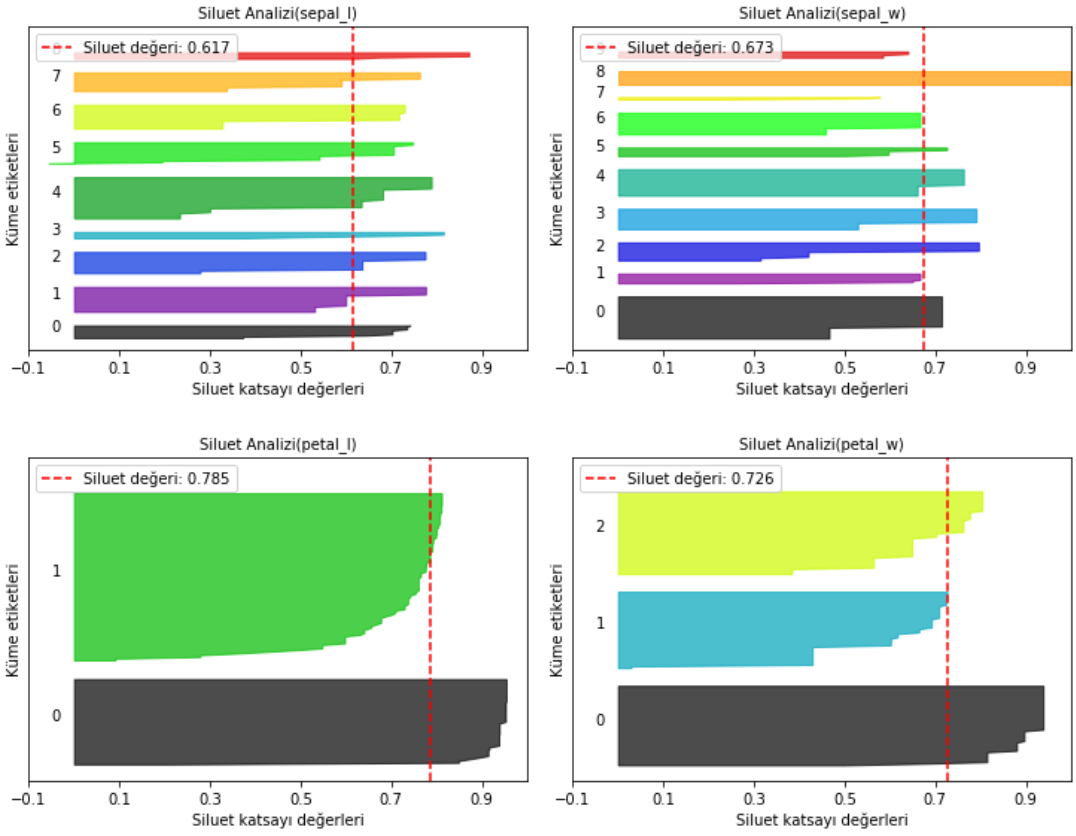
b(i) : $\underset{A \neq C}{minimum} \; d(i, C)$ olmak üzere siluet değeri s(i), aşağıdaki şekilde hesaplanır.

$$s(i) = \begin{cases} 1 - a(i)/b(i), & a(i) < b(i \\ 0 \quad , & a(i) = b(i) \\ b(i)/a(i) - 1, & a(i > b(i) \end{cases} \tag{3.9}$$

Bu ifadelerde, i'nin diğer gözlem değerlerine olan uzaklığı Öklid mesafesi kullanılarak hesaplanır. İfadeleri aşağıdaki şekilde formülize etmek mümkündür. Formülden, -1≤ s(i) ≤ 1 olduğu kolaylıkla söylenebilir.

$$s(i) = \frac{b(i) - a(i)}{max\ \{a(i), b(i)\}} \tag{3.10}$$

Şekil 3.2.'de Iris veri setinin öznitelikleri için k-ortalama algortiması ile elde edilen siluet katsayı değerleri görülmektedir. Siluet katsayısının en yüksek değere yaklaştığı küme sayısı, verinin bölünmesi gereken aralık sayısına eşdeğerdir. Siluet yöntemi ile k-ortalama algortimasında, Iris veri setinin *sepal_l* özniteliği için dokuz, *sepal_w* özniteliği için on, *petal_l* özniteliği için iki ve *petal_w* özniteliği için üç küme sonucu çıkmıştır. Bunun anlamı, Iris veri seti siluet yöntemi kullanılarak ayrıklaştırılmak istendiğinde, *sepal_l* özniteliği dokuz, *sepal_w* özniteliği on, *petal_l* özniteliği iki ve *petal_w* özniteliği üç aralığa bölünecek şekilde ayrıklaştırma işlemine tabi tutulması gereğidir.



Şekil 3.2. Iris veri seti için siluet kümeleri

3.4. Karekök Ki-Birleştirme

Karekök Ki-Birleştirme (kkKiB), veri setlerindeki her bir öznitelik için hesaplanan karekök değerine göre, verinin, kesikli değerlere dönüştürüldüğü bir yöntemi ifade etmektedir. Karekök değerine göre ayrıklaştırma işlemi aşağıdaki adımlar izlenerek yapılmaktadır:

- Veri setindeki her bir öznitelik, farklı değerleri için küçükten büyüğe doğru sıralanır
- Sıralama sonucu çıkan değerin karekökü alınır
- Karekök değeri en yakın tamsayıya yuvarlanır
- Mevcut öznitelik bu karekök değerine göre KiB ile ayrıklaştırılır

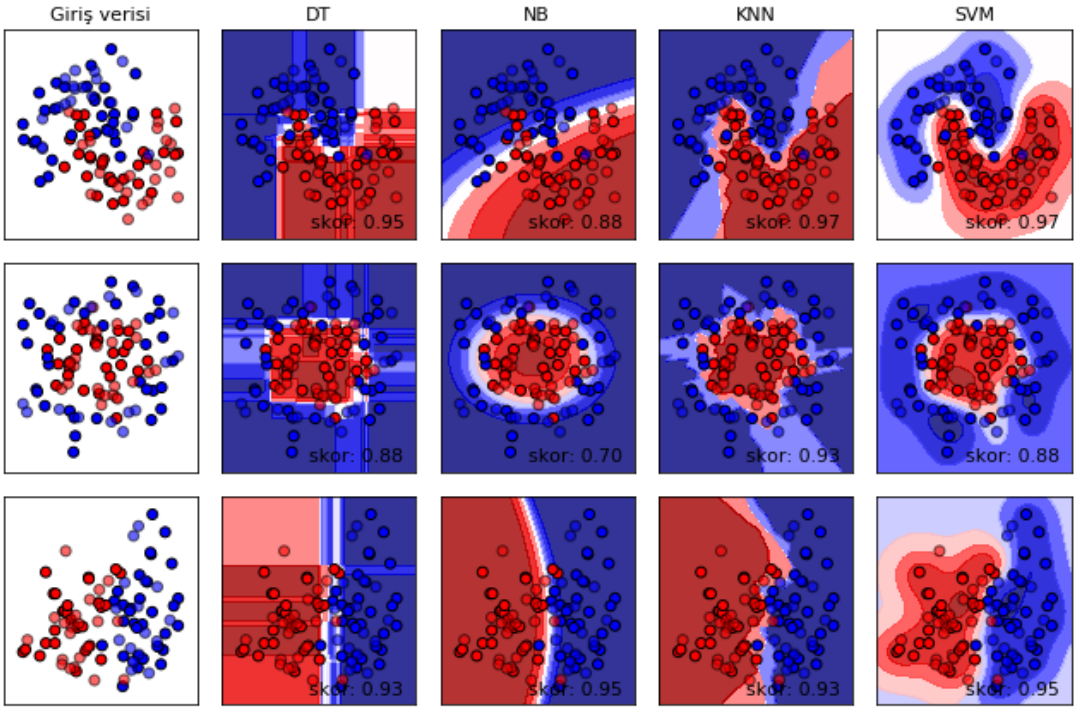
Örneğin, Iris veri setinde *sepal_l* özniteliği için farklı sayıda 35 adet değer bulunmaktadır. Bu öznitelik için karekök değeri yaklaşık 5.92 -35'in karekökü- olarak bulunmuş ve bu değer en yakın tamsayı olan 6'ya yuvarlanmıştır. Daha sonra bu öznitelik için KiB algoritmasında bu karekök değeri, özniteliğin ayrılacağı küme sayısı olarak belirlenmiştir.

3.5. 2-10'lu Ayrıklaştırma

2-10'lu ayrıklaştırma olarak adlandırılan yöntemde, veri setleri 2'den başlanarak 10'a kadar kümelere bölünerek ayrıklaştırma işlemine tabi tutulmaktadır. Diğer bir deyişle, veri setleri sırasıyla 2, 3, 4, 5, 6, 7, 8, 9 veya 10 kümeye ayrılacak şekilde KiB algoritması ile ayrıklaştırılmakta ve anlamlılık seviyesi (α değeri) 0.05 olarak alınan orijinal Ki-birleştirme algoritması ile karşılaştırılmaktadır.

3.6. Sınıflama Algoritmaları

Yöntemlerin başarımını ölçmede, aşağıda kısaca çalışma mantığı anlatılan DT, NB, KNN ve SVM sınıflama algoritmaları kullanılmaktadır. Bu algoritmaların üç farklı örnek veri kümesi üzerindeki sınıflama başarım sonuçları Şekil 3.3.'te görülmektedir.



Şekil 3.3. Sınıflama algoritmalarının örnek veri kümeleri üzerinde başarım sonuçları

3.6.1. Karar ağaçları (Decision trees-DT)

Karar ağaçları (Belson, 1959; Hunt ve ark., 1966; Breiman ve ark., 1984), sınıflandırma ve tahmin için kullanılan bir veri madenciliği ve makine öğrenmesi yaklaşımıdır. Bir karar ağacında, her bir iç düğüm bir öznitelik üzerinde bir koşul testini ifade eder. Her bir dal testin sonucunu ve her yaprak düğüm (terminal düğüm) ise bir sınıf etiketini içerir. Hedef değişkenin ayrı bir değer kümesi alabildiği ağaç modellerine sınıflandırma ağaçları denir; bu ağaç yapılarında, yapraklar sınıf etiketlerini temsil eder ve dallar, bu sınıf etiketlerine götüren özelliklerin birleşimlerini temsil eder. Hedef değişkenin sürekli değerler alabildiği karar ağaçlarına regresyon ağaçları denir. Karar ağaçları, anlaşılırlıkları ve basitlikleri nedeniyle sıklıkla kullanılan makine öğrenimi algoritmaları arasında yer alır (Wu ve ark., 2008; Piryonesi ve El-Diraby, 2020).

Karar ağaçlarını kullanarak verinin sınıflandırılması işlemi iki adımda gerçekleştirilir. İlk adım öğrenme sürecidir. Bunun için veri, eğitim ve test olmak üzere istenen oranda iki parçaya bölünür. Öğrenme sürecinde eğitim verisi, oluşturulacak modeli belirlemek

amacıyla analiz edilir. Analiz sonucu oluşturulan model, sınıflama kurallarını içeren bir karar ağacı olarak gösterilir. İkinci adım, model üzerinde test verisi kullanılarak, karar ağacının doğruluğunun, diğer bir deyişle sınıflama başarısının belirlendiği aşamadır. Eğer modelin doğruluğu kabul edilebilir bir oranda ise, kurallar yeni verilerin sınıflandırılması için kullanılır. Karar ağaçlarının bazı avantaj ve dezavantajları aşağıda sıralanmaktadır.

Avantajları:

- İnşa etme maliyeti düşüktür
- Anlaşılması ve yorumlanması kolaydır
- Bilinmeyen kayıtları sınıflandırmada son derece hızlıdır
- Küçük boyutlu ağaçlar için yorumlanabilirliği yüksektir
- Birçok veri seti için diğer yöntemler ile yarışabilen doğruluğa sahiptir
- Diğer sınıflama teknikleri ile birleştirilebilir
- Önemsiz öznitelikleri içermez

Dezavantajları:

- Aşırı öğrenme problemine yatkınlık gösterebilir
- Karar ağacı modelleri, genellikle öznitelikler üzerinde çok sayıda seviyeye sahip bölünmelere eğilimlidir
- Eğitim verilerindeki küçük değişiklikler, karar mantığında büyük değişikliklere neden olabilir
- Büyük ağaçların yorumlanması zor olabilir ve verdikleri kararlar sezgiye aykırı görünebilir
- Veride birçok değer belirsiz veya birçok sonuç birbiriyle bağlantılı ise hesaplamalar karmaşık hale gelebilir

Karar ağaçlarında, ağaç yapısını oluşturmada bilgi kazancı (information gain) (Quinlan, 1986), entropi (entropy) (Shannon, 1948), Gini indeksi (Gini index) (Gini, 1912) gibi ölçümler kullanılmaktadır. Gini safsızlık ölçüsü, karar ağacı algoritmalarında bir kök düğümden en uygun bölünmeye ve sonraki uygun bölünmelere karar vermek için kullanılan yöntemlerden biridir. Ağaç, Gini değeri en düşük özniteliğe göre bölünür. Diğer bir deyişle Gini indeksi en küçük olan öznitelik,

kök düğüm olarak tercih edilir ve devamında da hesaplanan Gini değerlerinin düşük olanlarına göre bölünme sürdürülür. İndeks, sadece ikili bölünme yapabildiği için nitelik değerleri birden fazla olsa bile, ikili gruplar halinde değerlendirilir. Gini indeksinin genel denklemi, Denklem 3.11'deki gibi ifade edilir.

$$Gini = 1 - \sum_{j=1}^{c} \left(p_j\right)^2 \tag{3.11}$$

Denklemdeki p_j değeri j. sınıfın olasılığı olarak tanımlanmaktadır. İkili gruplandırmada $Gini_{sağ}$ ve $Gini_{sol}$ değerleri hesaplandıktan sonra ilgili niteliğe ait Gini değeri bulunur.

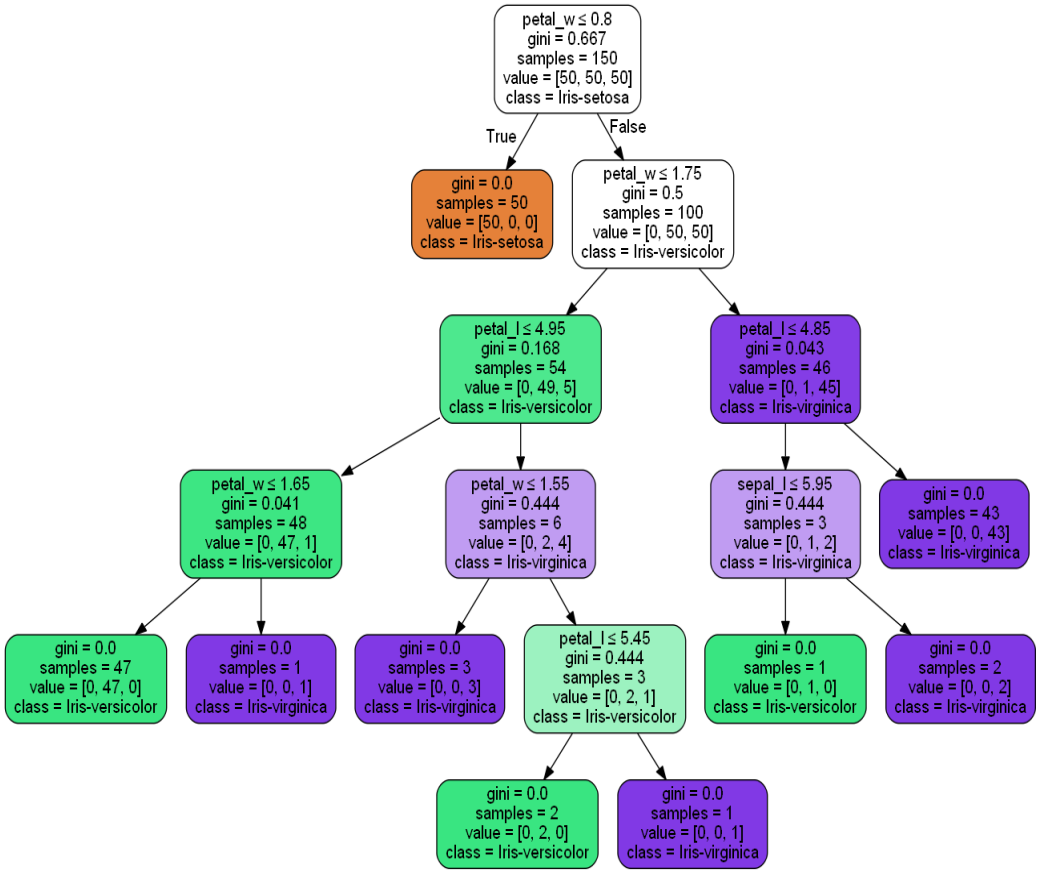
Li : Sol bölümde i grubunda bulunan örnek sayısı
Ri : Sağ bölğmde i grubunda bulunan örnek sayısı
c : Sınıfların sayısı
S : İlgili düğümdeki örnek sayısı
$|S_{sağ}|$: Sağ bölümdeki örnek sayısı
$|S_{sol}|$: Sol bölümdeki örnek sayısı olmak üzere; $Gini_{sağ}$, $Gini_{sol}$ ve ilgili niteliğe ait $Gini_{nitelik}$ değerleri aşağıdaki gibi hesaplanır:

$$Gini_{sağ} = 1 - \sum_{i=1}^{c} \left(\frac{R_i}{|S_{sağ}|}\right)^2 \tag{3.12}$$

$$Gini_{sol} = 1 - \sum_{i=1}^{c} \left(\frac{L_i}{|S_{sol}|}\right)^2 \tag{3.13}$$

$$Gini_{nitelik} = \frac{1}{S}\left(|S_{sağ}|Gini_{sağ} + |S_{sol}|Gini_{sol}\right) \tag{3.14}$$

Şekil 3.4., Iris veri seti için Gini algortiması kullanılarak oluşturulan bir karar ağacını göstermektedir.



Şekil 3.4. Iris veri seti için karar ağacı

Karar ağaçları, biyomedikal mühendisliğinde modellemeden (Shaikhina ve ark., 2015) finansal analizde kredi risk değerlendirmesine (Satchidananda ve Simha, 2006), astronomide galaksilerin sınıflandırılmasındana (Ball ve ark., 2006), tıpta tanı koymaya (Shouman ve ark., 2011), sistem kontrolünden (Colledanchise ve Ögren, 2016) imalata (Deradjat ve Minshall, 2018) kadar birçok alanda kullanılmaktadır.

3.6.2. Naive Bayes (NB)

Naive Bayes, Bayes teoreminin (Bayes, 1763), sınıf değişkeninin, değeri verilen her bir özellik çifti arasında koşullu bağımsızlık varsayımıyla uygulamasına dayanan olasılıksal ve denetimli bir öğrenme algoritmasıdır (Murty ve Devi, 2011). Basit bir ifadeyle, bir NB sınıflandırıcısı, bir sınıfın belirli bir özniteliğinin varlığının veya yokluğunun başka herhangi bir özniteliğin varlığı veya yokluğu ile ilgisi olmadığını varsayar. Örneğin, kırmızı, yuvarlak ve çapı yaklaşık 8 cm olan bir meyve elma olarak düşünülebilir. Bu öznitelikler birbirine veya diğer özniteliklerin varlığına bağlı olsa

bile, bir NB sınıflandırıcısı, tüm bu özniteliklerin bağımsız olarak bu meyvenin bir elma olma olasılığına katkıda bulunduğunu düşünür. NB sınıflandırıcısı, temeldeki varsayım doğru olmasa bile oldukça iyi performans gösterir. Sınıflandırma algoritmalarını karşılaştıran çalışmalar (Jordan, 2002; Rennie ve ark., 2003; Caruana ve Niculescu-Mizil, 2006), NB sınıflandırıcısının performans açısından lojistik regresyon, karar ağaçları ve SVM sınıflandırıcılarıyla karşılaştırılabilir olduğunu bulmuştur. Naive Bayes sınıflandırıcısı, sınıf değişkeni *y* ve bağımlı öznitelik vektörü *x* arasında aşağıdaki ilişkiyi (Denklem 3.15) ifade eder:

$$P(y/x_1,\ldots\ldots,x_n) = \frac{P(y)P(x_1,\ldots.x_n/y)}{P(x_1,\ldots.x_n)} \tag{3.15}$$

Yukarıdaki ifade (Denklem 3.15), aşağıda (Denklem 3.16) ifade edilen temel koşullu bağımsızlık varsayımı kullanılarak bütün *i* değerlerini kapsayacak şekilde daha basit bir formda Denklem 3.17'deki gibi yazılabilir (Zhang, 2004).

$$P(x_i/y,x_1,\ldots,x_{i-1},x_{i+1},\ldots x_n) = P(x_i/y) \tag{3.16}$$

$$P(y/x_1,\ldots,x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i/y)}{P(x_1,\ldots.x_n)} \tag{3.17}$$

Veri setindeki bütün girişler için $P(x_1,\ldots.x_n)$ sabit olduğundan, aşağıdaki sınıflandırma kuralı (Denklem 3.18) kullanılabilir:

$$P(y/x_1,\ldots,x_n) \propto P(y)\prod_{i=1}^{n} P(x_i/y) \tag{3.18}$$

Sınıf değişkeni (y), sadece iki sonuçlu değerler almaz. Sınıflandırmanın çok değişkenli olabileceği durumlar da olabilir. Bu nedenle, $\hat{y}$ maksimum olasılıklı y sınıfını ifade etmek üzere; aşağıdaki eşitlik (Denklem 3.19) kullanılır.

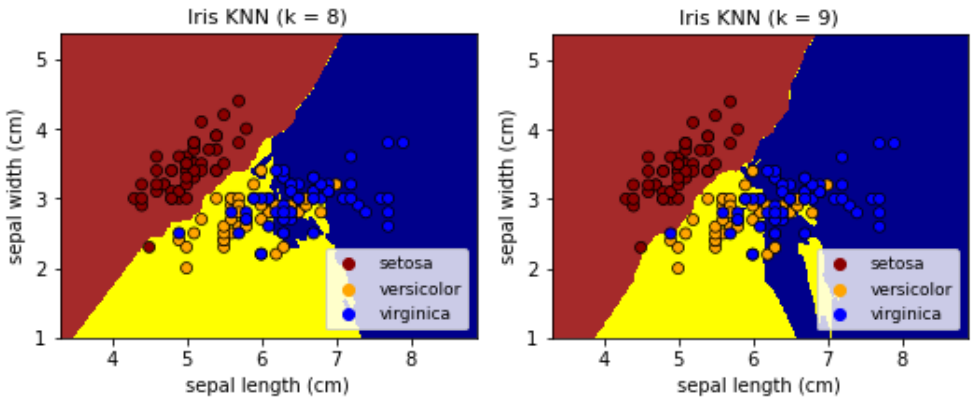
$$\hat{y} = \underset{y}{argmax} P(y)\prod_{i=1}^{n} P(x_i/y) \tag{3.19}$$

NB sınıflayıcıları, çoğunlukla duyarlılık analizi (McCandless ve ark., 2007), spam filtreleme (Almeida ve ark., 2011), ses ve metin sınıflandırma (Nigam ve ark., 2000), hastalık teşhisi (Lakoumentas, 2012), elektrokardiyografi (EKG) görüntülerinin sınıflandırılması (Wiggins, 2007) vb. alanlarda kullanılmaktadır. Hızlı ve kolay uygulanmaları, sınıflandırma için gerekli parametreleri tahmin etmede az sayıda eğitim verisi gerektirmeleri, NB'in önemli birer avantajıdır. En büyük dezavantajı ise, tahmin edicilerin (predictors) bağımsız olması gerekliliğidir. Oysa birçok gerçek dünya vakasında, tahmin ediciler bağımlıdır ve bu durum sınıflandırıcının performansını olumsuz yönde etkileyebilir. Buna rağmen gerçek dünya varsamında, NB'nin sınıflandırmadaki rekabetçi performansı şaşırtıcıdır (Zhang, 2005). NB'nin diğer bir dezavantajı ise sıfır frekans (zero frequency) problemidir. Sıfır frekans, bir kategorik değişkenin, test veri setinde eğitim veri setinde bulunmayan bir kategoriye sahip olması sonucunda ortaya çıkar. Böyle bir durumda model, kategorik değişkene sıfır olasılık atar ve bir tahmin yapamaz. NB sınıflayıcılar bu problemi Laplace yumuşatma (Laplace smoothing) kullanarak çözse de bu çözüm, sürekli öznitelik içeren veri setleri üzerinde her zaman iyi sonuç vermeyebilir (John ve Langley, 2013).

3.6.3. K-En yakın komşular (K-nearest neighbors-KNN)

En yakın komşu algoritması (Fix ve Hodges, 1951; Cover ve Hart, 1967), sınıflandırma ve regresyon için kullanılan parametrik olmayan, tembel öğrenme (lazy learning) tabanlı bir yöntemdir. KNN'nin parametrik olmayışı, çalışılan veriler üzerinde herhangi bir varsayımda bulunmadığı; tembel bir algoritma oluşu ise herhangi bir genelleme yapmak için eğitim veri noktalarını kullanmadığı anlamına gelir. KNN sınıflamasında, çıktı bir sınıf üyeliğidir. K pozitif bir sayı olmak üzere bir gözlem noktası, kendisine en yakın bu k adet komşusuna ait mesafesine bakılarak sınıflandırılır. Başka bir ifadeyle, o noktanın en yakın komşularının basit çoğunluk oyu hesaplanır ve nokta, en yakın komşuları içinde en çok temsilciye sahip olan veri sınıfına atanır. Bu durum, bütün komşuların eşit ağırlıklı olarak değerlendirildiği KNN sınıflandırmasıdır. Bir diğer yöntemde ise noktanın, en yakın k komşusunun her birine olan mesafesi hesaba katılarak sınıflandırmayı ağırlıklandırmaktır. En yakın k komşunun her birinin sınıfı, o noktadan gözlem noktasına olan mesafenin tersiyle

orantılı bir ağırlık ile çarpılır (Samworth, 2012). K değerinin veri setinin yapısına uygun olarak seçilmesi (Everitt ve ark., 2011) oldukça önemlidir. Gereğinden düşük ya da yüksek bir k değeri, sınıflama başarısını doğrudan etkilemektedir. Örneğin Şekil 3.5.'te Iris veri setinin iki özniteliği için yapılan KNN sınıflandırmasında, k değerini 8'den 9'a çıkarmak bile, bazı gözlem değerlerinin ait oldukları *setosa*, *versicolor* veya *virginica* sınıflarından farklı bir sınıfa atanmalarına yol açmaktadır. KNN'de mesafe ölçümü için Öklid, Minkowski, Manhattan vb. metrikler kullanılır. Tez kapsamında Denklem 3.8 ile ifade edilen Öklid mesafesi kullanılmaktadır.



Şekil 3.5. Iris veri setinin farklı iki k değeri için KNN ile sınıflandırılması

KNN, boyut indirgeme (dimension reduction) (Beyer ve ark., 1999), özellik çıkarımı (feature extraction) (Shaw ve Jebara, 2009), karar sınırı belirleme (decision boundary) (Bremner ve ark.,2005), eksik veri tamamlama (data imputation) (Batista ve Monard, 2002; Jonsson ve Wohlin, 2004; Zhang, 2012; Choudhury ve Kosorok, 2020), anomali tespiti (anomaly detection) (Ramaswamy ve ark.,, 2000; Wu ve ark., 2019), hastalık teşhisinde (Saini ve ark., 2015; Rajathi ve Radhamani, 2016) gibi bir çok alanda kullanılmaktadır. KNN'nin bazı avantaj ve dezavantajları aşağıda sıralanmaktadır.

Avantajları:

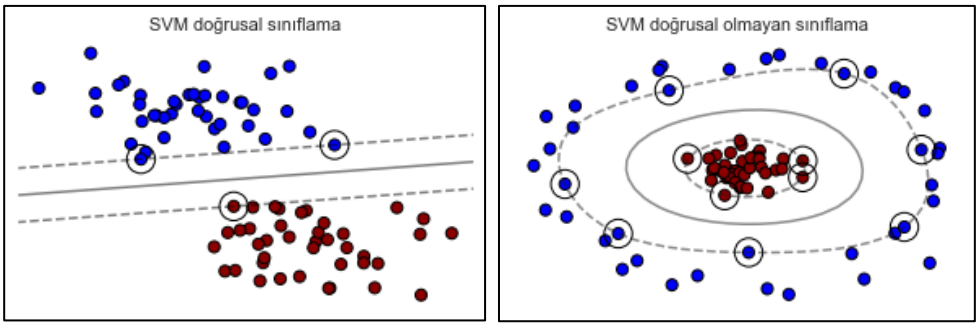
- KNN'nin uygulanması kolaydır
- KNN, tahmin yapmadan önce eğitim gerektirmediğinden, yeni veriler sorunsuz bir şekilde eklenebilir
- Yorumlanabilirliği yüksektir
- Veriler hakkında varsayım yapmaya, model oluşturmaya gerek yoktur

Dezavantajları:

- Doğruluk, verilerin kalitesine bağlıdır
- Büyük verilerle tahmin aşaması yavaş olabilir
- Gürültülü verilere, eksik ve aykırı değerlere karşı hassastır
- Yüksek bellek gerektirir
- Büyük boyutlu veriler için hesaplama maliyeti yüksektir

3.6.4. Destek vektör makineleri (Support vector machines-SVM)

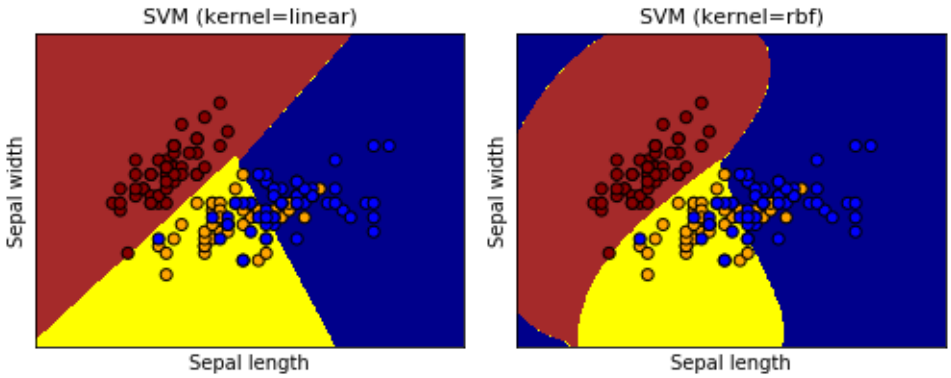
Destek vektör makineleri (Boser ve ark., 1992), sınıflandırma ve regresyon analizi için kullanılan denetimli bir öğrenme yöntemidir. Her biri iki kategoriden birine ait olarak işaretlenmiş bir dizi eğitim örneği verildiğinde, SVM eğitim algoritması, bir kategoriye veya diğerine yeni örnekler atayan ve onu olasılıklı olmayan ikili doğrusal sınıflandırıcı yapan bir model oluşturur. Model, iki kategori arasındaki boşluğun genişliğini en üst düzeye çıkarmak için eğitim örneklerini uzaydaki noktalarla eşler. Yeni örnekler daha sonra aynı alana eşlenir ve boşluğun hangi tarafına denk geldiklerine bağlı olarak ait oldukları kategori tahmin edilir. Bu durumda sınıfları ayırmak için doğru kullanmak yeterli olur. Ancak verilerin doğrusal dağılmadığı durumlarda bu yöntem kullanılamaz. Bunun yerine veriler, bir düzlem yardımıyla ayrıştırılır. Bunun için, SVM'ler, girdilerini yüksek boyutlu özellik alanlarına dolaylı olarak eşleyen bir çekirdek (kernel) yapısı kullanarak bir hiper düzlem (hyperplane) yardımıyla doğrusal olmayan sınıflandırmayı da etkin bir şekilde gerçekleştirebilir. Şekil 3.6., doğrusal bir veri kümesi (a) ve doğrusal olmayan bir veri kümesi (b) üzerinde karar sınırına en yakın veri noktaları olan, SVM'nin işaretlediği destek vektörleri (support vectors) -siyah çember ile işaretli- görülmektedir. Şekil 3.6 (a)'da sınıfları ayıran doğru parçası karar sınırı (decision boundary) ve kesikli çizgi ile gösterilen doğru parçaları arasında kalan alan ise marj (margin) olarak ifade edlir. SVM'nin hedefi, marjı mümkün olduğunca geniş tutacak şekilde bir karar sınırı belirleyerek sınıflandırmayı gerçekleştirmektir.



a) b)

Şekil 3.6. SVM'nin doğrusal (a) ve doğrusal olmayan (b) veri için destek vektörleri

Şekil 3.7.'de Iris veri seti için SVM ile çekirdek fonksiyonun doğrusal alındığı (kernel=linear) doğrusal ve çekirdek fonksiyonun radial basis function (kernel = rbf) olarak alındığı doğrusal olmayan bir sınıflandırma örneği görülmektedir.



Şekil 3.7. Iris veri setinin farklı iki kernel değeri için SVM ile sınıflandırılması

SVM, metin (Joachims, 1998; Wang ve ark., 2006), resim (Bazi ve Melgani, 2006; Tarabalka ve ark., 2010) uydu görüntülerinin sınıflandırılması (Maity, 2016), el yazısı tespiti (Maitra ve ark., 2015; Ayyaz ve ark., 2016), yüz tanıma (Pang ve ark., 2005), hastalık teşhisi (Sweilam ve ark.,2010; Shariaty ve ark., 2019; Shankar ve ark., 2020), biyolojide gen yapıları (Shukla ve ark., 2019), protein sınıflandırılması (Cuingnet ve ark., 2011), kimya (Devos ve ark., 2014) gibi birçok alanda kullanılan bir yöntemdir.

Avantajları:

- Yüksek boyutlu veriler üzerinde etkilidir.
- Boyut sayısının örnek sayısından fazla olduğu durumlar için de kullanışlıdır
- Karar işlevinde (destek vektörleri) eğitim noktalarının bir alt kümesini kullanır, bu nedenle bellek açısından da etkilidir

- Çok yönlüdür, karar işlevi için farklı çekirdek(kernel) yapıları kullanılabilir
- Aşırı uydurmadan (overfitting) probleminden kaçınmayı sağlayan bir düzenleme (regularisation) parametresi vardır

Dezavantajları:

- İyi bir çekirdek fonksiyonu seçmek kolay değildir.
- Büyük veri kümeleri için uzun eğitim süresi gerekebilir
- Nihai modeli, değişken ağırlıkları ve bireysel etkiyi anlamak ve yorumlamak zordur
- SVM, veri kümesinde fazla gürültü olduğunda çok iyi performans göstermez
- SVM, doğrudan olasılık tahminleri sağlamaz, bunlar nispeten maliyetli sayılabilecek çapraz doğrulamanın kullanıldığı bir hesaplama yöntemi ile elde edilir

3.7. Değerlendirme Metrikleri

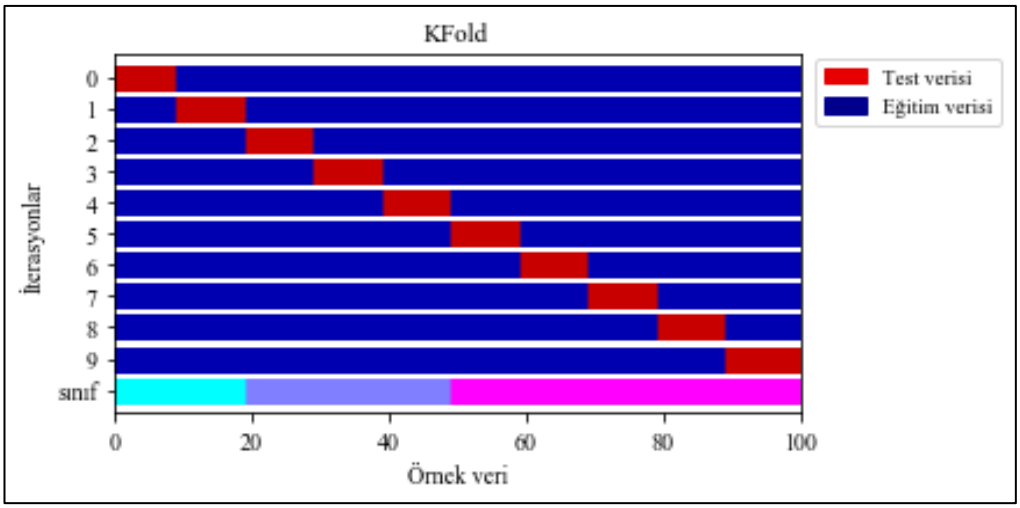
Ayrıklaştırma yöntemlerinin sınıflandırılmasında, veriye, Katmanlı K-Kat çapraz doğrulama (StratifiedKFold cross-validation) yöntemi uygulanmaktadır. Yöntemlerin sınıflama başarısı, doğruluk(accuracy), kesinlik (precision), duyarlılık(recall) ve f_1-skor(f_1-score) metrikleri kullanılarak ölçülmektedir.

3.7.1. StratifiedKFold

Çapraz doğrulama (cross-validation) (Allen, 1974; Stone, 1977), sınırlı bir veri örneği üzerinde makine öğrenimi modellerini değerlendirmek için kullanılan bir yeniden örnekleme yöntemidir. Yöntem, belirli bir veri örneğinin bölüneceği grupların sayısını ifade eden k adında bir parametreye sahiptir. Bu nedenle, genellikle k-kat çapraz doğrulama (k-fold cross validation) olarak adlandırılır. K için spesifik bir değer seçildiğinde, modele referans olarak, k yerine seçilen bu değer kullanılabilir. Örneğin k = 10 için yöntem, 10-kat çapraz doğrulama (10-fold cross validation) olarak ifade edilebilir.

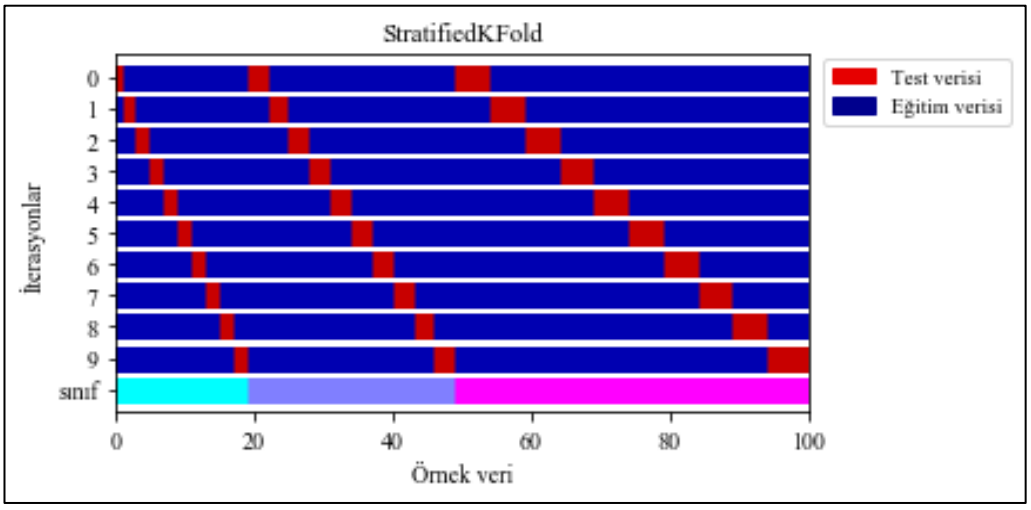
K-kat çapraz doğrulama yöntemi, bir modelin geliştirilme sürecinde aşırı öğrenme (overfitting) (Hand, 2007) ve yetersiz öğrenme (underfitting) (Everitt ve Skrondal, 2010) probleminin üstesinden gelmede önemli bir avantaj sağlar (Singh ve ark., 2018). Aşırı öğrenme, oluşturulan bir modelin eğitim verisindeki örüntüler yerine gözlemleri öğrenmesidir. Bu durumda eğitim için kullanılan veri kümesi öğrenilir, fakat model yeni gelen veri (test verisi) karşısında doğru tahminler yapamaz. Diğer bir deyişle veri üzerinde ezberleme tarzı bir öğrenme gerçekleştiği için model yeni gelen ve tanımadığı gözlemler için başarılı bir tahminleme ortaya koyamaz. Bundan dolayı aşırı öğrenmiş bir model, eğitim aşamasını küçük bir hata oranı ile tamamlar, ancak test aşamasındaki tahminde büyük bir hata oranı ortaya çıkar (Arlot ve Celisse, 2010). Yetersiz öğrenme, bir istatistiksel modelin veya bir makine öğrenmesi algoritmasının, verilerin temelindeki eğilimi yakalayamadığında ortaya çıkar. Yetersiz öğrenmenin oluşması, modelin veya algoritmanın verilere yeterince uymadığı anlamına gelir. Yetersiz öğrenmiş bir model, yeni verilerde sorunlu veya hatalı sonuçlara neden olur ve genellikle eğitim verilerinde bile kötü performans gösterir. Başka bir ifade ile bu model, hem yüksek eğitim hatası hem de yüksek test hatasına sahiptir. Yetersiz öğrenme, genellikle doğru bir model oluşturmak için yeterli veri olmadığında veya doğrusal olmayan verilerle doğrusal bir model oluşturulmaya çalışıldığında ortaya çıkar. Bu noktadan hareketle, çapraz doğrulama yöntemi, hem aşırı öğrenme, hem de yetersiz öğrenme sorununa karşı sıklıkla kullanılır.

Yöntem, gözlem setinin yaklaşık olarak eşit büyüklükte k-kata (k-fold) rastgele olarak bölünmesini içerir. İlk kat, doğrulama kümesi olarak alınır, kalan k - 1 kat ise eğitim verisi olarak kullanılır. Böylece, veri örneğindeki her bir gözlem ayrı bir gruba atanır ve süreç boyunca o grupta kalır. Bu, her bir gözlemin 1 kez test setinde ve k-1 kez de modeli eğitmek için kullanılan eğitim setinde yer aldığı anlamına gelir. Şekil 3.8., 100 adet gözlem değeri içeren örnek bir veri seti üzerinde 10-kat çapraz doğrulama yönteminin uygulanışını göstermektedir. Şekilden de anlaşılabileceği gibi, her bir iterasyonda test verisi bir sonraki alt kümeden seçilmekte ve verinin kalan kısmı eğitim için kullanılmaktadır.



Şekil 3.8. 10-kat çapraz doğrulama

K-kat çapraz doğrulama yöntemi bölümleme yaparken sınıf dağılımlarını dikkate almaz. Katmanlı k-kat çapraz doğrulama (StratifiedKFold cross validation), çapraz doğrulamayı bir adım öteye taşıyarak, veri kümesindeki sınıf dağılımlarını, eğitim ve test bölümlerinde korur. Bu açıdan, sınıf dağılımlarının dengesiz olduğu veya veri boyutunun küçük olduğu durumlarda genellikle yararlı bir yöntemdir. Örneğin, 3 sınıfa sahip, 100 adet gözlem değeri içeren ve sınıf dağılımları sırasıyla %20, %30 ve %50 olan bir veri kümesi olsun. Katmanlı 10-kat çapraz doğrulama yapılan bir durumda, her bir iterasyon için test verisi içinde; birinci sınıfa ait 2, ikinci sınıfa ait 3 ve üçüncü sınıfa ait 5 adet veri bulunur. Verinin geri kalanı ise eğitim için kullanılır. Şekil 3.9., bu 100 adet gözlem değerini içeren örnek bir veri kümesi için yapılan katmanlı 10-kat çapraz doğrulamayı göstermektedir. Tez kapsamında katmanlı 10-kat çapraz doğrulama yöntemi kullanılmaktadır.



Şekil 3.9. Katmanlı 10-kat çapraz doğrulama

3.7.2. Karmaşıklık matrisi

Karmaşıklık matrisi (confusion matrix) veya hata matrisi (error matrix) (Stehman, 1997), bir modelin başarım sonuçlarını gösteren önemli bir değerlendirme yöntemidir. Makine öğrenimi alanında ve özellikle istatistiksel sınıflandırma problemlerinde, bir algoritmanın performansının görselleştirilmesine izin veren özel bir tablo düzenidir ve yukarıda bahsi geçen metrikleri bulmada kullanılır. Tablo 3.5. karmaşıklık matrisini göstermektedir.

Tablo 3.5. Karmşıklık matrisi

	Tahmin Edilen (Predicted)	
Gerçek Durum(Actual)	True Positive(TP)	False Negative(FN)
	False Positive(FP)	True Negative(TN)

True Positive ve True Negative modelin doğru olarak tahmin ettiği, False Positive ve False Negative ise modelin yanlış olarak tahmin ettiği değerlerdir (Powers, 2011).

- True Positive (TP) : Pozitif olarak tahmin edilen ve gerçekte de pozitif olan değerlerdir.
- True Negative (TN): Negatif olarak tahmin edilen ve gerçekte de negatif olan değerlerdir.
- False Positive (FP) (Tip1 Hata): Pozitif tahmin edilen ama gerçekte negatif olan değerlerdir.
- False Negative (FN) (Tip2 Hata): Negatif tahmin edilen ama gerçekte pozitif olan değerlerdir.

Doğruluk (Accuracy): Doğruluk değeri modelde doğru tahmin edilen değerlerin veri kümesindeki toplam değerlere oranı ile hesaplanmaktır.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{3.20}$$

Kesinlik (Precision): Positive olarak tahmin edilen değerlerin gerçekte de kaç tanesinin pozitif olduğunu göstermektedir.

$$\text{Precision} = \frac{TP}{TP+FP} \tag{3.21}$$

Duyarlılık (Recall): Positive olarak tahmin edilmesi gereken değerlerin ne kadarının pozitif olarak tahmin edildiğini gösteren bir metriktir.

$$\text{Recall} = \frac{TP}{TP+FN} \tag{3.22}$$

F_1-skor (F_1-score): F_1-skor değeri, kesinlik (Precision) ve duyarlılık (Recall) değerlerinin harmonik ortalamasını göstermektedir**.**

$$F_1\text{-score} = 2 * \frac{Precision*Recall}{Precision+Recall} \tag{3.23}$$

BÖLÜM 4. ARAŞTIRMA BULGULARI

4.1. Analizlerde Kullanılan Yazılımlar ve Kodlar

Bu kısımda yöntemlerin uygulama esaslarından bahsedilmiştir. Bu kapsamda Bölüm 3'te ayrıntılı bir şekilde açıklanan yöntemlerin uygulamadaki karşılığının daha iyi anlaşılabilmesi için kodları da (pseudocode) verilmiştir. Yöntemlerin uygulanmasında Python programlama dili kullanılmıştır. Python, veri analizi için oldukça gelişmiş kütüphanelere sahiptir. Yöntemlerin bütün uygulama ve veri madenciliği işlemleri için, Python'ın temel makine öğrenmesi kütüphanesi olan *scikit-learn* kullanılmıştır. Veri setlerinin işlenmesi ve analizi için *pandas*; yöntemlerin ihtiyaç duyduğu matematiksel işlemler için *numpy*; verilerin görselleştirilmesinde ise *matplotlib* kütüphanesi kullanılmıştır. Python dilini uygulamak için birçok editör bulunmaktadır. Tezde, veri analizi için oldukça geniş bir uygulama alanına sahip olan Spyder kullanılmıştır. Bilimsel bir Python geliştirme ortamı olan Spyder, Anaconda ile birlikte gelen entegre bir geliştirme ortamıdır (integrated development environment). Anaconda, Python için açık kaynaklı olarak dağıtılan bir paket ve çevre yöneticisidir.

```
input:veriseti V
for sutun in columns.V:
    fonksiyon KiB(aralik_sayisi))
        farkli_degerler=sorted(set(V[sutun]))
        for i in farkli_degerler
            araliklar = [farkli_degerler[i], farkli_degerler[i]]#başlangıçta her değer bir aralıktır
        while araliklar > aralik_sayisi
                for j=0 to (len(araliklar))
                    her bitişik çift için denklem (3.1)'i kullanarak χ2 hesapla
                    yeni_araliklar = en küçük χ2'ye sahip çiftleri birleştir
                araliklar = yeni_aralikalar
        a=0
        for k in araliklar:
            for  indis, satir in V.itterrows():
                if V.at[indis , sutun] and satir[sutun]>= k[0] and satir[sutun]<= k[1]:
                V.at[indis, sutun] = a
            a=a+1
output:  Ayrık V
```

Şekil 4.1. KiB(k) yönteminin kodu

Şekil 4.1.'de tezde önerilen yöntemlere temel teşkil eden ve Python dili ile yazılan KiB(k) yönteminin sözde kodu verilmiştir. Parametre olarak verilen 'k' değeri, veri setlerinin; ya da sKiB ve kkKiB algoritmalarında, verinin herbir özniteliğinin ayrıklaştırma esnasında bölüneceği aralık sayısını ifade etmektedir. Şekil 4.2. dKiB, Şekil 4.3. sKiB, Şekil 4.4. kkKiB, Şekil 4.5. 2-10 kümeleme yönteminin sözde kodunu ifade etmektedir.

```
input: veriseti V
import KMeans
veri = V[columns !='class']
for k=2 to 10
    k-ortalama= KMeans(n_clusters=k)
    k-ortalama.fit(veri)
    wcss eğrisini çiz
k = wcss eğrisinde dirsek oluşturan k değeri
KiB(k)
output: Ayrık V
```

Şekil 4.2. dKiB yönteminin kodu

```
input: veriseti V
import KMeans, silhouette_score
for sutun in  V[columns !='class']
    for k=2 to 10
        k-ortalama = KMeans (n_clusters=k)
        kumele = k-ortalama.fit_predict(sutun)
        siluet = silhouette_score (sutun,kumele)
     k= en yüksek siluet değeri
     KiB(k)
output: Ayrık V
```

Şekil 4.3. sKiB yönteminin kodu

```
input: veriseti V
import math
for sutun in  V[columns !='class']
    farkli_degerler=sorted(set(V[sutun]))
    deger = square(len(farkli_değerler))
    karekok1 = math.floor(math.sqrt(deger))
    karekok2 = math.sqrt(deger)
    if karekok1 + 0.50 < karekok2:
        karekok=karekok1+1
    else:
        karekok=karekok1
    KiB(karekok)
output: Ayrık V
```

Şekil 4.4. kkKiB yönteminin kodu

```
input: veriseti V
for i=2 to 10
    KiB(i)
output: Ayrık V
```

Şekil 4.5. 2-10 kümeleme yönteminin kodu

Orijinal KiB algoritmasının yani KiB'in anlamlılık seviyesi ile uygulandığı (α=0,05) yöntem için, R programlama dili ile yazılmış ve istatistik alanında sıklıkla kullanılan CRAN-R (R Foundation for Statistical, 2016) kütüphanesi kullanılmıştır. R dilinin uygulanmasında R studio yazılımı tercih edilmiştir.

4.2. Ayrıklaştırma Bulguları ve Analiz

Bu aşamadan itibaren yukarıda bahsi geçen veri setleri üzerinde elde edilen ayrıklaştırma sonuçlar değerlendirilmektedir. Veri setleri için öncelikle KiB, dKiB, sKiB ve kkKiB yöntemleri ile elde edilen sınıflama sonuçları tablo üzerinde gösterilmektedir. Daha sonra ise 2-10 arası kümeleme yöntemi ile elde edilen sonuçların KiB yöntemi ile karşılaştırmalı sonuçları verilmektedir. Ayrıca her bir veri seti için elde edilen sonuçların görsel grafikleri de analiz sonuçlarına eklenmektedir. Sonuçların görselleştirilmesinde izlenen yöntem KiB, dKiB, sKiB ve kkKiB için sınıflama başarım sonuçlarının grafiğini göstermek şeklinde; 2-10 arası kümeleme için ise ortalama F_1-skor değerlerine ait grafiklerin gösterimi şeklinde tercih edilmektedir. Örneğin Occupancy veri setinde 2-Küme için DT, 0.9855; NB, 0.9758; KNN, 0.9696 ve SVM, 0.9855 sonuçlarını üretmiştir. Bu veri setini 2-Kümeli F_1-skor değeri bu dört yöntemin bulduğu sonucun ortalaması olan 0.9791'dir. Her bir veri setinin kalan diğer kümeleme yöntemleri için de aynı yol takip edilmektedir.

Değerlendirmede başarı, doğruluk değerine göre öncelenmektedir. Doğruluk değeri standart sapmayı da içerecek şekilde eşit olduğu takdirde F_1-skor değeri başarı için ikincil ölçüt olarak kullanılmaktadır. F_1-skor kesinlik ve duyarlılık değerlerinin harmonik ortalaması olduğu için bu metriklerin hepsini içerecek şekilde bir değerlendirme yapılmış olmaktadır. Veri setlerinin KiB, dKiB, sKiB ve kkKiB yöntemleri ile ayrıklaştırılmış haline ait sınıflama sonuçları Tablo 4.1.'de verilmektedir. Veri setlerinin 2-10 kümeleme yöntemiyle ayrıklaştırılmasının, KiB yöntemi ile karşılaştırılma sonuçları sırasıyla Tablo 4.2., Tablo 4.3., Tablo 4.4., Tablo 4.5., Tablo 4.6., Tablo 4.7., Tablo 4.8., Tablo 4.9., Tablo 4.10., Tablo 4.11. ve Tablo 4.12.'de verilmektedir.

Tablo 4.1. KiB, dKiB, sKiB ve kkKiB yöntemleri ile ayrıklaştırılan veri setleri üzerinde elde edilen sonuçlar

Veri Seti	Yöntem	Sınıflama algoritması	Doğruluk	Kesinlik	Duyarlılık	F_1-skor
Iris	KiB	DT	0,9600 (+/- 0,09)	0,9600	0,9600	0,9600
		NB	0,9733 (+/- 0,07)	0,9534	0,9533	0,9533
		KNN	**0,9733 (+/- 0,07)**	0,9600	0,9600	0,9600
		SVM	0,9600 (+/- 0,09)	0,9471	0,9467	0,9466
	dKiB	DT	**0,9600 (+/- 0,07)**	0,9471	0,9467	0,9466
		NB	**0,9733 (+/- 0,07)**	0,9738	0,9733	0,9733
		KNN	0,9333 (+/- 0,12)	0,9488	0,9467	0,9469
		SVM	**0,9600 (+/- 0,07)**	0,9477	0,9467	0,9468
	sKiB	DT	0,9533 (+/- 0,09)	0,9485	0,9467	0,9466
		NB	0,9600 (+/- 0,09)	0,9619	0,9600	0,9599
		KNN	0,9200 (+/- 0,12)	0,9242	0,9200	0,9203
		SVM	0,9533 (+/- 0,09)	0,9564	0,9533	0,9536
	kkKiB	DT	**0,9600 (+/- 0,07)**	0,9471	0,9467	0,9466
		NB	0,9667 (+/- 0,07)	0,9668	0,9667	0,9667
		KNN	0,9467 (+/- 0,08)	0,9298	0,9267	0,9270
		SVM	**0,9600 (+/- 0,07)**	0,9550	0,9533	0,9533
Bupa	KiB	DT	0,6204 (+/- 0,18)	0,6434	0,6348	0,6370
		NB	0,6347 (+/- 0,16)	0,6495	0,6435	0,6453
		KNN	0,6290 (+/- 0,12)	0,6147	0,6116	0,6128
		SVM	0,6235 (+/- 0,13)	0,6391	0,6377	0,6031
	dKiB	DT	0,6142 (+/- 0,17)	0,5714	0,6345	0,6013
		NB	**0,6371 (+/- 0,18)**	0,6418	0,6454	0,6387
		KNN	**0,6836 (+/- 0,15)**	0,6881	0,6950	0,6915
		SVM	0,6550 (+/- 0,11)	0,6801	0,6573	0,6587
	sKiB	DT	**0,6377 (+/- 0,25)**	0,6102	0,7448	0,6708
		NB	0,5733 (+/- 0,16)	0,6706	0,5623	0,5333
		KNN	0,5976 (+/- 0,23)	0,6040	0,6207	0,6122
		SVM	0,6464 (+/- 0,19)	0,6754	0,5310	0,5946
	kkKiB	DT	0,6350 (+/- 0,15)	0,6406	0,6406	0,6406
		NB	0,6142 (+/- 0,13)	0,5789	0,6069	0,5926
		KNN	0,6752 (+/- 0,13)	0,6931	0,6550	0,6735
		SVM	**0,6699 (+/- 0,14)**	0,6572	0,6522	0,6213

Tablo 4.1. (Devamı)

Heart(Statlog)	KiB	DT	0,7074 (+/- 0,24)	0,7258	0,7500	0,7377
		NB	**0,8407 (+/- 0,06)**	0,8469	0,8450	0,8458
		KNN	0,8000 (+/- 0,09)	0,7647	0,7800	0,7723
		SVM	0,8148 (+/- 0,08)	0,8203	0,8192	0,8197
	dKiB	DT	0,7000 (+/- 0,19)	0,7895	0,7500	0,7692
		NB	0,8370 (+/- 0,08)	0,8517	0,8475	0,8492
		KNN	**0,8148 (+/- 0,13)**	0,8306	0,8296	0,8284
		SVM	0,8074 (+/- 0,10)	0,8376	0,8370	0,8361
	sKiB	DT	**0,7519 (+/- 0,17)**	0,7626	0,7630	0,7628
		NB	0,8333 (+/- 0,13)	0,8480	0,8481	0,8478
		KNN	0,8111 (+/- 0,08)	0,8075	0,8074	0,8063
		SVM	0,8111 (+/- 0,11)	0,8261	0,7917	0,8085
	kkKiB	DT	0,7333 (+/- 0,24)	0,7807	0,7417	0,7607
		NB	0,8370 (+/- 0,08)	0,8362	0,8083	0,8220
		KNN	0,8111 (+/- 0,12)	0,8253	0,8142	0,8174
		SVM	**0,8185 (+/- 0,09)**	0,8341	0,8333	0,8323
WholeSale	KiB	DT	0,8729 (+/- 0,10)	0,8712	0,8705	0,8708
		NB	0,8975 (+/- 0,11)	0,8927	0,8932	0,8929
		KNN	0,9042 (+/- 0,12)	0,8997	0,9000	0,8998
		SVM	0,9068 (+/- 0,06)	0,9067	0,9068	0,9051
	dKiB	DT	**0,9114 (+/- 0,12)**	0,8959	0,8955	0,8956
		NB	**0,9203 (+/- 0,11)**	0,9302	0,9227	0,9240
		KNN	0,9045 (+/- 0,14)	0,9020	0,9000	0,9007
		SVM	**0,9181 (+/- 0,12)**	0,9161	0,9159	0,9160
	sKiB	DT	0,8750 (+/- 0,10)	0,8817	0,8948	0,8876
		NB	0,9090 (+/- 0,11)	0,9079	0,9068	0,9072
		KNN	**0,9112 (+/- 0,12)**	0,9004	0,9000	0,9002
		SVM	0,9136 (+/- 0,13)	0,9211	0,9159	0,9170
	kkKiB	DT	0,8706 (+/- 0,09)	0,8803	0,8795	0,8799
		NB	0,8930 (+/- 0,12)	0,8959	0,8955	0,8956
		KNN	0,9046 (+/- 0,10)	0,9091	0,9091	0,9091
		SVM	0,9067 (+/- 0,11)	0,9014	0,9023	0,9012

Tablo 4.1. (Devamı)

Ecoli	KiB	DT	0,8024 (+/- 0,11)	0,8041	0,8095	0,8058
		NB	0,6821 (+/- 0,10)	0,7531	0,6964	0,7067
		KNN	0,8481 (+/- 0,04)	0,8318	0,8452	0,8351
		SVM	0,8589 (+/- 0,08)	0,8402	0,8631	0,8512
	dKiB	DT	0,8309 (+/- 0,07)	0,8260	0,8333	0,8277
		NB	0,7391 (+/- 0,10)	0,8086	0,7381	0,7624
		KNN	**0,8758 (+/- 0,08)**	0,8551	0,8631	0,8572
		SVM	**0,8763 (+/- 0,08)**	0,8563	0,8661	0,8558
	sKiB	DT	**0,8603 (+/- 0,08)**	0,8404	0,8423	0,8407
		NB	0,7672 (+/- 0,10)	0,7966	0,7530	0,7612
		KNN	0,8070 (+/- 0,12)	0,8082	0,8185	0,8096
		SVM	0,8090 (+/- 0,09)	0,7175	0,8095	0,7560
	kkKiB	DT	0,8066 (+/- 0,11)	0,7978	0,8095	0,8019
		NB	**0,8235 (+/- 0,13)**	0,8377	0,8214	0,8291
		KNN	0,8613 (+/- 0,07)	0,8131	0,8244	0,8178
		SVM	0,8550 (+/- 0,08)	0,8284	0,8482	0,8375
Vertebral	KiB	DT	0,8032 (+/- 0,26)	0,8221	0,8226	0,8223
		NB	0,7935 (+/- 0,50)	0,8194	0,7935	0,7991
		KNN	0,8194 (+/- 0,24)	0,8214	0,8126	0,8167
		SVM	0,8161 (+/- 0,24)	0,8255	0,8305	0,8279
	dKiB	DT	0,7968 (+/- 0,29)	0,8387	0,8387	0,8387
		NB	0,7645 (+/- 0,60)	0,7666	0,8050	0,7599
		KNN	0,7839 (+/- 0,17)	0,7960	0,8088	0,8015
		SVM	0,8032 (+/- 0,31)	0,8149	0,8026	0,8082
	sKiB	DT	**0,8516 (+/- 0,24)**	0,8583	0,8516	0,8536
		NB	0,7452 (+/- 0,58)	0,7450	0,7802	0,7393
		KNN	0,8065 (+/- 0,16)	0,7596	0,7677	0,7538
		SVM	0,7839 (+/- 0,36)	0,7807	0,8048	0,7886
	kkKiB	DT	0,8066 (+/- 0,11)	0,8276	0,8460	0,8351
		NB	**0,8235 (+/- 0,13)**	0,8063	0,7484	0,7566
		KNN	**0,8613 (+/- 0,07)**	0,7851	0,9048	0,8407
		SVM	**0,8550 (+/- 0,08)**	0,8947	0,8095	0,8500

Tablo 4.1. (Devamı)

Yeast	KiB	DT	0,4765 (+/- 0,07)	0,5315	0,5303	0,5281
		NB	0,1537 (+/- 0,08)	0,4007	0,1543	0,1770
		KNN	**0,5656 (+/- 0,09)**	0,5452	0,5586	0,5476
		SVM	0,5988 (+/- 0,08)	0,5872	0,6024	0,5880
	dKiB	DT	0,5010 (+/- 0,12)	0,5276	0,5317	0,5271
		NB	0,1422 (+/- 0,06)	0,4875	0,1429	0,1599
		KNN	0,5602 (+/- 0,10)	0,5583	0,5606	0,5539
		SVM	0,6040 (+/- 0,09)	0,5966	0,6078	0,5981
	sKiB	DT	**0,5502 (+/- 0,10)**	0,5630	0,5532	0,5453
		NB	0,2033 (+/- 0,09)	0,3522	0,2075	0,2333
		KNN	0,5310 (+/- 0,08)	0,5040	0,4993	0,4897
		SVM	0,5395 (+/- 0,09)	0,5267	0,5391	0,5179
	kkKiB	DT	0,4556 (+/- 0,09)	0,5022	0,5034	0,5011
		NB	**0,4340 (+/- 0,10)**	0,3877	0,4326	0,3345
		KNN	0,5515 (+/- 0,08)	0,5465	0,5593	0,5488
		SVM	**0,6110 (+/- 0,09)**	0,5930	0,6105	0,5974
UKMD	KiB	DT	**0,9160 (+/- 0,07)**	0,9155	0,9156	0,9155
		NB	0,8513 (+/- 0,08)	0,8653	0,8586	0,8593
		KNN	0,5613 (+/- 0,13)	0,5846	0,5732	0,5743
		SVM	0,7078 (+/- 0,18)	0,7879	0,8525	0,8189
	dKiB	DT	0,8592 (+/- 0,11)	0,8704	0,8685	0,8690
		NB	**0,8563 (+/- 0,08)**	0,8711	0,8635	0,8639
		KNN	**0,6921 (+/- 0,19)**	0,7447	0,7000	0,7216
		SVM	**0,8585 (+/- 0,08)**	0,8856	0,8766	0,8785
	sKiB	DT	0,6685 (+/- 0,19)	0,6972	0,6700	0,6241
		NB	0,5693 (+/- 0,17)	0,5559	0,5658	0,4838
		KNN	0,6712 (+/- 0,20)	0,5327	0,5324	0,5324
		SVM	0,6813 (+/- 0,21)	0,7058	0,6725	0,6226
	kkKiB	DT	0,8989 (+/- 0,08)	0,9061	0,9057	0,9058
		NB	0,8559 (+/- 0,08)	0,8696	0,8635	0,8643
		KNN	0,6852 (+/- 0,09)	0,6978	0,7519	0,7239
		SVM	0,8538 (+/- 0,10)	0,8472	0,9457	0,8938

Tablo 4.1. (Devamı)

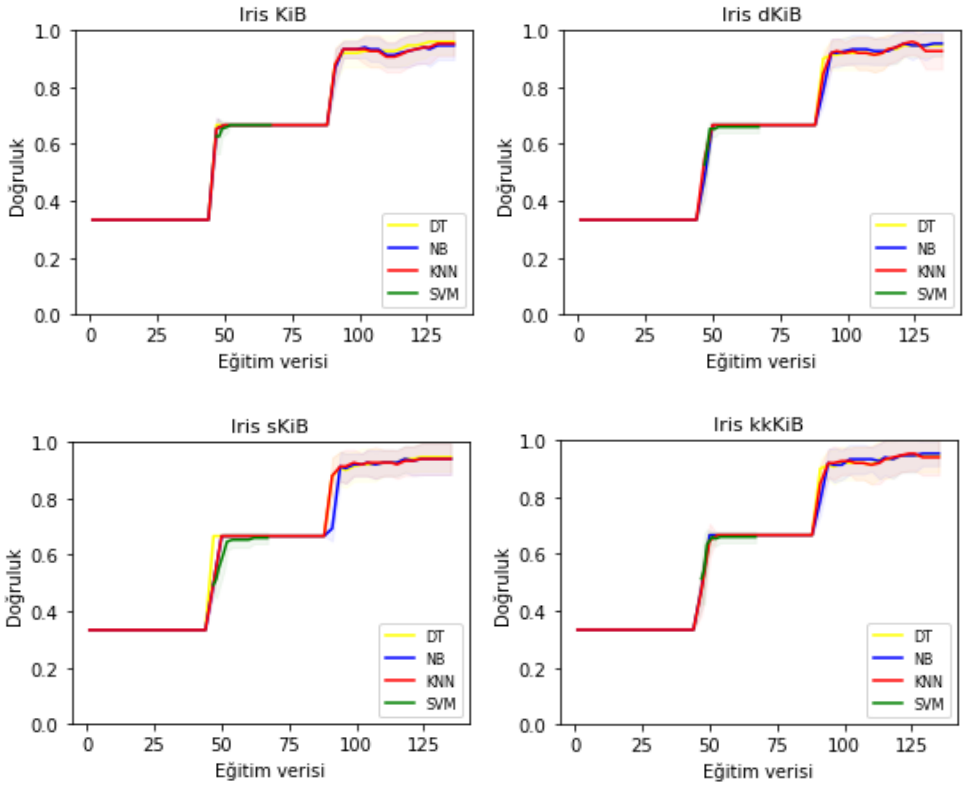
Occupancy	KiB	DT	0,9925 (+/- 0,00)	0,9829	0,9815	0,9822
		NB	0,9804 (+/- 0,01)	0,9173	0,9966	0,9553
		KNN	0,9934 (+/- 0,00)	0,9816	0,9873	0,9844
		SVM	**0,9939 (+/- 0,00)**	0,9765	0,9951	0,9857
	dKiB	DT	**0,9942 (+/- 0,00)**	0,9779	0,9946	0,9862
		NB	**0,9938 (+/- 0,00)**	0,9779	0,9932	0,9855
		KNN	0,9884 (+/- 0,02)	0,9821	0,9624	0,9721
		SVM	0,9938 (+/- 0,00)	0,9779	0,9932	0,9855
	sKiB	DT	0,9937 (+/- 0,00)	0,9770	0,9937	0,9852
		NB	0,9603 (+/- 0,02)	0,8437	0,9956	0,9134
		KNN	**0,9936 (+/- 0,00)**	0,9779	0,9922	0,9850
		SVM	0,9938 (+/- 0,00)	0,9779	0,9932	0,9855
	kkKiB	DT	0,9942 (+/- 0,00)	0,9835	0,9888	0,9861
		NB	0,9843 (+/- 0,01)	0,9345	0,9951	0,9638
		KNN	0,9922 (+/- 0,00)	0,9805	0,9824	0,9815
		SVM	0,9937 (+/- 0,00)	0,9765	0,9941	0,9852
Wilt	KiB	DT	**0,9771 (+/- 0,02)**	0,8409	0,7088	0,7692
		NB	**0,9595 (+/- 0,01)**	0,8736	0,2912	0,4368
		KNN	0,9690 (+/- 0,01)	0,7681	0,6092	0,6795
		SVM	0,9740 (+/- 0,01)	0,9412	0,5517	0,6957
	dKiB	DT	0,9665 (+/- 0,01)	0,8779	0,4406	0,5867
		NB	0,9585 (+/- 0,01)	0,7632	0,3333	0,4640
		KNN	0,9572 (+/- 0,02)	0,5985	0,6284	0,6131
		SVM	0,9671 (+/- 0,01)	0,8923	0,4444	0,5934
	sKiB	DT	0,9523 (+/- 0,02)	0,5824	0,4061	0,4786
		NB	0,7320 (+/- 0,04)	0,1357	0,7395	0,2294
		KNN	0,9532 (+/- 0,22)	0,1863	0,7318	0,2970
		SVM	0,9527 (+/- 0,02)	0,5899	0,4023	0,4784
	kkKiB	DT	0,9764 (+/- 0,01)	0,8101	0,7356	0,7711
		NB	0,9483 (+/- 0,00)	1,0000	0,0421	0,0809
		KNN	**0,9719 (+/- 0,01)**	0,8238	0,6092	0,7004
		SVM	**0,9802 (+/- 0,01)**	0,9365	0,6782	0,7867

Tablo 4.1. (Devamı)

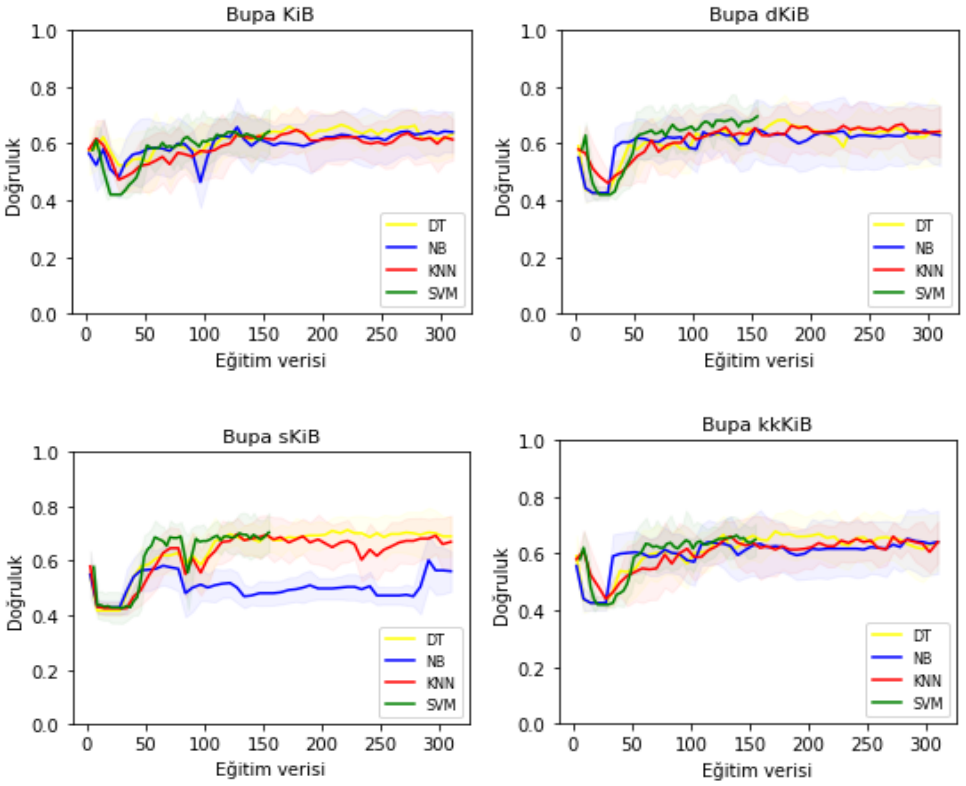
Wifi	KiB	DT	**0,9735 (+/- 0,02)**	0,9735	0,9735	0,9735
		NB	0,9750 (+/- 0,02)	0,9759	0,9750	0,9751
		KNN	**0,9820 (+/- 0,02)**	0,9822	0,9820	0,9820
		SVM	**0,9850 (+/- 0,01)**	0,9851	0,9850	0,9850
	dKiB	DT	0,9670 (+/- 0,03)	0,9669	0,9670	0,9670
		NB	0,9710 (+/- 0,02)	0,9724	0,9710	0,9712
		KNN	0,9720 (+/- 0,03)	0,9724	0,9720	0,9720
		SVM	0,9805 (+/- 0,02)	0,9806	0,9805	0,9805
	sKiB	DT	0,9620 (+/- 0,03)	0,9620	0,9620	0,9620
		NB	0,9675 (+/- 0,02)	0,9683	0,9675	0,9676
		KNN	0,9615 (+/- 0,02)	0,9617	0,9615	0,9615
		SVM	0,9715 (+/- 0,02)	0,9716	0,9715	0,9715
	kkKiB	DT	0,9690 (+/- 0,01)	0,9690	0,9690	0,9690
		NB	**0,9760 (+/- 0,02)**	0,9771	0,9760	0,9761
		KNN	0,9815 (+/- 0,01)	0,9816	0,9815	0,9815
		SVM	0,9825 (+/- 0,02)	0,9826	0,9825	0,9825

Tablo 4.1.'deki sonuçlar dikkate alındığında; Iris veri seti üzerinde KiB, dKiB ve kkKiB algoritmalarının, DT ve SVM sınıflayıcılar ile aynı sonuçları verdiği görülmektedir. Ancak dKiB ve kkKiB biraz daha düşük bir standart sapma ile bu sonuçları verdiği için daha başarılı kabul edilebilir. NB için KiB ve dKiB aynı sonucu vermektedir. KNN üzerinde ise KiB, daha iyi sonuç üretmektedir. Iris veri seti üzerinde bütün yöntemler, yaklaşık değer üretmekle birlikte; bütün sonuçlar göz önüne alındığında önerilen yöntemlerin orijinal KiB algortimasından daha başarılı olduğu söylenebilir.

Bupa veri seti için Tablo 4.1 incelendiğinde; bu veri seti üzerinde DT ile en iyi sonucu, sKiB yönteminin verdiği görülmektedir. NB ve KNN için en iyi sonuçlar, dKiB yöntemi ile elde edilmektedir. SVM için ise kkKiB yöntemi en başarılı sonucu bulmaktadır. Böylece Bupa veri seti üzerinde önerilen üç yöntemin de orijinal KiB yöntemine göre daha iyi sonuçlar verdiği ortaya çıkmaktadır. Iris ve Bupa veri setlerine ait başarım garfikleri sırasıyla Şekil 4.6. ve Şekil 4.7.'de verilmektedir.

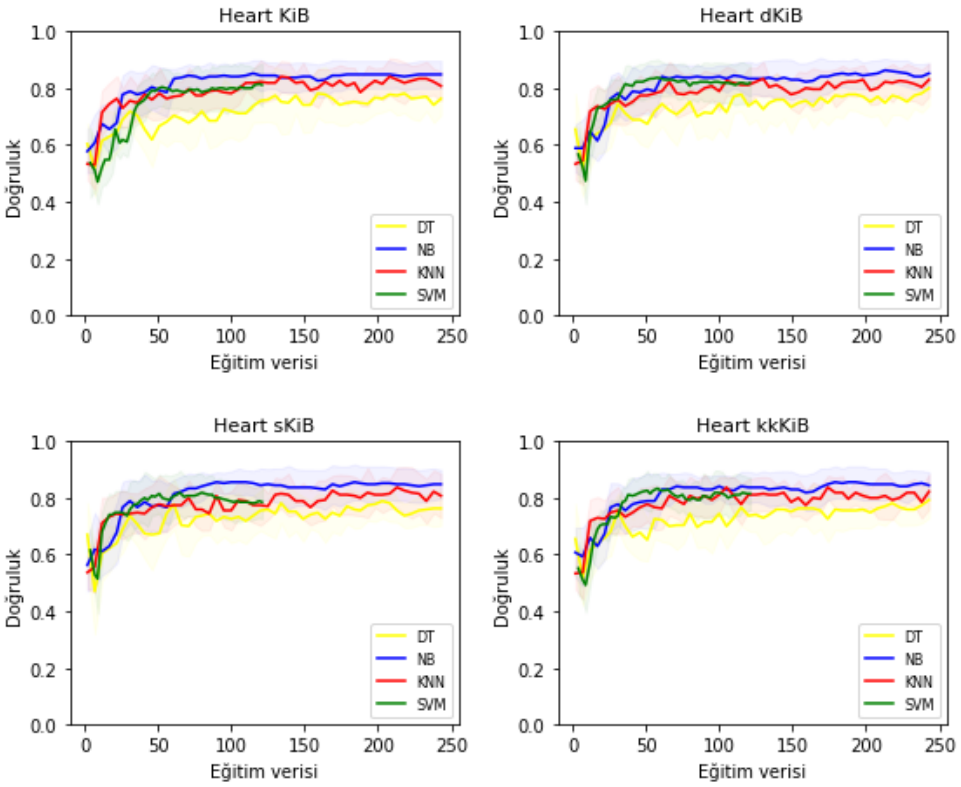


Şekil 4.6. Iris veri setinin KiB, dKiB, sKiB ve kkKiB için sınıflama başarım grafiği



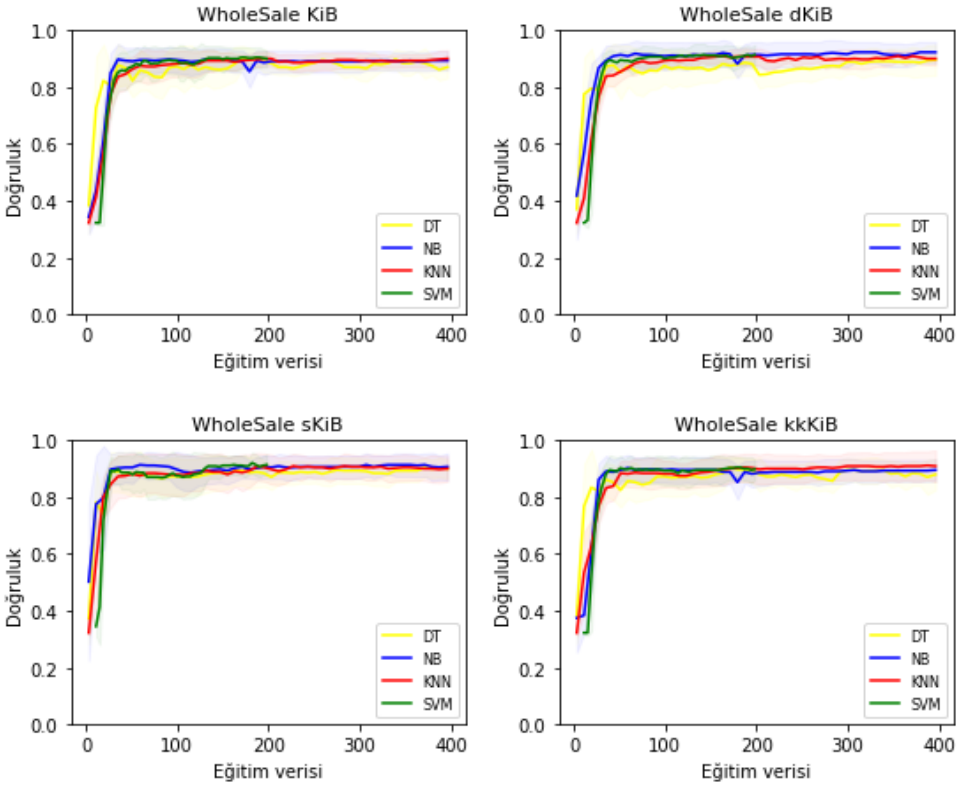
Şekil 4.7. Bupa veri setinin KiB, dKiB, sKiB ve kkKiB için sınıflama başarım grafiği

Heart veri seti üzerinde NB ile KiB, DT ile sKiB, KNN ile dKiB yöntemleri en iyi sonuçları vermektedir. SVM için ise en iyi sonuç kkKiB yöntemi ile elde edilmektedir. Bu ver seti üzerinde ayrıklaştırma yöntemleri için elde edilen sınflama sonuçları analiz edildiğinde nispeten yakın değerler ortaya çıktığını söylemek münkündür, ancak özellikle DT sınıflayıcı için diğer iki yöntem %70 bandında sonuç üretirken, kkKiB yöntemi için bu oran %73, sKiB yöntemi için ise bu oran %75'e kadar çıkmaktadır. Snıflama başarısı dikkate alındığında elde edilen bu artış oldukça önem arz etmektedir. Heart veri seti için sınıflama algoritmalarının başarım grafikleri Şekil 4.8.'de verilmektedir.

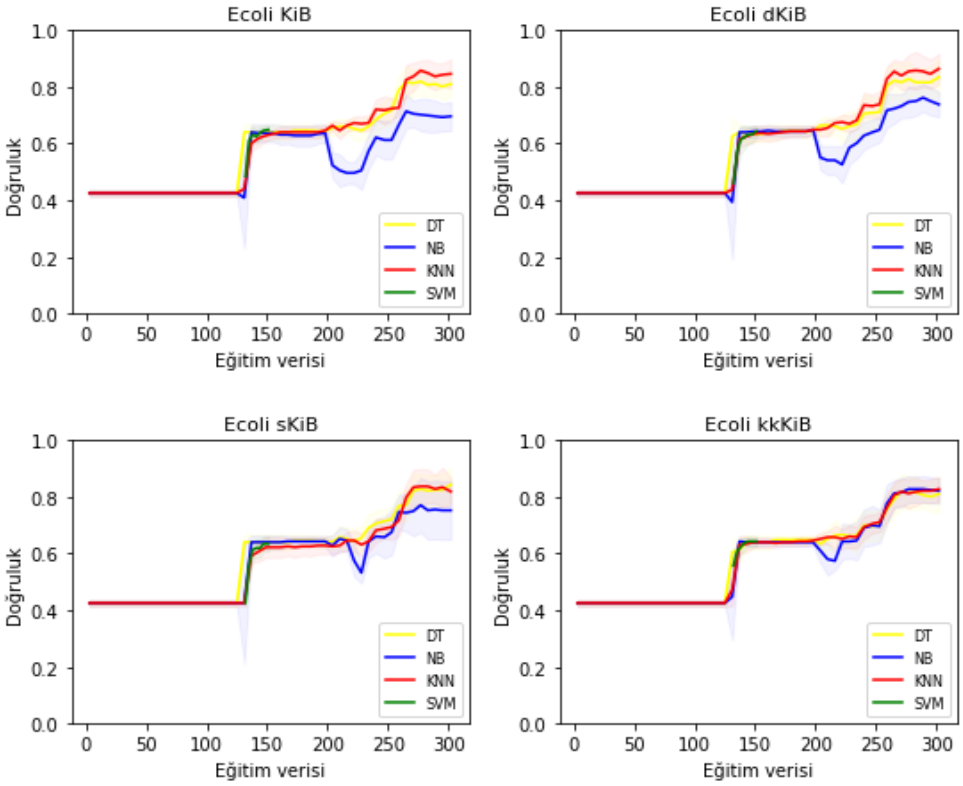


Şekil 4.8. Heart veri setinin KiB, dKiB, sKiB ve kkKiB için sınıflama başarım grafiği

WholeSale veri seti üzerinde KNN ile en iyi sonucu sKiB; NB, SVM ve DT ile en iyi sonuçları dKiB yöntemi vermektedir. Elde edilen sonuçlar önerilen yöntemlerin orijinal KiB algortimasıyla elde edilen sonuçlara benzer ve bazı algoritmalar için daha iyi sonuçlar ürettiğini ortaya koymaktadır. WholeSale veri seti üzerinde sınıflama algoritmalarının başarım grafikleri Şekil 4.9.'da verilmektedir.



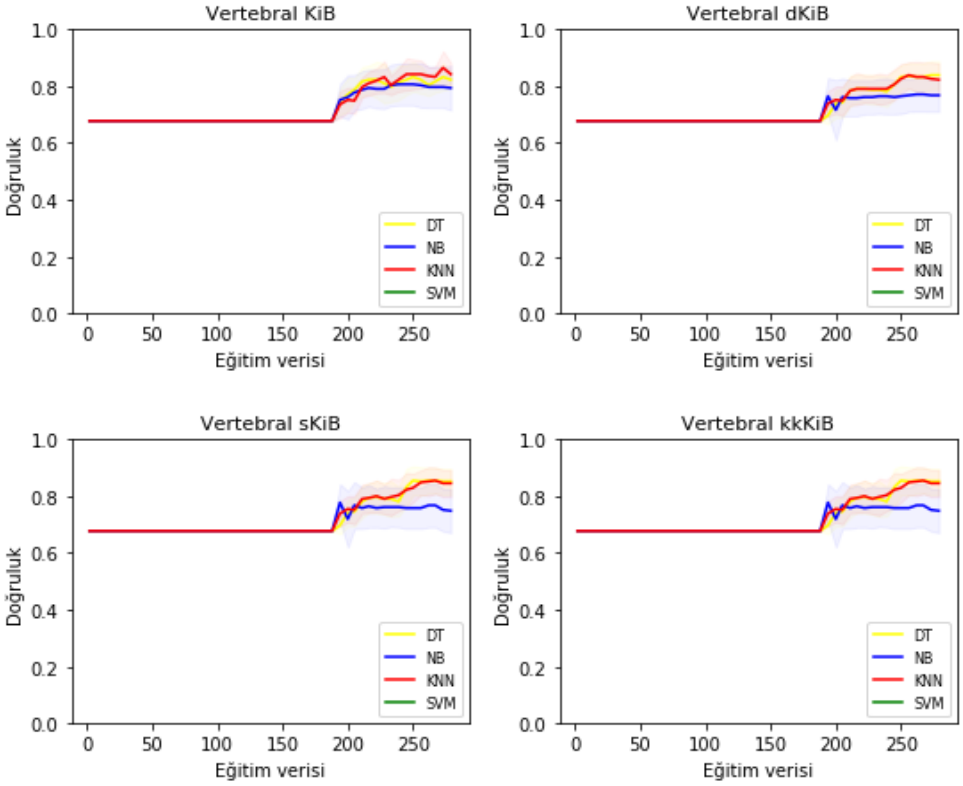
Şekil 4.9. WholeSale veri setinin KiB, dKiB, sKiB ve kkKiB için sınıflama başarım grafiği



Şekil 4.10. Ecoli veri setinin KiB, dKiB, sKiB ve kkKiB için sınıflama başarım grafiği

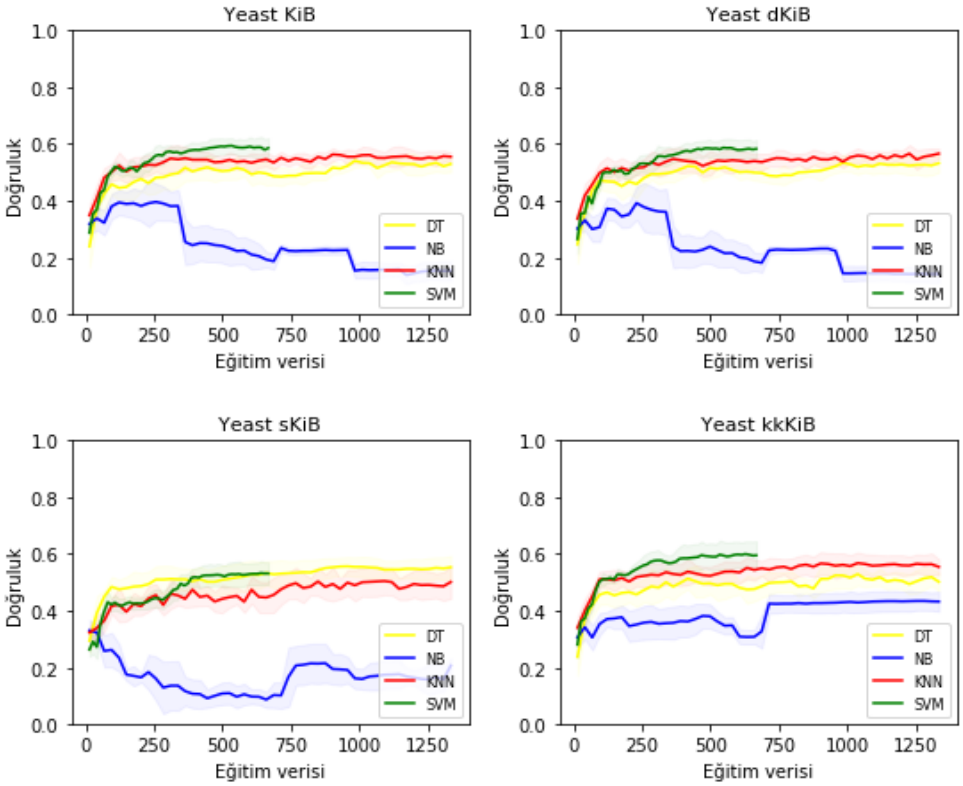
Tablo 4.1. incelendiğinde Ecoli veri seti için dKiB ayrıklaştırmanın, KNN ve SVM'de; sKiB ayrıklaştırmanın DT'de ve kkKiB ayrıklaştırmanın da NB'de daha iyi sonuç verdiği anlaşılmaktadır. Elde edilen veriler, bu veri seti üzerinde önerilen üç yöntemin de orijinal KiB algoritmasından daha başarılı olduğunu ortaya koymaktadır. Ecoli veri setine ait başarım grafikleri Şekil 4.10.'da verilmektedir.

Vertebral veri seti için kkKiB'in NB, KNN ve SVM'de; sKiB'in ise DT'de daha iyi sonuç verdiği görülmektedir. Bu veri seti için de önerilen yöntemlerin, özellikle de kkKiB'in orijinal KiB'den daha başarılı olduğunu söylemek mümkündür. Vertebral veri setine ait başarım grafikleri Şekil 4.11.'de verilmektedir.

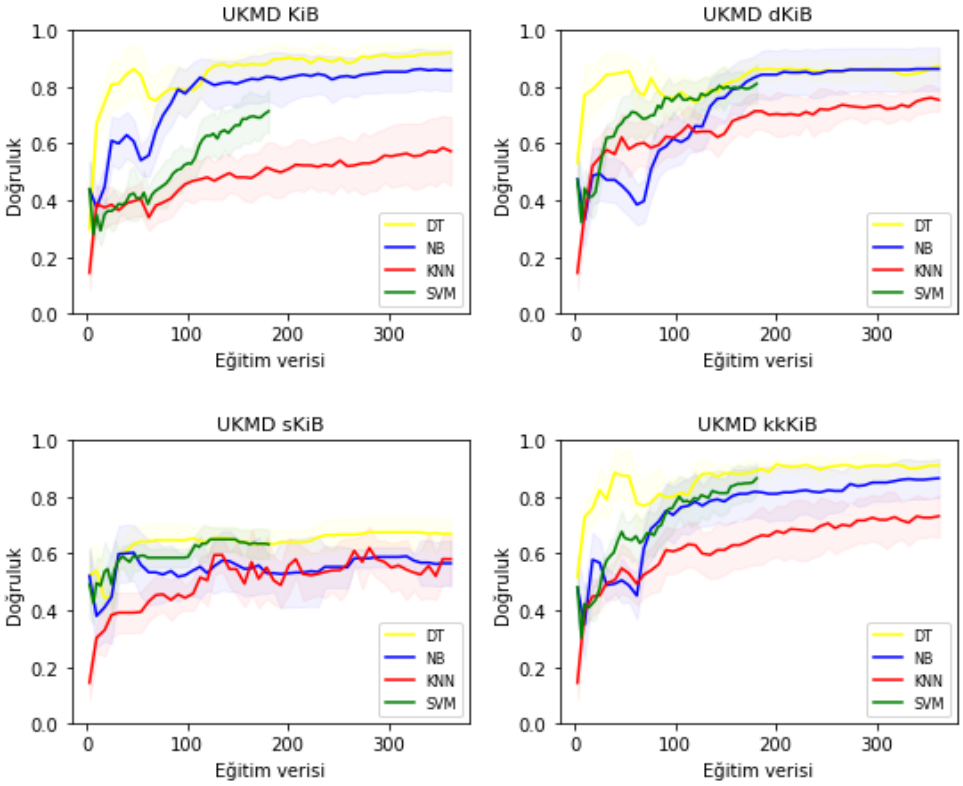


Şekil 4.11. Vertebral veri setinin KiB, dKiB, sKiB ve kkKiB için sınıflama başarım grafiği

Yeast veri seti için KNN sınıflayıcı KiB algoritmasıyla, SVM ve NB sınıflayıcılar kkKiB algoritmasıyla ve DT sınıflayıcı sKiB algoritmasıyla daha başarılı olmaktadır. Bu sonuçlar, önerilen yöntemlerin bu veri seti için genel olarak KiB'den daha iyi performans gösterdiğini ortaya koymaktadır. Yeast veri setine ait başarım grafikleri Şekil 4.12.'de verilmektedir.



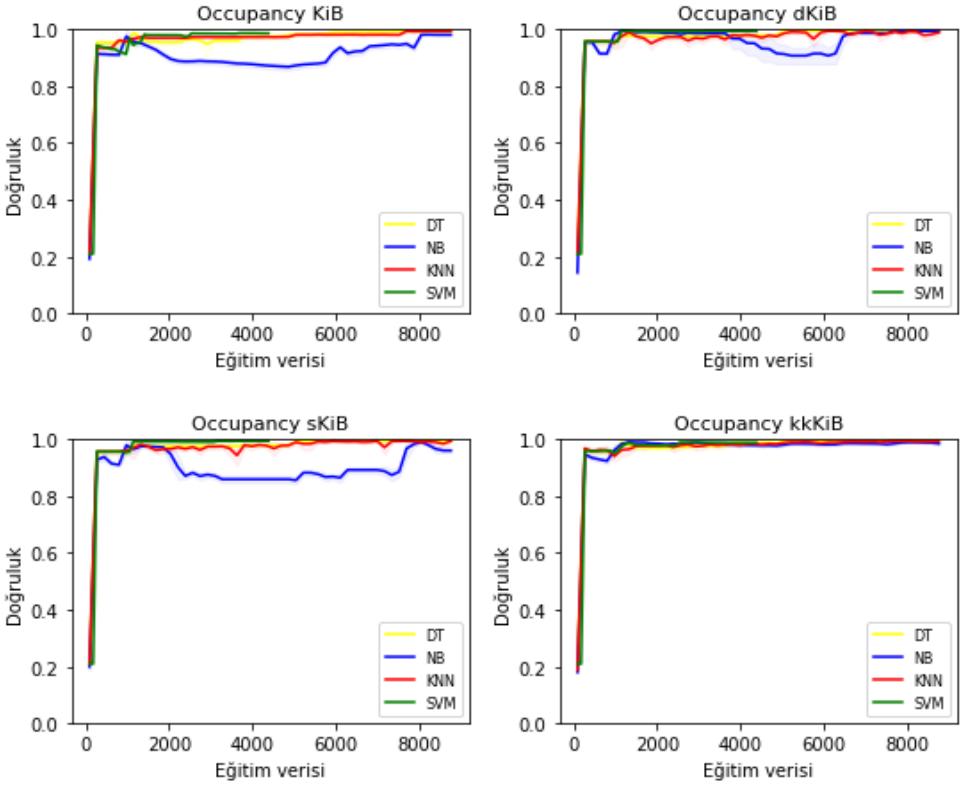
Şekil 4.12. Yeast veri setinin KiB, dKiB, sKiB ve kkKiB için sınıflama başarım grafiği



Şekil 4.13. UKMD veri setinin KiB, dKiB, sKiB ve kkKiB için sınıflama başarım grafiği

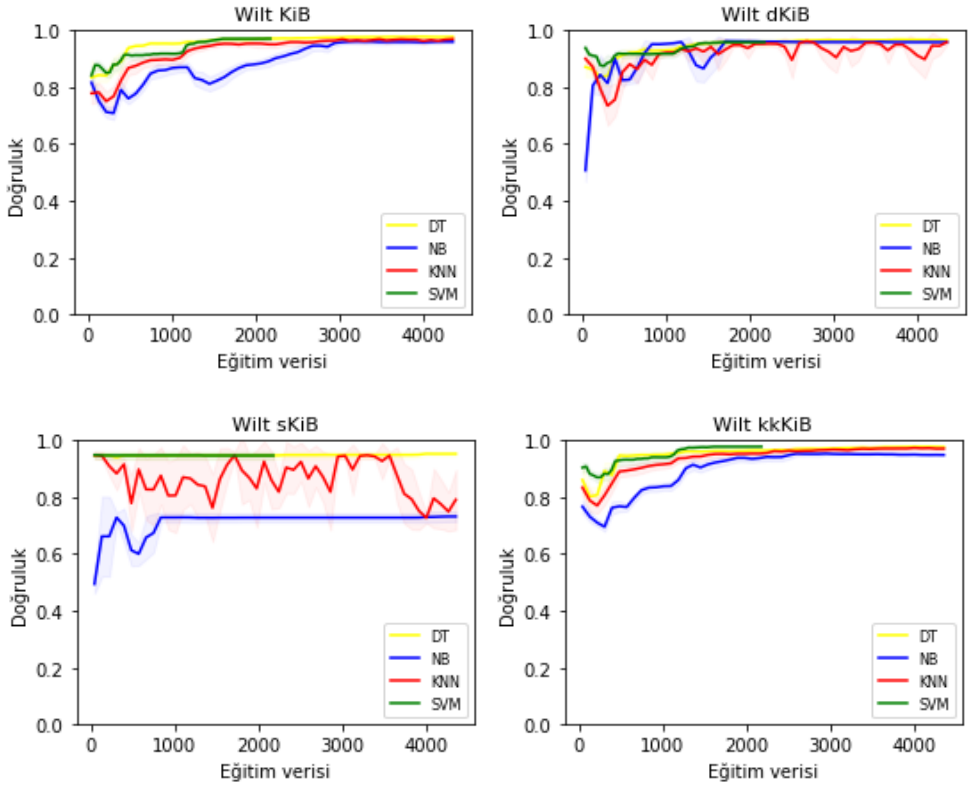
UKMD veri seti için, DT, KiB ile; NB, KNN ve SVM ise dKiB ile daha iyi sınıflama başarısı göstermektedir. Sonuçlar, UKMD veri seti için DT sınıflayıcı dışındaki diğer sınıflama algoritmalarının, önerilen dKiB yöntemi ile daha başarılı olduğunu ortaya koymaktadır. UKMD veri setine ait başarım grafikleri Şekil 4.13.'te verilmektedir.

Tablo 4.1. incelendiğinde; Occupancy veri seti üzerinde DT sınıflayıcı, dKiB ve kkKiB yöntemleri ile daha başarılı sınıflama yapmaktadır. Sınıflayıcının, bu iki yöntemle ayrıklaştırılan veri üzerindeki sınıflamaya ait standart sapmaları da eşit çıkmaktadır. NB için yine dKiB, KNN için sKiB ve SVM için KiB yöntemleri daha başarılı olmaktadır. Bu veri seti için de önerilen ayrıklaştırma yöntemlerinin sınıflama algoritmaları üzerinde daha olumlu sonuçlar ürettiğini söylemek mümkündür. Occupancy veri setine ait başarım grafikleri Şekil 4.14.'te verilmektedir.

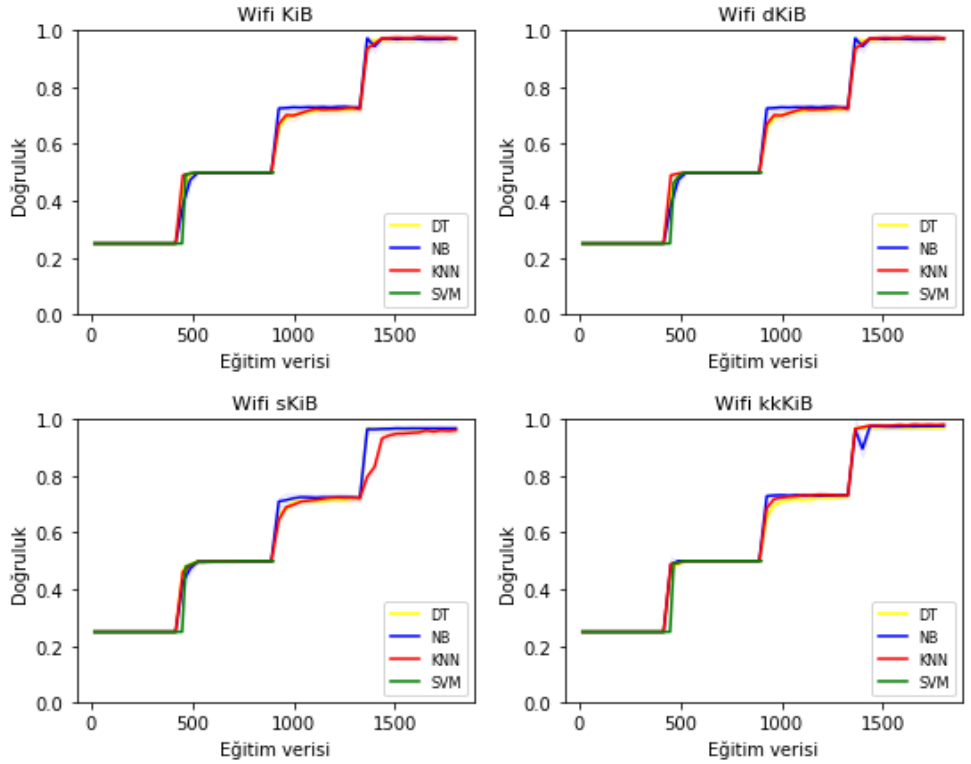


Şekil 4.14. Occupancy veri setinin KiB, dKiB, sKiB ve kkKiB için sınıflama başarım grafiği

Wilt veri seti üzerinde, DT ve NB sınıflayıcılar KiB algoritmasıyla; KNN ve SVM sınıflayıcılar ise kkKiB algoritmasıyla daha başarılı sınıflama performansı göstermektedir. Wilt veri setine ait başarım grafikleri Şekil 4.15.'te verilmektedir.



Şekil 4.15. Wilt veri setinin KiB, dKiB, sKiB ve kkKiB için sınıflama başarım grafiği



Şekil 4.16. Wifi veri setinin KiB, dKiB, sKiB ve kkKiB için sınıflama başarım grafiği

Wifi veri setine ait başarım grafikleri Şekil 4.16.'da verilmektedir. Wifi veri seti üzerinde NB, kkKiB ile daha iyi performans göstermektedir DT, KNN ve SVM ise KiB ile daha başarılı olmaktadır, ancak önerilen yöntemler de orijinal KiB algoritması ile çok yakın değerler elde etmektedir. Bundan dolayı önerilen yöntemlerin bu veri seti üzerinde başarısız olduğu sonucuna varılamaz.

Başarım sonuçları kesinlik, duyarlılık ve F_1-skor açısından değerlendirildiğinde ise genel tablo değişmemekle birlikte yöntemlerin bu metrikler açısından başarıları değişkenlik göstermektedir. Örneğin, Iris veri setinde dKiB ve KiB aynı doğruluk değerlerine sahip olduğu halde, hem kesinlik hem de duyarlılık değerleri açısından dKiB daha iyi performans göstermektedir. Heart veri setinde NB için KiB ayrıklaştırma, doğruluk açısından daha başrılı iken; sKiB kesinlik ve duyarlılık açısından daha başarılıdır.Yeast veri setinde KiB, KNN için daha doğru bir sınıflama başarısı gösterirken; dKiB daha yüksek kesinlik ve duyarlılık değerleri elde etmektedir. Occupancy veri setinde KiB, SVM sınıflayıcı için doğruluk açısından dKiB ve sKiB yöntemlerinden daha iyi sonuç elde ettiği halde, kesinlik açısından bu iki yöntem daha başarılı olmaktadır. Wilt veri seti için KiB, DT için doğruluk açısından kkKiB'den daha iyi sonuç ürettiği halde, duyarlılık açısından kkKiB daha başarılı olmaktadır. Bu veri seti üzerinde yöntemlerin duyarlılık kriteri açısından gösterdiği farklılık dikkate değer boyuttadır. Çünkü bu veri setinde pozitif sınıfa ait veri sayısı 261 tanedir. Veri setinin 4839 örnekten oluştuğu düşünüldüğünde bu, oldukça düşük bir sayıdır. DT için değelendirildiğinde KiB algoritmasının duyarlılık değeri %70.88 iken, kkKiB algoritmasının duyarlılık değeri %73.56 olmaktadır. NB için KiB yönteminin duyarlılık değeri %29.12, sKiB yönteminin duyarlılık değeri %73.95 çıkmıştır. Yine KNN için de sKiB yöntemi %73.18 ile KiB'den daha başarılı sonuç elde emektedir. KiB'in KNN için elde ettiği değer %60.92 olmaktadır. SVM için de kkKiB yöntemi %67.82 ile en başarılı sonucu elde etmektedir. Yöntemler, diğer veri setleri üzerinde, diğer metrik değerleri için sınıflama doğruluğuna paralel sonuçlar üretmektedir.

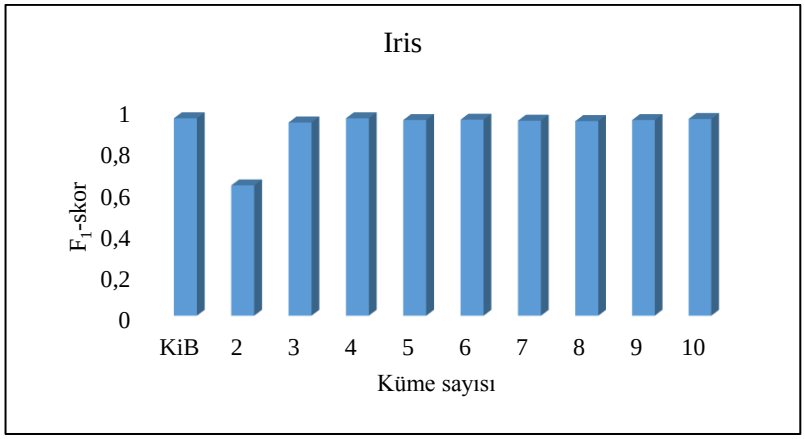
Tablo 4.1.'deki veriler bütün olarak ele alındığında; önerilen dKiB, sKiB ve kkKiB yöntemlerinin hem sınıflama doğruluğu, hem de kesinlik ve duyarlılık değerleri açısından genel itibariyle orijinal KiB algoritmasına göre daha başarılı oldukları söylenebilir.

Bu aşamadan itibaren önerilen diğer bir yöntem olan 2-10 arası kümelemenin, KiB algoritması ile karşılaştırmalı sonuçları veri setleri üzerinde ayrı ayrı incelenmektedir. Tablo 4.2. Iris veri setinin 2-10 kümeleme için sınıflama değerlerini göstermektedir.

Tablo 4.2. Iris veri setinin 2-10 kümeleme için sınıflama sonuçları

Küme Sayısı	Yöntem	Doğruluk	Kesinlik	Duyarlılık	F_1-skor
KiB	DT	0,9600 (+/- 0,09)	0,9600	0,9600	0,9600
	NB	0,9733 (+/- 0,07)	0,9534	0,9533	0,9533
	KNN	**0,9733 (+/- 0,07)**	0,9600	0,9600	0,9600
	SVM	0,9600 (+/- 0,09)	0,9471	0,9467	0,9466
2 Küme	DT	0,6867 (+/- 0,06)	0,6767	0,6733	0,6433
	NB	0,6867 (+/- 0,06)	0,6767	0,6733	0,6433
	KNN	0,6933 (+/- 0,07)	0,7849	0,6933	0,6197
	SVM	0,7000 (+/- 0,07)	0,7849	0,6933	0,6197
3 Küme	DT	0,9467 (+/- 0,12)	0,9401	0,9400	0,9400
	NB	0,9467 (+/- 0,10)	0,9338	0,9333	0,9333
	KNN	0,9467 (+/- 0,12)	0,9333	0,9333	0,9333
	SVM	0,9467 (+/- 0,12)	0,9333	0,9333	0,9333
4 Küme	DT	**0,9667 (+/- 0,09)**	0,9668	0,9667	0,9667
	NB	0,9533 (+/- 0,09)	0,9471	0,9467	0,9466
	KNN	0,9667 (+/- 0,07)	0,9679	0,9667	0,9668
	SVM	0,9600 (+/- 0,09)	0,9407	0,9400	0,9402
5 Küme	DT	0,9600 (+/- 0,07)	0,9471	0,9467	0,9466
	NB	**0,9733 (+/- 0,07)**	0,9738	0,9733	0,9733
	KNN	0,9333 (+/- 0,12)	0,9211	0,9200	0,9204
	SVM	**0,9667 (+/- 0,07)**	0,9477	0,9467	0,9468
6 Küme	DT	0,9533 (+/- 0,09)	0,9338	0,9333	0,9333
	NB	0,9667 (+/- 0,07)	0,9600	0,9600	0,9600
	KNN	0,9600 (+/- 0,07)	0,9547	0,9533	0,9536
	SVM	0,9600 (+/- 0,07)	0,9475	0,9467	0,9469
7 Küme	DT	0,9533 (+/- 0,12)	0,9276	0,9267	0,9266
	NB	0,9600 (+/- 0,09)	0,9534	0,9533	0,9533
	KNN	0,9533 (+/- 0,09)	0,9475	0,9467	0,9469
	SVM	0,9533 (+/- 0,09)	0,9475	0,9467	0,9469
8 Küme	DT	0,9467 (+/- 0,12)	0,9204	0,9200	0,9200
	NB	0,9533 (+/- 0,12)	0,9534	0,9533	0,9533
	KNN	0,9400 (+/- 0,11)	0,9332	0,9333	0,9332
	SVM	0,9600 (+/- 0,07)	0,9604	0,9600	0,9601
9 Küme	DT	0,9467 (+/- 0,12)	0,9268	0,9267	0,9267
	NB	0,9600 (+/- 0,09)	0,9600	0,9600	0,9600
	KNN	0,9467 (+/- 0,08)	0,9397	0,9400	0,9398
	SVM	0,9667 (+/- 0,07)	0,9610	0,9600	0,9600
10 Küme	DT	0,9533 (+/- 0,09)	0,9338	0,9333	0,9333
	NB	0,9600 (+/- 0,07)	0,9534	0,9533	0,9533
	KNN	0,9600 (+/- 0,07)	0,9604	0,9600	0,9601
	SVM	0,9600 (+/- 0,09)	0,9610	0,9600	0,9600

Tablo 4.2.'de görüldüğü gibi Iris verisi için, DT algoritması verinin 4 kümeye, SVM algoritması ise verinin 5 kümeye ayrıldığı durumda KiB'den daha iyi sonuç vermektedir. NB için KiB ve verinin 5 kümeye ayrıklaştırıldığı durum aynı sonucu vermektedir. KNN için KiB algoritması daha başarılı sonuç üretmektedir. Iris veri setinin 2-10 kümeleme için F_1-skor değerleri Şekil 4.17.'de görülmektedir.

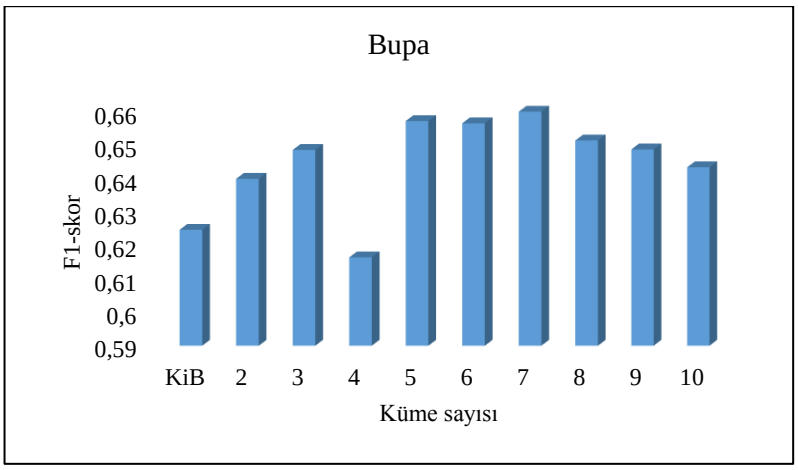


Şekil 4.17. Iris için 2-10 kümeleme F_1-skor değerlerinin KiB ile karşılaştırılması

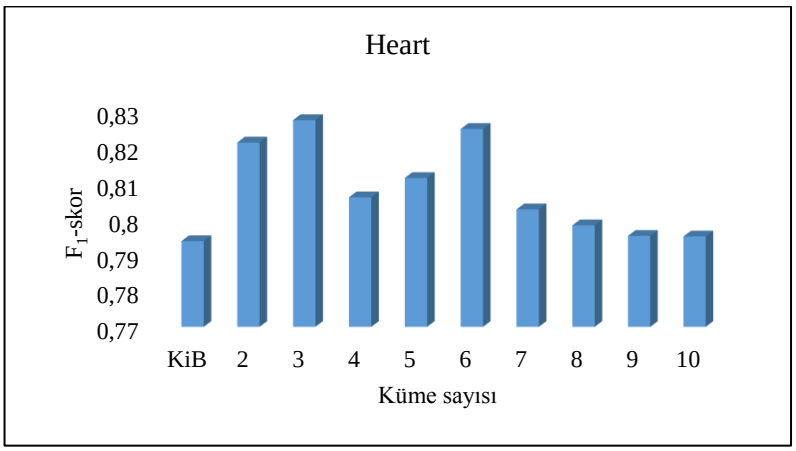
Tablo 4.3. Bupa veri setinin 2-10 kümeleme için sınıflama sonuçları

Küme Sayısı	Yöntem	Doğruluk	Kesinlik	Duyarlılık	F_1-skor
KiB	DT	0,6204 (+/- 0,18)	0,6434	0,6348	0,6370
	NB	0,6347 (+/- 0,16)	0,6495	0,6435	0,6453
	KNN	0,6290 (+/- 0,12)	0,6147	0,6116	0,6128
	SVM	0,6235 (+/- 0,13)	0,6391	0,6377	0,6031
2 Küme	DT	**0.6724 (+/- 0.15)**	0,6780	0,6812	0,6717
	NB	0,5850 (+/- 0,18)	0,6610	0,5797	0,5630
	KNN	0,6550 (+/- 0,11)	0,6317	0,6348	0,6327
	SVM	0,6988 (+/- 0,14)	0,7139	0,7072	0,6916
3 Küme	DT	0,6608 (+/- 0,15)	0,6791	0,6696	0,6716
	NB	0,6024 (+/- 0,22)	0,6520	0,5942	0,5866
	KNN	0,6641 (+/- 0,12)	0,6507	0,6552	0,6529
	SVM	0,6580 (+/- 0,20)	0,6833	0,6870	0,6823
4 Küme	DT	0,6375 (+/- 0,22)	0,6547	0,6435	0,6457
	NB	0,5361 (+/- 0,15)	0,6735	0,5333	0,4847
	KNN	0,6752 (+/- 0,15)	0,6652	0,6667	0,6658
	SVM	0,6755 (+/- 0,17)	0,6747	0,6783	0,6691
5 Küme	DT	0,6517 (+/- 0,18)	0,6941	0,6841	0,6860
	NB	0,5763 (+/- 0,13)	0,6217	0,5797	0,5763
	KNN	**0,6839 (+/- 0,10)**	0,6846	0,6870	0,6852
	SVM	**0,7015 (+/- 0,16)**	0,6877	0,6899	0,6803
6 Küme	DT	0,6141 (+/- 0,17)	0,6489	0,6377	0,6400
	NB	0,6231 (+/- 0,17)	0,6618	0,6348	0,6358
	KNN	0,6343 (+/- 0,13)	0,6472	0,6522	0,6470
	SVM	0,6958 (+/- 0,17)	0,7091	0,7101	0,7023
7 Küme	DT	0,6142 (+/- 0,17)	0,6529	0,6464	0,6483
	NB	0,6371 (+/- 0,18)	0,6548	0,6406	0,6429
	KNN	0,6836 (+/- 0,15)	0,6737	0,6754	0,6743
	SVM	0,6580 (+/- 0,08)	0,6817	0,6841	0,6735
8 Küme	DT	0,6286 (+/- 0,15)	0,6560	0,6464	0,6486
	NB	0,6229 (+/- 0,15)	0,6554	0,6551	0,6552
	KNN	0,6487 (+/- 0,17)	0,6472	0,6493	0,6480
	SVM	0,6466 (+/- 0,12)	0,6636	0,6667	0,6529
9 Küme	DT	0,6318 (+/- 0,18)	0,6444	0,6406	0,6420
	NB	0,6406 (+/- 0,09)	0,6548	0,6580	0,6556
	KNN	0,6406 (+/- 0,05)	0,6414	0,6464	0,6417
	SVM	0,6376 (+/- 0,16)	0,6628	0,6667	0,6547
10 Küme	DT	0,6292 (+/- 0,25)	0,6372	0,6372	0,6372
	NB	**0,6437 (+/- 0,10)**	0,6453	0,6493	0,6460
	KNN	0,6492 (+/- 0,09)	0,6532	0,6580	0,6528
	SVM	0,6490 (+/- 0,13)	0,6522	0,6551	0,6367

Tablo 4.3.'te Bupa veri seti için sınıflama değerleri verilmektedir. Tablo 4.3. incelendiğinde Bupa veri seti üzerinde, KiB yönteminin hiçbir sınıflama algoritması için en iyi performansı göstermediği rahatlıkla söylenebilir. Burada DT'nin 2 küme için, KNN ve SVM'nin 5 küme için, NB'nin de 10 küme için en iyi sonuçları verdiği görülmektedir. Bupa veri setinin 2-10 kümeleme için F_1-skor değerleri Şekil 4.18.'de görülmektedir.



Şekil 4.18. Bupa için 2-10 kümeleme F_1-skor değerlerinin KiB ile karşılaştırılması



Şekil 4.19. Heart için 2-10 kümeleme F_1-skor değerlerinin KiB ile karşılaştırılması

Heart veri seti için 2-10 kümeleme sınıflama sonuçları Tablo 4.4.'te verilmektedir. Tablo 4.4.'teki sonuçlar dikkate alındığında önerilen yöntemin, KiB algoritmasından daha iyi sonuçlar elde ettiği görülmektedir. Tablodan veriyi, DT için 6 kümeye; KNN için 3 kümeye; SVM için 4 kümeye; NB için 5 kümeye ayırmanın daha başarılı sonuçlar ürettiği sonucuna varılabilir. Heart veri setinin 2-10 kümeleme için F_1-skor değerleri Şekil 4.19.'da görülmektedir.

Tablo 4.4. Heart veri setinin 2-10 kümeleme için sınıflama sonuçları

Küme Sayısı	Yöntem	Doğruluk	Kesinlik	Duyarlılık	F_1-skor
KiB	DT	0,7074 (+/- 0,24)	0,7258	0,7500	0,7377
	NB	0,8407 (+/- 0,06)	0,8469	0,8450	0,8458
	KNN	0,8000 (+/- 0,09)	0,7647	0,7800	0,7723
	SVM	0,8148 (+/- 0,08)	0,8203	0,8192	0,8197
2 Küme	DT	0.7519 (+/- 0.16)	0,7887	0,7889	0,7888
	NB	0.8556 (+/- 0.13)	0,8443	0,8444	0,8443
	KNN	0.8259 (+/- 0.11)	0,8219	0,8222	0,8219
	SVM	0.8370 (+/- 0.12)	0,8300	0,8296	0,8298
3 Küme	DT	0.7593 (+/- 0.20)	0,7849	0,7852	0,7850
	NB	0.8407 (+/- 0.12)	0,8368	0,8370	0,8367
	KNN	**0.8481 (+/- 0.11)**	0,8446	0,8444	0,8438
	SVM	0.8370 (+/- 0.12)	0,8442	0,8444	0,8441
4 Küme	DT	0.7185 (+/- 0.21)	0,7258	0,7500	0,7377
	NB	0.8370 (+/- 0.11)	0,8405	0,8407	0,8404
	KNN	0.8296 (+/- 0.09)	0,8120	0,8111	0,8096
	SVM	**0.8370 (+/- 0.10)**	0,8369	0,8370	0,8365
5 Küme	DT	0.7000 (+/- 0.20)	0,7395	0,7333	0,7364
	NB	**0.8556 (+/- 0.10)**	0,8517	0,8519	0,8516
	KNN	0.8333 (+/- 0.07)	0,8186	0,8185	0,8176
	SVM	0.8333 (+/- 0.12)	0,8411	0,8407	0,8400
6 Küme	DT	**0.8000 (+/- 0.19)**	0,7996	0,8000	0,7994
	NB	0.8370 (+/- 0.08)	0,8518	0,8519	0,8514
	KNN	0.8148 (+/- 0.13)	0,8169	0,8148	0,8129
	SVM	0.8333 (+/- 0.10)	0,8376	0,8370	0,8361
7 Küme	DT	0.7259 (+/- 0.23)	0,7739	0,7417	0,7574
	NB	0.8370 (+/- 0.05)	0,8480	0,8481	0,8478
	KNN	0.7963 (+/- 0.13)	0,7847	0,7852	0,7845
	SVM	0.8074 (+/- 0.12)	0,8225	0,8222	0,8212
8 Küme	DT	0.7407 (+/- 0.27)	0,7565	0,7250	0,7404
	NB	0.8296 (+/- 0.07)	0,8407	0,8407	0,8402
	KNN	0.7815 (+/- 0.16)	0,7925	0,7926	0,7914
	SVM	0.8185 (+/- 0.12)	0,8225	0,8222	0,8212
9 Küme	DT	0.7407 (+/- 0.27)	0,7699	0,7250	0,7468
	NB	0.8333 (+/- 0.08)	0,8480	0,8481	0,8478
	KNN	0.7778 (+/- 0.17)	0,7736	0,7741	0,7730
	SVM	0.8074 (+/- 0.12)	0,8150	0,8148	0,8138
10 Küme	DT	0.7481 (+/- 0.25)	0,7586	0,7333	0,7458
	NB	0.8370 (+/- 0.09)	0,8407	0,8407	0,8402
	KNN	0.7815 (+/- 0.14)	0,7772	0,7778	0,7771
	SVM	0.8037 (+/- 0.14)	0,8186	0,8185	0,8176

WholeSale veri seti için 2-10 kümeleme sınıflama sonuçları Tablo 4.5.'te verilmektedir. Tablo 4.5.'teki veriler incelendiğinde, WholeSale veri seti için 2-10 kümeleme yaklaşımı kullanılarak yapılan ayrıklaştırma işleminin, KiB algoritmasından daha iyi sonuçlar verdiği görülmektedir. NB ve SVM algoritmaları 6 küme için KiB'den daha başarılı sonuçlar elde ederken, DT 7 küme için, KNN ise 10 küme için yine KiB'den daha başarılı sonuçlar elde etmektedir. Ayrıca 2-10 kümelema ayrıklaştırmasının KiB'e göre sınıflama sınıflama başarısını arttırması da dikkate değer bir olgudur. Örneğin NB için KiB %70 başarı sağlarken bu oran önerilen yöntemde %80 olmaktadır. WholeSale veri setinin 2-10 kümeleme için F_1-skor değerleri Şekil 4.20.'de görülmektedir.

Tablo 4.5. WholeSale veri setinin 2-10 kümeleme için sınıflama sonuçları

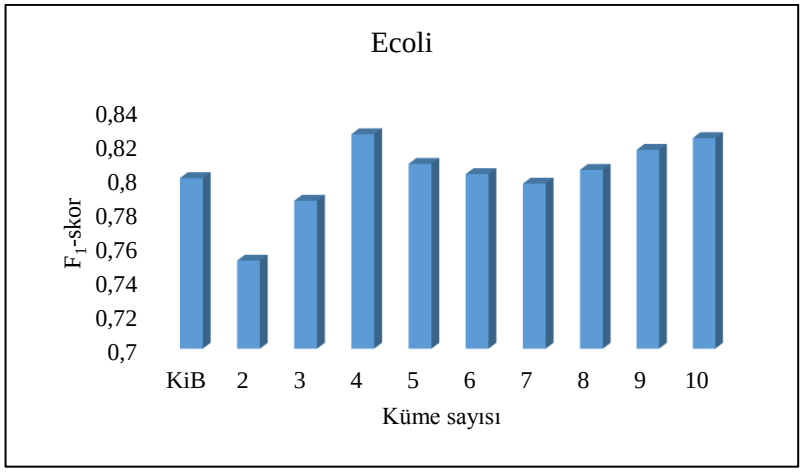
Küme Sayısı	Yöntem	Doğruluk	Kesinlik	Duyarlılık	F_1-skor
KiB	DT	0,8729 (+/- 0,10)	0,8712	0,8705	0,8708
	NB	0,8975 (+/- 0,11)	0,8927	0,8932	0,8929
	KNN	0,9042 (+/- 0,12)	0,8997	0,9000	0,8998
	SVM	0,6819 (+/- 0,02)	0,9067	0,9068	0,9051
2 Küme	DT	0,8750 (+/- 0,10)	0,8796	0,8773	0,8781
	NB	0,9090 (+/- 0,11)	0,9124	0,9091	0,9100
	KNN	0,9112 (+/- 0,12)	0,9150	0,9114	0,9123
	SVM	0,9113 (+/- 0,13)	0,9220	0,9159	0,9172
3 Küme	DT	0,8931 (+/- 0,10)	0,9014	0,9000	0,9005
	NB	0,9112 (+/- 0,09)	0,9182	0,9182	0,9182
	KNN	0,9089 (+/- 0,10)	0,9054	0,9045	0,9049
	SVM	0,9112 (+/- 0,13)	0,9161	0,9136	0,9144
4 Küme	DT	0,8818 (+/- 0,09)	0,8959	0,8955	0,8956
	NB	0,9090 (+/- 0,13)	0,9140	0,9136	0,9138
	KNN	0,9112 (+/- 0,12)	0,9225	0,9227	0,9226
	SVM	0,9136 (+/- 0,11)	0,9194	0,9182	0,9186
5 Küme	DT	0,8932 (+/- 0,08)	0,8934	0,8932	0,8933
	NB	0,9180 (+/- 0,12)	0,9251	0,9227	0,9234
	KNN	0,8885 (+/- 0,11)	0,8902	0,8909	0,8905
	SVM	0,6796 (+/- 0,03)	0,4587	0,6773	0,5470
6 Küme	DT	0,8955 (+/- 0,12)	0,9049	0,9045	0,9047
	NB	**0,9227 (+/- 0,11)**	0,9336	0,9273	0,9284
	KNN	0,9022 (+/- 0,11)	0,9239	0,9227	0,9231
	SVM	**0,9227 (+/- 0,12)**	0,9258	0,9227	0,9235
7 Küme	DT	**0,9068 (+/- 0,12)**	0,8934	0,8932	0,8933
	NB	0,8134 (+/- 0,11)	0,8634	0,8386	0,8219
	KNN	0,8978 (+/- 0,11)	0,9144	0,9136	0,9139
	SVM	0,6952 (+/- 0,13)	0,7867	0,6886	0,5726
8 Küme	DT	0,8980 (+/- 0,11)	0,8881	0,8886	0,8883
	NB	0,9181 (+/- 0,12)	0,9230	0,9159	0,9173
	KNN	0,8930 (+/- 0,14)	0,9016	0,8977	0,8988
	SVM	0,9182 (+/- 0,12)	0,9084	0,9091	0,9082
9 Küme	DT	0,9000 (+/- 0,12)	0,8905	0,8886	0,8893
	NB	0,9112 (+/- 0,09)	0,9158	0,9114	0,9124
	KNN	0,9067 (+/- 0,12)	0,9008	0,8977	0,8987
	SVM	0,9180 (+/- 0,13)	0,9062	0,9068	0,9056
10 Küme	DT	0,8774 (+/- 0,12)	0,8956	0,8932	0,8940
	NB	0,9135 (+/- 0,11)	0,9158	0,9114	0,9124
	KNN	**0,9113 (+/- 0,11)**	0,9079	0,9068	0,9072
	SVM	0,9226 (+/- 0,11)	0,9037	0,9045	0,9036

Şekil 4.20. WholeSale için 2-10 kümeleme F_1-skor değerlerinin KiB ile karşılaştırılması

Tablo 4.6. Ecoli veri setinin 2-10 kümeleme için sınıflama sonuçları

Küme Sayısı	Yöntem	Doğruluk	Kesinlik	Duyarlılık	F_1-skor
KiB	DT	0,8024 (+/- 0,11)	0,8041	0,8095	0,8058
	NB	0,6821 (+/- 0,10)	0,7531	0,6964	0,7067
	KNN	0,8481 (+/- 0,04)	0,8318	0,8452	0,8351
	SVM	0,8589 (+/- 0,08)	0,8402	0,8631	0,8512
2 Küme	DT	0,8366 (+/- 0,05)	0,8370	0,8333	0,8277
	NB	0,6725 (+/- 0,16)	0,7750	0,6577	0,7029
	KNN	0,7162 (+/- 0,13)	0,8234	0,7232	0,6527
	SVM	0,8249 (+/- 0,05)	0,8295	0,8274	0,8229
3 Küme	DT	0,8199 (+/- 0,12)	0,8049	0,8125	0,8064
	NB	0,6196 (+/- 0,20)	0,7981	0,6280	0,6729
	KNN	0,8359 (+/- 0,10)	0,8132	0,8214	0,8145
	SVM	0,8631 (+/- 0,11)	0,8552	0,8631	0,8523
4 Küme	DT	0,8309 (+/- 0,07)	0,8260	0,8333	0,8277
	NB	0,7391 (+/- 0,10)	0,8086	0,7381	0,7624
	KNN	0,8758 (+/- 0,08)	0,8520	0,8631	0,8552
	SVM	0,8763 (+/- 0,08)	0,8563	0,8661	0,8558
5 Küme	DT	0,7991 (+/- 0,10)	0,8038	0,8125	0,8069
	NB	0,7052 (+/- 0,14)	0,7776	0,7054	0,7243
	KNN	0,8668 (+/- 0,08)	0,8449	0,8542	0,8476
	SVM	0,8676 (+/- 0,09)	0,8489	0,8601	0,8533
6 Küme	DT	0,7928 (+/- 0,12)	0,7860	0,7887	0,7859
	NB	0,6873 (+/- 0,13)	0,7729	0,7054	0,7170
	KNN	0,8607 (+/- 0,07)	0,8616	0,8720	0,8657
	SVM	0,8525 (+/- 0,09)	0,8288	0,8512	0,8396
7 Küme	DT	0,7921 (+/- 0,09)	0,7732	0,7738	0,7709
	NB	0,6892 (+/- 0,14)	0,7554	0,7054	0,7140
	KNN	0,8580 (+/- 0,07)	0,8438	0,8542	0,8482
	SVM	**0,8773 (+/- 0,09)**	0,8482	0,8690	0,8529
8 Küme	DT	0,7963 (+/- 0,10)	0,7954	0,8006	0,7964
	NB	0,7285 (+/- 0,11)	0,7490	0,7411	0,7306
	KNN	0,8551 (+/- 0,09)	0,8435	0,8571	0,8489
	SVM	0,8593 (+/- 0,10)	0,8317	0,8542	0,8421
9 Küme	DT	0,8337 (+/- 0,11)	0,8363	0,8423	0,8384
	NB	0,7200 (+/- 0,11)	0,7627	0,7321	0,7314
	KNN	**0,8799 (+/- 0,10)**	0,8477	0,8601	0,8520
	SVM	0,8635 (+/- 0,06)	0,8326	0,8542	0,8430
10 Küme	DT	**0,8418 (+/- 0,10)**	0,8389	0,8452	0,8407
	NB	**0,7422 (+/- 0,14)**	0,7892	0,7500	0,7600
	KNN	0,8461 (+/- 0,06)	0,8408	0,8542	0,8453
	SVM	0,8547 (+/- 0,06)	0,8364	0,8571	0,8462

Ecoli veri seti için 2-10 kümeleme sınıflama sonuçları Tablo 4.6.'da verilmektedir. Tablo Tablo 4.6. incelendiğinde, sınıflama başarısı açısından 2-10 kümeleme yapılarak elde edilen ayrık verinin, orijinal KiB algoritması ile elde edilen ayrık veriden daha iyi sonuçlar verdiği görülmektedir. SVM sınıflayıcı, veri seti 7 küme; KNN sınıflayıcı veri seti 9 küme; DT ve NB ise veri seti 10 küme olacak şekilde ayrıklaştırıldığında daha iyi sonuçlar üretmektedir. Ecoli veri setinin 2-10 kümeleme için F_1-skor değerleri Şekil 4.21.'de görülmektedir.

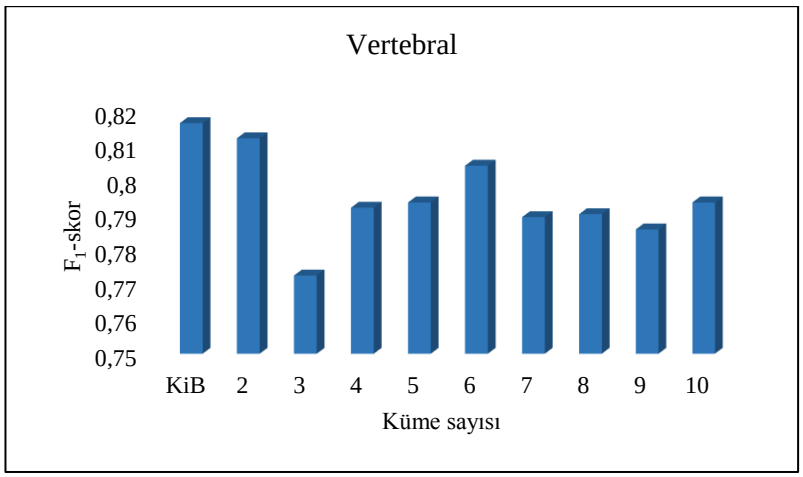


Şekil 4.21. Ecoli için 2-10 kümeleme F_1-skor değerlerinin KiB ile karşılaştırılması

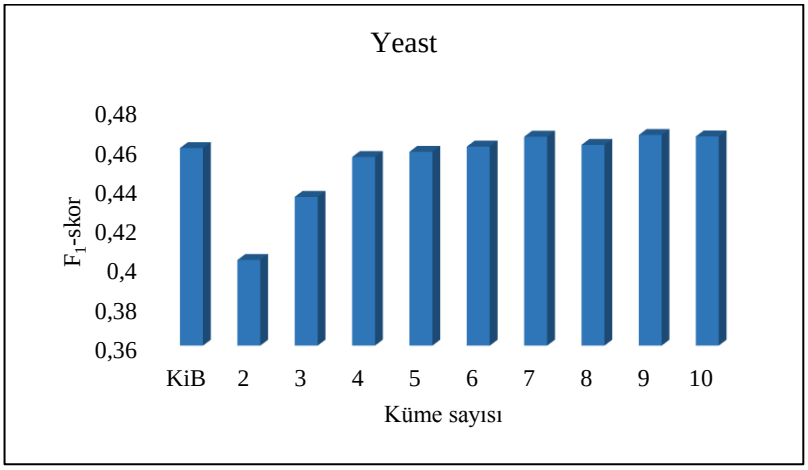
Tablo 4.7. Vertebral veri setinin 2-10 kümeleme için sınıflama sonuçları

Küme Sayısı	Yöntem	Doğruluk	Kesinlik	Duyarlılık	F_1-skor
KiB	DT	0,8032 (+/- 0,26)	0,8221	0,8226	0,8223
	NB	0,7935 (+/- 0,50)	0,8194	0,7935	0,7991
	KNN	**0,8194 (+/- 0,24)**	0,8214	0,8126	0,8167
	SVM	0,8161 (+/- 0,24)	0,8255	0,8305	0,8279
2 Küme	DT	0,8258 (+/- 0,41)	0,8531	0,8258	0,8306
	NB	**0,8065 (+/- 0,49)**	0,8296	0,8065	0,8114
	KNN	0,7968 (+/- 0,41)	0,8090	0,7968	0,8004
	SVM	0,8129 (+/- 0,46)	0,8415	0,8000	0,8062
3 Küme	DT	0,8129 (+/- 0,28)	0,8387	0,8387	0,8387
	NB	0,7613 (+/- 0,61)	0,8111	0,7581	0,7659
	KNN	0,7258 (+/- 0,12)	0,7262	0,7258	0,6756
	SVM	0,7806 (+/- 0,36)	0,8193	0,8065	0,8101
4 Küme	DT	0,7968 (+/- 0,29)	0,8155	0,8155	0,8155
	NB	0,7645 (+/- 0,60)	0,8290	0,7677	0,7753
	KNN	0,7839 (+/- 0,17)	0,7865	0,7871	0,7697
	SVM	0,8032 (+/- 0,31)	0,8149	0,8026	0,8082
5 Küme	DT	0,8323 (+/- 0,29)	0,8443	0,8387	0,8406
	NB	0,7548 (+/- 0,58)	0,8176	0,7581	0,7659
	KNN	0,7484 (+/- 0,22)	0,7746	0,7555	0,7633
	SVM	0,7968 (+/- 0,34)	0,8041	0,8057	0,8049
6 Küme	DT	**0,8419 (+/- 0,30)**	0,8598	0,8516	0,8539
	NB	0,7419 (+/- 0,58)	0,8048	0,7452	0,7535
	KNN	0,7710 (+/- 0,25)	0,7980	0,8032	0,7972
	SVM	0,7968 (+/- 0,34)	0,8181	0,8097	0,8124
7 Küme	DT	0,8032 (+/- 0,31)	0,8226	0,8100	0,8157
	NB	0,7516 (+/- 0,53)	0,8096	0,7613	0,7689
	KNN	0,7581 (+/- 0,17)	0,7768	0,7502	0,7604
	SVM	0,7968 (+/- 0,29)	0,8066	0,8212	0,8128
8 Küme	DT	0,8161 (+/- 0,31)	0,8302	0,8302	0,8302
	NB	0,7581 (+/- 0,53)	0,7518	0,7871	0,7507
	KNN	0,7581 (+/- 0,21)	0,7711	0,7505	0,7588
	SVM	0,7935 (+/- 0,28)	0,8178	0,8257	0,8215
9 Küme	DT	0,8065 (+/- 0,29)	0,8018	0,8032	0,8024
	NB	0,7645 (+/- 0,52)	0,7588	0,7943	0,7597
	KNN	0,7548 (+/- 0,28)	0,7710	0,7764	0,7735
	SVM	**0,8194 (+/- 0,25)**	0,8081	0,8081	0,8081
10 Küme	DT	0,7871 (+/- 0,30)	0,8009	0,8032	0,8019
	NB	0,7710 (+/- 0,51)	0,7653	0,8017	0,7665
	KNN	0,7774 (+/- 0,33)	0,8209	0,7774	0,7844
	SVM	0,8065 (+/- 0,31)	0,8213	0,8226	0,8219

Vertebral veri seti için 2-10 kümeleme sınıflama sonuçları Tablo 4.7.'de verilmektedir. Tablo 4.7. incelendiğinde; Vertebral veri seti için KNN'nin KiB algoritması, NB'nin 2 küme ayrıklaştırma, DT'nin 6 küme ayrıklaştırma ve son olarak SVM'nin 9 küme ayrıklaştırma ile daha başarılı olduğu görülmektedir. Vertebral veri setinin 2-10 kümeleme için F_1-skor değerleri Şekil 4.22.'de görülmektedir.



Şekil 4.22. Vertebral için 2-10 kümeleme F_1-skor değerlerinin KiB ile karşılaştırılması

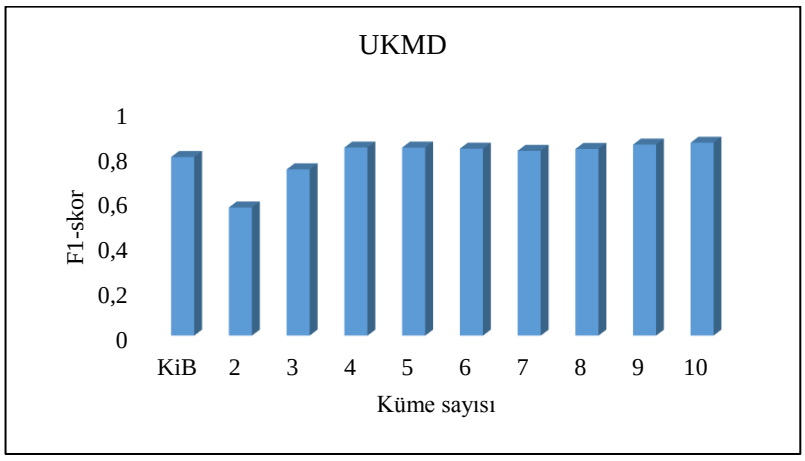


Şekil 4.23. Yeast için 2-10 kümeleme F_1-skor değerlerinin KiB ile karşılaştırılması

Yeast veri setinin 2-10 kümeleme için F_1-skor değerleri Şekil 4.23.'te görülmektedir. Yeast veri seti için 2-10 kümeleme sınıflama sonuçları Tablo 4.8.'de verilmektedir. Tablo 4.8. incelendiğinde; Yeast veri seti için KNN'nin KiB algoritması, DT'nin 2 küme ayrıklaştırma, NB'nin ve SVM'nin 7 küme ayrıklaştırma ile daha başarılı olduğu görülmektedir. Bu verilere göre 2-10 kmeleme yaklaşımının hem Vertebral hem de Yeast üzerinde, KNN dışındaki diğer algoritmalar için daha iyi sonuçlar elde ettiği söylenebilir.

Tablo 4.8. Yeast veri setinin 2-10 kümeleme için sınıflama sonuçları

Küme Sayısı	Yöntem	Doğruluk	Kesinlik	Duyarlılık	F_1-skor
KiB	DT	0,4765 (+/- 0,07)	0,5315	0,5303	0,5281
	NB	0,1537 (+/- 0,08)	0,4007	0,1543	0,1770
	KNN	**0,5656 (+/- 0,09)**	0,5452	0,5586	0,5476
	SVM	0,5988 (+/- 0,08)	0,5872	0,6024	0,5880
2 Küme	DT	**0,5367 (+/- 0,08)**	0,5535	0,5377	0,5298
	NB	0,0829 (+/- 0,09)	0,1689	0,0802	0,0685
	KNN	0,4740 (+/- 0,09)	0,4998	0,4906	0,4763
	SVM	0,5522 (+/- 0,10)	0,5523	0,5492	0,5391
3 Küme	DT	0,5204 (+/- 0,07)	0,5393	0,5357	0,5268
	NB	0,0924 (+/- 0,06)	0,2671	0,0876	0,0857
	KNN	0,5255 (+/- 0,08)	0,5477	0,5580	0,5476
	SVM	0,5832 (+/- 0,10)	0,5849	0,5863	0,5815
4 Küme	DT	0,5325 (+/- 0,08)	0,5518	0,5526	0,5461
	NB	0,1282 (+/- 0,10)	0,4097	0,1307	0,1476
	KNN	0,5355 (+/- 0,06)	0,5479	0,5633	0,5535
	SVM	0,5820 (+/- 0,10)	0,5736	0,5856	0,5754
5 Küme	DT	0,5010 (+/- 0,12)	0,5276	0,5317	0,5271
	NB	0,1422 (+/- 0,06)	0,4875	0,1429	0,1599
	KNN	0,5602 (+/- 0,10)	0,5480	0,5580	0,5480
	SVM	0,6040 (+/- 0,09)	0,5966	0,6078	0,5981
6 Küme	DT	0,4804 (+/- 0,09)	0,5130	0,5148	0,5124
	NB	0,1504 (+/- 0,06)	0,5032	0,1489	0,1650
	KNN	0,5434 (+/- 0,10)	0,5562	0,5667	0,5551
	SVM	0,6141 (+/- 0,09)	0,6103	0,6213	0,6113
7 Küme	DT	0,4764 (+/- 0,09)	0,5159	0,5155	0,5142
	NB	**0,1645 (+/- 0,09)**	0,6746	0,1658	0,1808
	KNN	0,5440 (+/- 0,10)	0,5662	0,5775	0,5653
	SVM	**0,6164 (+/- 0,10)**	0,6019	0,6132	0,6035
8 Küme	DT	0,4725 (+/- 0,08)	0,5108	0,5081	0,5078
	NB	0,1557 (+/- 0,07)	0,6376	0,1577	0,1703
	KNN	0,5540 (+/- 0,11)	0,5688	0,5775	0,5658
	SVM	0,6103 (+/- 0,10)	0,6035	0,6139	0,6033
9 Küme	DT	0,4730 (+/- 0,11)	0,5156	0,5162	0,5148
	NB	0,1617 (+/- 0,07)	0,4563	0,1678	0,1858
	KNN	0,5513 (+/- 0,10)	0,5666	0,5755	0,5642
	SVM	0,6081 (+/- 0,09)	0,6049	0,6125	0,6025
10 Küme	DT	0,4707 (+/- 0,08)	0,5161	0,5148	0,5149
	NB	0,1603 (+/- 0,06)	0,4811	0,1631	0,1815
	KNN	0,5653 (+/- 0,12)	0,5708	0,5815	0,5704
	SVM	0,5969 (+/- 0,08)	0,6052	0,6098	0,5976



Şekil 4.24. UKMD için 2-10 kümeleme F_1-skor değerlerinin KiB ile karşılaştırılması

Tablo 4.9. UKMD veri setinin 2-10 kümeleme için sınıflama sonuçları

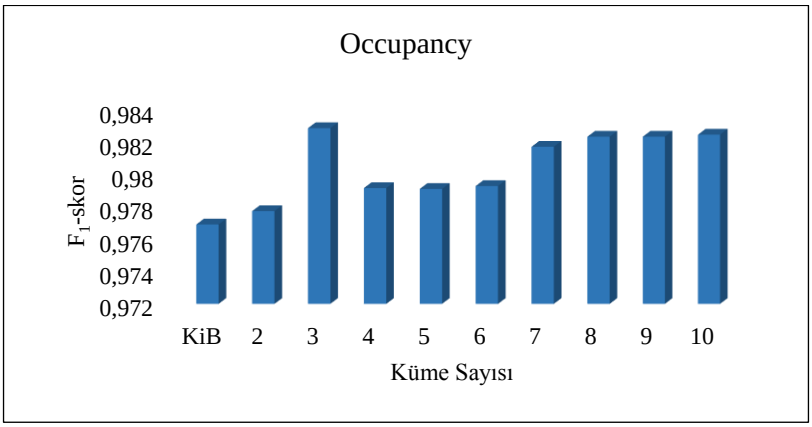
Küme Sayısı	Yöntem	Doğruluk	Kesinlik	Duyarlılık	F_1-skor
KiB	DT	**0,9160 (+/- 0,07)**	0,9155	0,9156	0,9155
	NB	0,8513 (+/- 0,08)	0,8653	0,8586	0,8593
	KNN	0,5613 (+/- 0,13)	0,5846	0,5732	0,5743
	SVM	0,7078 (+/- 0,18)	0,7879	0,8525	0,8189
2 Küme	DT	0,6685 (+/- 0,19)	0,6972	0,6700	0,6241
	NB	0,5693 (+/- 0,17)	0,5559	0,5658	0,4838
	KNN	0,5212 (+/- 0,20)	0,5574	0,5462	0,5423
	SVM	0,6813 (+/- 0,21)	0,7058	0,6725	0,6226
3 Küme	DT	0,7710 (+/- 0,14)	0,7533	0,7342	0,7364
	NB	0,7523 (+/- 0,17)	0,8318	0,7767	0,7338
	KNN	0,7062 (+/- 0,12)	0,7089	0,6928	0,6978
	SVM	0,8263 (+/- 0,06)	0,7503	0,8288	0,7805
4 Küme	DT	0,8810 (+/- 0,06)	0,9039	0,9007	0,9009
	NB	0,8460 (+/- 0,09)	0,8624	0,8536	0,8549
	KNN	**0,7667 (+/- 0,13)**	0,7692	0,6000	0,6742
	SVM	**0,9032 (+/- 0,09)**	0,9103	0,9057	0,9056
5 Küme	DT	0,8666 (+/- 0,09)	0,8835	0,8809	0,8816
	NB	0,8437 (+/- 0,08)	0,8601	0,8511	0,8524
	KNN	0,7192 (+/- 0,25)	0,7467	0,7172	0,7274
	SVM	0,8856 (+/- 0,09)	0,8812	0,8734	0,8728
6 Küme	DT	0,8592 (+/- 0,11)	0,8704	0,8685	0,8690
	NB	0,8563 (+/- 0,08)	0,8711	0,8635	0,8639
	KNN	0,6921 (+/- 0,19)	0,7805	0,6400	0,7033
	SVM	0,8585 (+/- 0,08)	0,8856	0,8766	0,8785
7 Küme	DT	0,8912 (+/- 0,07)	0,8914	0,8908	0,8908
	NB	0,8441 (+/- 0,10)	0,8661	0,8586	0,8590
	KNN	0,6769 (+/- 0,12)	0,7225	0,6798	0,6943
	SVM	0,8366 (+/- 0,09)	0,8696	0,8000	0,8333
8 Küme	DT	0,9062 (+/- 0,09)	0,9043	0,9032	0,9032
	NB	0,8341 (+/- 0,08)	0,8584	0,8511	0,8521
	KNN	0,6702 (+/- 0,10)	0,6617	0,7213	0,6902
	SVM	0,8520 (+/- 0,13)	0,8936	0,8400	0,8660
9 Küme	DT	0,8938 (+/- 0,07)	0,9038	0,9032	0,9032
	NB	0,8559 (+/- 0,08)	0,8647	0,8586	0,8595
	KNN	0,7322 (+/- 0,10)	0,7368	0,8033	0,7686
	SVM	0,8688 (+/- 0,10)	0,8750	0,8400	0,8571
10 Küme	DT	0,8964 (+/- 0,08)	0,9038	0,9032	0,9034
	NB	**0,8634 (+/- 0,08)**	0,8727	0,8660	0,8666
	KNN	0,7525 (+/- 0,13)	0,7812	0,8197	0,8000
	SVM	0,8664 (+/- 0,10)	0,9091	0,8000	0,8511

UKMD veri setinin 2-10 kümeleme için F_1-skor değerleri Şekil 4.24.'te; sınıflama sonuçları Tablo 4.9.'da verilmektedir. Tablo 4.9 incelendiğinde UKMD veri seti üzerinde DT sınıflayıcı, KiB algortiması ile; KNN ve SVM sınıflayıcılar, 4 küme ayrıklaştırma ile; NB sınıflayıcı ise 10 küme ayrıklaştırma ile daha başarılı sonuçlar üretmektedir.

Occupancy veri seti için 2-10 kümeleme sınıflama başarım sonuçları Tablo 4.10.'da verilmektedir. Tabloya göre KNN ve SVM sınıflayıcılar KiB ile, NB 2 küme ve DT 4 küme ayrıklaştırma ile daha başarılı olmaktadır. Occupancy veri setinin 2-10 kümeleme için F_1-skor değerleri Şekil 4.25.'te görülmektedir.

Tablo 4.10. Occupancy veri setinin 2-10 kümeleme için sınıflama sonuçları

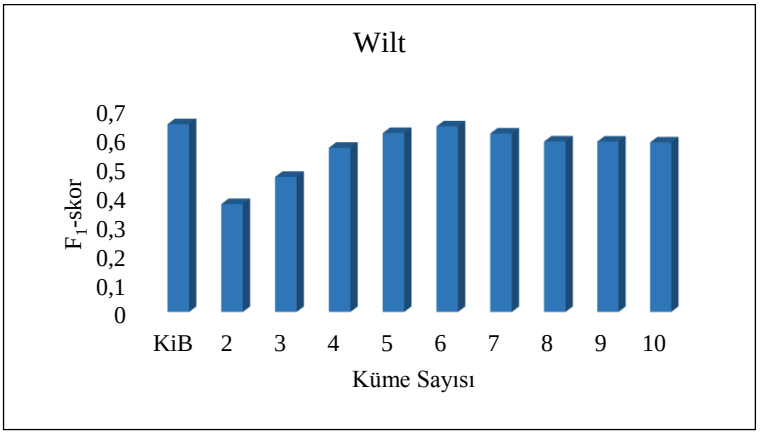
Küme Sayısı	Yöntem	Doğruluk	Kesinlik	Duyarlılık	F_1-skor
KiB	DT	0,9925 (+/- 0,00)	0,9829	0,9815	0,9822
	NB	0,9804 (+/- 0,01)	0,9173	0,9966	0,9553
	KNN	**0,9934 (+/- 0,00)**	0,9816	0,9873	0,9844
	SVM	**0,9939 (+/- 0,00)**	0,9765	0,9951	0,9857
2 Küme	DT	0,9938 (+/- 0,00)	0,9779	0,9932	0,9855
	NB	**0,9898 (+/- 0,01)**	0,9775	0,9741	0,9758
	KNN	0,9874 (+/- 0,01)	0,9820	0,9575	0,9696
	SVM	0,9938 (+/- 0,00)	0,9779	0,9932	0,9855
3 Küme	DT	0,9942 (+/- 0,00)	0,9779	0,9946	0,9862
	NB	0,9938 (+/- 0,00)	0,9779	0,9932	0,9855
	KNN	0,9884 (+/- 0,02)	0,9821	0,9624	0,9721
	SVM	0,9938 (+/- 0,00)	0,9779	0,9932	0,9855
4 Küme	DT	**0,9944 (+/- 0,00)**	0,9789	0,9946	0,9867
	NB	0,9627 (+/- 0,02)	0,8521	0,9951	0,9181
	KNN	0,9890 (+/- 0,02)	0,9821	0,9653	0,9737
	SVM	0,9938 (+/- 0,00)	0,9779	0,9932	0,9855
5 Küme	DT	0,9943 (+/- 0,00)	0,9789	0,9941	0,9864
	NB	0,9587 (+/- 0,02)	0,8370	0,9976	0,9103
	KNN	0,9882 (+/- 0,01)	0,9731	0,9707	0,9719
	SVM	0,9938 (+/- 0,00)	0,9779	0,9932	0,9855
6 Küme	DT	0,9938 (+/- 0,00)	0,9779	0,9932	0,9855
	NB	0,9586 (+/- 0,02)	0,8367	0,9976	0,9101
	KNN	0,9901 (+/- 0,01)	0,9780	0,9746	0,9763
	SVM	0,9936 (+/- 0,00)	0,9770	0,9932	0,9850
7 Küme	DT	0,9941 (+/- 0,00)	0,9788	0,9932	0,9859
	NB	0,9643 (+/- 0,01)	0,8560	0,9980	0,9216
	KNN	0,9915 (+/- 0,01)	0,9795	0,9800	0,9798
	SVM	0,9937 (+/- 0,00)	0,9770	0,9937	0,9852
8 Küme	DT	0,9941 (+/- 0,00)	0,9788	0,9932	0,9859
	NB	0,9684 (+/- 0,01)	0,8706	0,9980	0,9300
	KNN	0,9913 (+/- 0,01)	0,9786	0,9800	0,9793
	SVM	0,9937 (+/- 0,00)	0,9770	0,9937	0,9852
9 Küme	DT	0,9942 (+/- 0,00)	0,9788	0,9937	0,9862
	NB	0,9689 (+/- 0,01)	0,8728	0,9976	0,9310
	KNN	0,9907 (+/- 0,01)	0,9776	0,9780	0,9778
	SVM	0,9937 (+/- 0,00)	0,9761	0,9946	0,9853
10 Küme	DT	0,9937 (+/- 0,00)	0,9783	0,9922	0,9852
	NB	0,9803 (+/- 0,01)	0,9162	0,9976	0,9551
	KNN	0,9919 (+/- 0,00)	0,9772	0,9844	0,9808
	SVM	**0,9939 (+/- 0,00)**	0,9761	0,9956	0,9857



Şekil 4.25. Occupancy için 2-10 kümeleme F_1-skor değerlerinin KiB ile karşılaştırılması

Tablo 4.11. Wilt veri setinin 2-10 kümeleme için sınıflama sonuçları

Küme Sayısı	Yöntem	Doğruluk	Kesinlik	Duyarlılık	F_1-skor
KiB	DT	**0,9771 (+/- 0,02)**	0,8409	0,7088	0,7692
	NB	0,9595 (+/- 0,01)	0,8736	0,2912	0,4368
	KNN	0,9690 (+/- 0,01)	0,7681	0,6092	0,6795
	SVM	0,9740 (+/- 0,01)	0,9412	0,5517	0,6957
2 Küme	DT	0,9523 (+/- 0,02)	0,5824	0,4061	0,4786
	NB	0,7320 (+/- 0,04)	0,1357	0,7395	0,2294
	KNN	0,8132 (+/- 0,22)	0,1863	0,7318	0,2970
	SVM	0,9527 (+/- 0,02)	0,5899	0,4023	0,4784
3 Küme	DT	0,9665 (+/- 0,01)	0,8779	0,4406	0,5867
	NB	0,8806 (+/- 0,13)	0,2144	0,4559	0,2917
	KNN	0,8829 (+/- 0,11)	0,2646	0,6590	0,3776
	SVM	0,9673 (+/- 0,01)	0,8815	0,4559	0,6010
4 Küme	DT	0,9665 (+/- 0,01)	0,8779	0,4406	0,5867
	NB	0,9585 (+/- 0,01)	0,7632	0,3333	0,4640
	KNN	0,9572 (+/- 0,02)	0,5985	0,6284	0,6131
	SVM	0,9671 (+/- 0,01)	0,8923	0,4444	0,5934
5 Küme	DT	0,9711 (+/- 0,01)	0,8343	0,5785	0,6833
	NB	**0,9628 (+/- 0,01)**	0,7872	0,4253	0,5522
	KNN	0,9347 (+/- 0,07)	0,4331	0,6820	0,5298
	SVM	0,9723 (+/- 0,01)	0,8588	0,5824	0,6941
6 Küme	DT	0,9709 (+/- 0,02)	0,7913	0,6245	0,6981
	NB	0,9572 (+/- 0,01)	0,6800	0,3908	0,4964
	KNN	0,9671 (+/- 0,02)	0,7217	0,6360	0,6762
	SVM	0,9721 (+/- 0,01)	0,8889	0,5517	0,6809
7 Küme	DT	0,9760 (+/- 0,02)	0,8251	0,7050	0,7603
	NB	0,9471 (+/- 0,01)	0,5410	0,1264	0,2050
	KNN	0,9709 (+/- 0,02)	0,7679	0,6590	0,7093
	SVM	**0,9793 (+/- 0,01)**	0,9215	0,6743	0,7788
8 Küme	DT	0,9756 (+/- 0,02)	0,8235	0,6973	0,7552
	NB	0,9366 (+/- 0,02)	0,2326	0,0766	0,1153
	KNN	0,9717 (+/- 0,01)	0,7870	0,6513	0,7128
	SVM	0,9779 (+/- 0,01)	0,9185	0,6475	0,7596
9 Küme	DT	0,9744 (+/- 0,02)	0,8072	0,6897	0,7438
	NB	0,9448 (+/- 0,01)	0,4464	0,0958	0,1577
	KNN	0,9669 (+/- 0,02)	0,7167	0,6398	0,6761
	SVM	0,9783 (+/- 0,01)	0,9239	0,6513	0,7640
10 Küme	DT	0,9735 (+/- 0,02)	0,7904	0,6935	0,7388
	NB	0,9471 (+/- 0,01)	0,5862	0,0651	0,1172
	KNN	**0,9729 (+/- 0,01)**	0,8155	0,6437	0,7195
	SVM	0,9773 (+/- 0,01)	0,9037	0,6475	0,7545



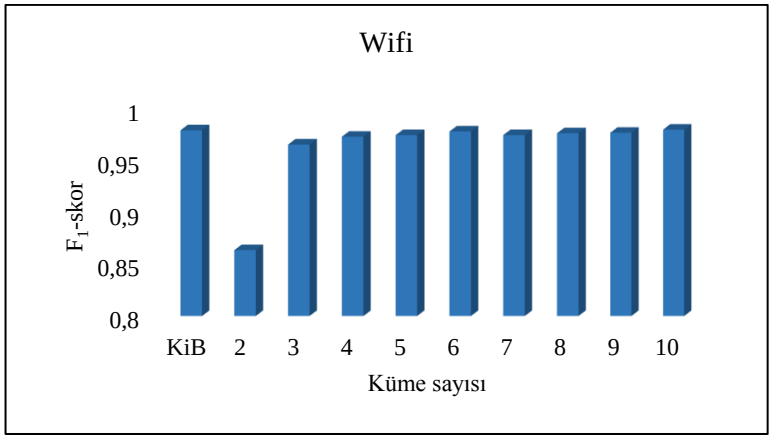
Şekil 4.26. Wilt için 2-10 kümeleme F_1-skor değerlerinin KiB ile karşılaştırılması

Wilt veri seti için 2-10 kümeleme sınıflama başarım sonuçları Tablo 4.11.'de verilmektedir. Tabloya göre DT sınıflayıcı KiB ile, NB 5 küme, SVM 7 küme ve KNN 10 küme ayrıklaştırma ile daha başarılı olmaktadır. Wilt veri setinin 2-10 kümeleme için F_1-skor değerleri Şekil 4.26.'da görülmektedir.

Tablo 4.12. Wifi veri setinin 2-10 kümeleme için sınıflama sonuçları

Küme Sayısı	Yöntem	Doğruluk	Kesinlik	Duyarlılık	F_1-skor
KiB	DT	0,9735 (+/- 0,02)	0,9735	0,9735	0,9735
	NB	0,9750 (+/- 0,02)	0,9759	0,9750	0,9751
	KNN	0,9820 (+/- 0,02)	0,9822	0,9820	0,9820
	SVM	**0,9850 (+/- 0,01)**	0,9851	0,9850	0,9850
2 Küme	DT	0,9345 (+/- 0,02)	0,9365	0,9345	0,9347
	NB	0,7510 (+/- 0,01)	0,8081	0,7510	0,6764
	KNN	0,9055 (+/- 0,04)	0,9084	0,9055	0,9044
	SVM	0,9390 (+/- 0,02)	0,9401	0,9390	0,9390
3 Küme	DT	0,9660 (+/- 0,03)	0,9662	0,9660	0,9660
	NB	0,9590 (+/- 0,03)	0,9609	0,9590	0,9591
	KNN	0,9635 (+/- 0,03)	0,9636	0,9635	0,9634
	SVM	0,9725 (+/- 0,03)	0,9728	0,9725	0,9725
4 Küme	DT	0,9670 (+/- 0,03)	0,9669	0,9670	0,9670
	NB	0,9710 (+/- 0,02)	0,9724	0,9710	0,9712
	KNN	0,9720 (+/- 0,03)	0,9724	0,9720	0,9720
	SVM	0,9805 (+/- 0,02)	0,9806	0,9805	0,9805
5 Küme	DT	0,9655 (+/- 0,03)	0,9655	0,9655	0,9655
	NB	0,9760 (+/- 0,01)	0,9768	0,9760	0,9760
	KNN	0,9730 (+/- 0,02)	0,9733	0,9730	0,9730
	SVM	0,9825 (+/- 0,02)	0,9826	0,9825	0,9825
6 Küme	DT	0,9705 (+/- 0,02)	0,9705	0,9705	0,9705
	NB	**0,9775 (+/- 0,02)**	0,9783	0,9775	0,9775
	KNN	0,9790 (+/- 0,01)	0,9791	0,9790	0,9790
	SVM	0,9840 (+/- 0,01)	0,9842	0,9840	0,9840
7 Küme	DT	0,9655 (+/- 0,02)	0,9654	0,9655	0,9654
	NB	0,9740 (+/- 0,02)	0,9751	0,9740	0,9741
	KNN	0,9780 (+/- 0,01)	0,9780	0,9780	0,9780
	SVM	0,9795 (+/- 0,02)	0,9797	0,9795	0,9795
8 Küme	DT	0,9690 (+/- 0,02)	0,9690	0,9690	0,9690
	NB	0,9760 (+/- 0,02)	0,9772	0,9760	0,9762
	KNN	0,9765 (+/- 0,02)	0,9766	0,9765	0,9765
	SVM	0,9825 (+/- 0,02)	0,9826	0,9825	0,9825
9 Küme	DT	0,9695 (+/- 0,02)	0,9695	0,9695	0,9695
	NB	0,9770 (+/- 0,02)	0,9780	0,9770	0,9771
	KNN	0,9795 (+/- 0,02)	0,9796	0,9795	0,9795
	SVM	0,9800 (+/- 0,02)	0,9802	0,9800	0,9800
10 Küme	DT	**0,9765 (+/- 0,03)**	0,9765	0,9765	0,9765
	NB	0,9760 (+/- 0,02)	0,9768	0,9760	0,9761
	KNN	**0,9830 (+/- 0,02)**	0,9831	0,9830	0,9830
	SVM	0,9825 (+/- 0,02)	0,9827	0,9825	0,9825

Wifi veri seti için 2-10 kümeleme sınıflama başarım sonuçları Tablo 4.12.'de verilmektedir. Tabloya göre SVM sınıflayıcı KiB ile, NB sınıflayıcı 6 küme, DT ve KNN ise 10 küme ayrıklaştırma ile daha başarılı olmaktadır. Wifi veri setinin 2-10 kümeleme için F_1-skor değerleri Şekil 4.27.'de görülmektedir.



Şekil 4.27. Wifi için 2-10 kümeleme F_1-skor değerlerinin KiB ile karşılaştırılması

Veri setleri için, gerek sınıflama sonuçlarının verildiği tablolardan, gerek F_1-skor değerlerinin karşılaştırmalı sonuçlarının verildiği şekillerden rahatlıkla anlaşılabileceği gibi, önerilen 2-10 kümeleme yöntemi, 3 ve daha fazla sayıda küme için KiB'e eş değer ya da daha iyi sonuçlar üretmektedir. Veri setlerinin farklı sınıflama algoritmaları için sınıflama başarısı, farklı küme değerleri için değişkenlik göstermekle beraber Şekil 4.17. ilâ Şekil 4.27. arasında verilen oralama F_1-skor değerleri incelendiğinde, 2-10 kümeleme yönteminin KiB yöntemine göre genelde daha başarılı olduğu kolayca söylenebilir.

BÖLÜM 5. TARTIŞMA VE SONUÇ

Bu çalışmada 11 farklı veri seti üzerinde, önerilen dört farklı ayrıklaştırma yönteminin literatürde iyi bilinen ve sıklıkla kullanılan KiB ayrıklaştırma algoritması ile karşılaştırılması incelenmektedir. Kerber'in önerdiği orijinal KiB algoritmasında (Kerber, 1992) χ^2-eşik değeri Ki-kare tablosundan (Chi-square table) elde edilmektedir. Önerilen yöntemler ise herhangi bir tabloya ihtiyaç duymaksızın farklı yöntemlerle verinin bölüneceği küme sayısını bulmaya dayanmaktadır. Bu yöntemlerden dKiB ve sKiB yine literatürde iyi bilinen k-ortalamalar (k-means) algoritmasından yararlanılarak, çalışmaya konu olan herbir veri setinin bölüneceği en uygun küme sayısını bulmaya dayanmaktadır. Bir diğer yöntem olan kkKiB algoritmasında, veri setleri her bir özniteliğinin hesaplanan kendi karekök değerine göre kümelere ayrılmaktadır. 2-10'lu ayrıklaştırma yönteminde ise temel amaç veriyi 2'den başlayıp 10'a kadar sıralı bir şekilde kümelere ayırmak ve farklı kümelere ayırmanın veri setleri üzerindeki ayrıklaştırma sonuçlarını gözlemlemeye dayanmaktadır.

Çalışmada kullanılan veri setlerinin tamamı, gerçek dünya verisine (real world data) sahip ve liteatürde çeşitli çalışmalarda kullanılan veri setleridir. Ayrıklaştırma işleminin mantığına uygun olarak çoğunlukla sürekli özniteliklerden oluşan bu veri setleri, özellikle tercih edilmektedir. Heart veri seti gibi bazı nominal öznitelikler içeren veri setlerinin bu öznitelikleri, ayrıklaştırma işlemine tabi tutulmamakla beraber sınıflama esnasında veriye dahil edilmektedir. Veri setleri farklı boyutlarda ve farklı sayıda sınıf etiketinden oluşan; hacimlerine göre de küçük, orta ve büyük ölçekli sayılabilecek veri setleridir. KiB algoritması, sınıf etiketlerini dikkate alarak ayrıklaştırma yaptığı için, algoritmaların farklı sayıda sınıf etiketi içeren veriler üzerinde test edilmesi önem arz etmektedir. Böylece önerilen yöntemlerin KiB algoritması ile karşılaştırılmasında veriler her açıdan değerlendirilmiş olmaktadır.

Yöntemlerin performansını karşılaştırmada kullanılan sınıflama algoritmaları da, literatürde ayrıklaştırma algoritmalarının başarısını test etmede sıklıkla kullanılan yöntemlerdir. Özellikle karar ağçları (DT) ve Naive Bayes (BN) bu alanda başat rol oynamaktadır. Çünkü bu algoritmalar, yapıları gereği ayrık veriler ile daha iyi performans ortaya koymaktadır. Sınıflama işlemi veriye katmanlı 10-kat çapraz doğrulama yöntemi uygulanarak yapılmaktadır. Bu durum olası bir aşırı öğrenme (overfitting) ya da eksik öğrenme (underfitting) durumunu en aza indirgeyerek daha güvenilir bir sınıflama yapmayı sağlamaktadır. Ayrıca sınıflama başarısını ölçmede birden fazla değerlendirme ölçütü kullanılarak, tahmin başarısı verinin hem pozitif hem de negatif sınıfları için ölçülmektedir. Bu açıdan doğruluk değeri sınıflamanın performansı hakkında genel bir fikir verse de, tahmin edilen değerleri bütün olarak değerlendirdiği için başarı hakkında kesin bir fikir vermeyebilir. Örneğin 90 tane negatif ve 10 tane de pozitif vaka içeren bir veri seti için, negatif vakaların tamamını doğru tahmin eden ancak hiçbir pozitif vakayı yakalayamayan bir sınıflama algoritmasının sınıflama doğruluk (accuracy) değeri %90'dır. Ancak hem kesinlik (precision) hem de duyarlılık (recall) değerleri 0'dır. Bundan dolayı çalışmada kullanılan metrikler, yöntemlerin başarısını karşılaştırmada her açıdan daha sağlıklı değerlendirme yapmayı olanaklı kılmaktadır.

Çalışmada elde edilen sonuçlar incelendiğinde önerilen yöntemlerin genel anlamda orijinal KiB algoritmasından daha başarılı olduğu görülmektedir. Bölüm 4'te araştırma bulguları herbir veri seti için ayrıntılı bir şekilde verilmektedir. Ancak bu veri setlerini bütün olarak ele alıp ortalama değerlere bakmakta da fayda bulunmaktadır. Örneğin KiB algoritmasının çalışmada incelenen 11 veri seti için ortalama sınıflama doğruluk değeri %82,70 iken bu oran dKiB algoritması için %82,92, sKiB algritması için %80,22, kkKiB algoritması için ise %84,07 olmaktadır. 2-10'lu ayrıklaştırmada elde edilen f_1-skor değerleri her bir veri seti için KiB ile karşılaştırıldığında; Iris için KiB'in ortalama f_1-skor değeri %95,50, 4-küme f_1-skor değeri %95,51; Bupa için KiB'in ortalama f_1-skor değeri %62,45, 7-küme f_1-skor değeri %65,98; Heart için KiB'in ortalama f_1-skor değeri %79,39, 3-küme f_1-skor değeri %82,74; WholeSale için KiB'in ortalama f_1-skor değeri %89,21, 6-küme f_1-skor değeri %92,04; Ecoli için KiB'in ortalama f_1-skor değeri %79,97, 4-küme f_1-skor değeri %82,52; Vertebral için KiB'in

ortalama f_1-skor değeri %81,65, 2-küme f_1-skor değeri %81,22, Yeast için KiB'in ortalama f_1-skor değeri %46,02, 9-küme f_1-skor değeri %46,68, UKMD için KiB'in ortalama f_1-skor değeri %79.21, 10-küme f_1-skor değeri %85,52, Wilt için KiB'in ortalama f_1-skor değeri %64,53, 6-küme f_1-skor değeri %63,79, Occupancy için KiB'in ortalama f_1-skor değeri %97,69, 3-küme f_1-skor değeri %98,28, Wifi için KiB'in ortalama f_1-skor değeri %97,89, 6-küme f_1-skor değeri %97,95 olmaktadır.

Sonuçlar ortalama değerlerine göre bütün olarak değerlendirildiğinde önerilen yöntemlerin orijinal KiB algoritmasından genel olarak daha başarılı sonuçlar ürettiği görülmektedir. Özellikle verinin öznitelik karekök değerlerine göre ayrıklaştırıldığı kkKiB algoritmasının KiB'e göre oldukça başarılı olduğu dikkate değer bir olgudur. Orijinal KiB algoritmasında uygulama anlamlılık seviyesi ve serbestlik derecesi üzerinden yapıldığından Ki-kare tablosuna ihtiyaç duyulmaktadır. Tezde önerilen yöntemler için herhangi bir tablo kullanmak gerekmez. Verilerin kendi iç dinamikleri kullanılarak K-ortalama veya veri öznitelik karekök değerlerine göre verinin bölüneceği aralık sayısı belirlenmektedir. Kısım 4.1'de önerilen yöntemlere ait algoritmalar dikkatle incelendiğinde, önerilen yöntemlerin uygulanması oldukça basit ve anlaşılması da oldukça kolaydır. Veri analizi açısından bahsedilen bu iki konunu önemi dikkate alındığında, önerilen yöntemlerin literatüre katkı yapacağı düşünülmektedir

KAYNAKLAR

Ali, Z., Shahzad, W. 2016. Comparative Study of Discretization Methods on the Performance of Associative Classifiers. In 2016 International Conference on Frontiers of Information Technology, IEEE, 87-92.

Allen, D. M. 1974. The relationship between variable selection and data agumentation and a method for prediction. technometrics, 16(1), 125-127.

Almeida, T. A., Almeida, J.,Yamakami, A. 2011. Spam filtering: how the dimensionality reduction affects the accuracy of Naive Bayes classifiers. J ournal of Internet Services and Applications, 1(3), 183-200.

Arlot, S., Celisse, A. 2010. A survey of cross-validation procedures for model selection. Statistics surveys, 4, 40-79.

Asuncion, A., Newman, D. 2007. UCI machine learning repository.

Ayyaz, M. N., Javed, I., Mahmood, W. 2016. Handwritten character recognition using multiclass svm classification with hybrid feature extraction. Pakistan Journal of Engineering and Applied Sciences.

Ball, N. M., Brunner, R. J., Myers, A. D., Tcheng, D. 2006. Robust machine learning applied to astronomical data sets. I. star-galaxy classification of the Sloan Digital Sky Survey DR3 using decision trees. The Astrophysical Journal, 650(1), 497.

Batista, G. E., Monard, M. C. 2002. A study of K-nearest neighbour as an imputation method. His, 87(251-260), 48.

Bayes, T. 1763. LII. An essay towards solving a problem in the doctrine of chances. By the late Rev. Mr. Bayes, FRS communicated by Mr. Price, in a letter to John Canton, AMFR S. Philosophical transactions of the Royal Society of London, (53), 370-418.

Bazi, Y., Melgani, F. 2006. Toward an optimal SVM classification system for hyperspectral remote sensing images. IEEE Transactions on geoscience and remote sensing, 44(11), 3374-3385.

Belson, W. A. 1959. Matching and prediction on the principle of biological classification. Journal of the Royal Statistical Society: Series C (Applied Statistics), 8(2), 65-75.

Bertier, P., Bouroche, J. M. 1981. Analyse des Données Multidimensionnelles, PUF, coll. Systèmes-Décisions, Paris.

Beyer, K., Goldstein, J., Ramakrishnan, R., Shaft, U. 1999. When is “nearest neighbor” meaningful?. In International conference on database theory (pp. 217-235). Springer, Berlin, Heidelberg.

Breiman, L., Friedman, J. H., Olshen, R. A., Stone, C. J. 1984. Classification and regression trees. Belmont, CA: Wadsworth. International Group, 432, 151-166.

Boser, B. E., Guyon, I. M., Vapnik, V. N. 1992. A training algorithm for optimal margin classifiers. In Proceedings of the fifth annual workshop on Computational learning theory (pp. 144-152). ACM.

Boulle, M. 2004. Khiops: A statistical discretization method of continuous attributes. Machine learning, 55(1), 53-69.

Bremner, D., Demaine, E., Erickson, J., Iacono, J., Langerman, S., Morin, P., Toussaint, G. 2005. Output-sensitive algorithms for computing nearest-neighbour decision boundaries. Discrete & Computational Geometry, 33(4), 593-604.

Caruana, R., Niculescu-Mizil, A. 2006. An empirical comparison of supervised learning algorithms. In Proceedings of the 23rd international conference on Machine learning (pp. 161-168).

Cebeci, Z., Yildiz, F. 2017. Comparison of Chi-square based algorithms for discretization of continuous chicken egg quality traits, J. Agric. Inform, 8 (1): 13-22.

Chen, Q., Huang, M., Xu, Q., Wang, H., Wang, J. 2020. Reinforcement Learning-Based Genetic Algorithm in Optimizing Multidimensional Data Discretization Scheme. Mathematical Problems in Engineering, 2020.

Chmielewski, M. R., Grzymala-Busse, J. W. 1996. Global discretization of continuous attributes as preprocessing for machine learning. International Journal of Approximate Reasoning, 15(4):319-331.

Choudhury, A., Kosorok, M. R. 2020. Missing data imputation for classification problems. arXiv preprint arXiv:2002.10709.

Colledanchise, M., Ögren, P. 2016. How behavior trees modularize hybrid control systems and generalize sequential behavior compositions, the subsumption architecture, and decision trees. IEEE Transactions on robotics, 33(2), 372-389.

Cover, T., Hart, P. 1967. Nearest neighbor pattern classification. IEEE transactions on information theory, 13(1), 21-27.

Cuingnet, R., Rosso, C., Chupin, M., Lehéricy, S., Dormont, D., Benali, H., Colliot, O. 2011. Spatial regularization of SVM for the detection of diffusion alterations associated with stroke outcome. Medical image analysis, 15(5), 729-737.

Dai, J. H. 2004. A genetic algorithm for discretization of decision systems. In Proceedings of 2004 International Conference on Machine Learning and Cybernetics, IEEE, (3):1319-1323.

Deradjat, D., Minshall, T. 2018. Decision trees for implementing rapid manufacturing for mass customisation. CIRP Journal of Manufacturing Science and Technology, 23, 156-171.

Devos, O., Downey, G., Duponchel, L. 2014. Simultaneous data pre-processing and SVM classification model selection based on a parallel genetic algorithm applied to spectroscopic data of olive oils. Food chemistry, 148, 124-130.

Drias, H. Moulai, H. Rehkab, N. 2018. LR-SDiscr: An efficient algorithm for supervised discretization. In Asian Conference on Intelligent Information and Database Systems (pp. 266-275). Springer, Cham.

Everitt B.S., Skrondal A. 2010. Cambridge Dictionary of Statistics, Cambridge University Press.

Everitt, B. S., Landau, S., Leese, M., Stahl, D. 2011. Miscellaneous clustering methods. Cluster analysis, 215-255.

Fayyad, U., Piatetsky-Shapiro, G., Smyth, P. 1996. From data mining to knowledge discovery in databases. AI Magazine, 17(3): 37-37.

Fix, E., Hodges, J.L. 1951. Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties (Report). USAF School of Aviation Medicine, Randolph Field, Texas.

Gini, C. 1912. Variabilità e mutabilità. Reprinted in Memorie di metodologica statistica (Ed. Pizetti, E).

Gonzalez-Abril, L., Cuberos, F. J., Velasco, F., Ortega, J. A. 2009. Ameva: An autonomous discretization algorithm. Expert Systems with Applications, 36(3), 5327-5332.

Han, J., Kamber, M., Pei, J. 2011. Data Mining: Concepts and Techniques, 3rd. Edition, Morgan Kaufmann, USA, 5-7.

Hand, D. J. 2007. Principles of data mining. Drug safety, 30(7), 621-622.

Hunt, E. B., Marin, J., Stone, P. J. 1966. Experiments in induction.

Joachims, T. 1998. Text categorization with support vector machines: Learning with many relevant features. In European conference on machine learning (pp. 137-142). Springer, Berlin, Heidelberg.

John, G. H., Langley, P. 2013. Estimating continuous distributions in Bayesian classifiers. arXiv preprint arXiv:1302.4964.

Jonsson, P., Wohlin, C. 2004. An evaluation of k-nearest neighbour imputation using likert data. In 10th International Symposium on Software Metrics, 2004. Proceedings. (pp. 108-118). IEEE.

Jordan, A. 2002. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. Advances in neural information processing systems, 14(2002), 841.

Kalpana, P., Mani, K. 2017. An exploratory analysis between the feature selection algorithms IGMBD and IGChiMerge. IJ Information Technology and Computer Science, 7, 61-68.

Kerber, R. 1992. Chimerge: Discretization of numeric attributes. In Proceedings of the tenth national conference on Artificial intelligence, San Jose, California, 123-128.

Koçoğlu.,F.,Ö. 2012. Veri madenciliği sürecinde veri ayrıklaştırma yöntemlerinin karşılaştırılması ve bir uygulama, İstanbul Üniversitesi, Fen Bilimleri Enstitüsü, Enformatik Anabilim Dalı, Yüksek Lisans Tezi.

Lakoumentas, J. 2012. Optimizations of the naïve-Bayes classifier for the prognosis of B-Chronic Lymphocytic Leukemia incorporating flow cytometry data, Computer Methods and Programs in Biomedicine, 108 (1), 158-167.

Lavangnananda, K., Chattanachot, S. 2017. Study of discretization methods in classification. 9th International Conference on Knowledge and Smart Technology, 50-55.

Li, F., Yang, M., Li, Y., Zhang, M., Wang, W., Yuan, D., Tang, D. 2020. An improved clear cell renal cell carcinoma stage prediction model based on gene sets. BMC Bioinformatics, 21, 1-15.

Liu, H., Hussain, F., Tan, C. L. ve Dash, M. 2002. Discretization: An enabling technique, Data mining and knowledge discovery, 6 (4): 393-423.

Liu, H., Setiono, R. 1995. Chi2: Feature selection and discretization of numeric attributes. In Proceedings of 7th IEEE International Conference on Tools with Artificial Intelligence (pp. 388-391). IEEE.

MacQueen, J. 1967. Some methods for classification and analysis of multivariate observations. In Proceedings of the fifth Berkeley symposium on mathematical statistics and probability (Vol. 1, No. 14, pp. 281-297). 1. University of California Press. pp. 281–297

Maitra, D. S., Bhattacharya, U., Parui, S. K. 2015. CNN based common approach to handwritten character recognition of multiple scripts. In 2015 13th International Conference on Document Analysis and Recognition (ICDAR) (pp. 1021-1025). IEEE.

Maity, A. 2016. Supervised Classification of RADARSAT-2 Polarimetric Data for Different Land Features. arXiv preprint arXiv:1608.00501.

McCallum, A., Nigam, K. 1998. A comparison of event models for naive bayes text classification. In AAAI-98 workshop on learning for text categorization (Vol. 752, No. 1, pp. 41-48).

McCandless, L. C., Gustafson, P., Levy, A. 2007. Bayesian sensitivity analysis for unmeasured confounding in observational studies. Statistics in medicine, 26(11), 2331-2347.

Minnie, D.,Srinivasan, S. 2013. Preprocessing of Automated Blood Cell Counter Data and Discretization of Data Using Chi Merge Algorithm in Clinical Pathology. Advances in Computing and Information Technology, Springer, Berlin, Heidelberg, 511-519.

Mitov, I., Ivanova, K., Markov, K., Velychko, V., Stanchev, P., Vanhoof, K. 2009. Comparison of discretization methods for preprocessing data for pyramidal growing network classification method. New trends in intelligent technologies, sofia, 31-39.

Murty, M. N., Devi, V. S. 2011. Pattern recognition: An algorithmic approach. Springer Science & Business Media.

Nigam, K., McCallum, A. K., Thrun, S., Mitchell, T. 2000. Text classification from labeled and unlabeled documents using EM. Machine learning, 39(2), 103-134.

Qu, W., Yan, D., Sang, Y., Liang, H., Kitsuregawa, M., Li, K. 2008. A novel Chi2 algorithm for discretization of continuous attributes. In Asia-Pacific Web Conference (pp. 560-571). Springer, Berlin, Heidelberg.

Quinlan, J. R. 1986. Induction of decision trees. Machine learning, 1(1), 81-106.

Özekes, S. 2003. Data Mining Models and Application Areas, İstanbul Commerce University Journal of Science, 3, 65-82.

Pang, S., Kim, D., Bang, S. Y. 2005. Face membership authentication using SVM classification tree generated by membership-based LLE data partition. IEEE transactions on Neural networks, 16(2), 436-446.

Piryonesi, S. M., El-Diraby, T. E. 2020. Data analytics in asset management: Cost-effective prediction of the pavement condition index. Journal of Infrastructure Systems, 26(1), 04019036.

Powers, David M. W. 2011. Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation. Journal of Machine Learning Technologies. 2 (1): 37–63.

Rajathi, S., Radhamani, G. 2016. Prediction and analysis of Rheumatic heart disease using kNN classification with ACO. In 2016 International Conference on Data Mining and Advanced Computing (SAPIENCE) (pp. 68-73). IEEE.

Ramaswamy, S., Rastogi, R., Shim, K. 2000. Efficient algorithms for mining outliers from large data sets. In Proceedings of the 2000 ACM SIGMOD international conference on Management of data (pp. 427-438).

Ramkumar, G. D., Swami, A. N. 1998. Clustering Data Without Distance Functions, IEEE Data Eng. Bull., 21(1): 9-14.

Richeldi, M., Rossotto, M. 1995. Class-driven statistical discretization of continuous attributes. In European Conference on Machine Learning (pp. 335-338). Springer, Berlin, Heidelberg.

Rennie, J., Shih, L., Teevan, J., Karger, D. 2003. Tackling the poor assumptions of Naive Bayes classifiers (PDF). ICML.

Ropero, R. F., Renooij, S.,Van der Gaag, L. C. 2018. Discretizing environmental data for learning Bayesian-network classifiers. Ecological modelling, 368, 391-403.

Rosati, S., Balestra, G., Giannini, V., Mazzetti, S., Russo, F., Regge, D. 2015. ChiMerge discretization method: Impact on a computer aided diagnosis system for prostate cancer in MRI. In 2015 IEEE International Symposium on Medical Measurements and Applications (MeMeA) Proceedings (pp. 297-302). IEEE.

Rousseeuw, P. J. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, Journal of computational and applied mathematics, 20, 53-65.

Saini, R., Bindal, N., Bansal, P. 2015. Classification of heart diseases from ECG signals using wavelet transform and kNN classifier. In International Conference on Computing, Communication & Automation (pp. 1208-1215). IEEE.

Salama, M. A., Hassanien, A. E., Fahmy, A. A. 2011. Feature evaluation based fuzzy C-mean classification. IEEE International Conference on Fuzzy System, IEEE, 2534-2539.

Samworth, Richard J. 2012. Optimal weighted nearest neighbour classifiers, Annals of Statistics. 40 (5): 2733–2763.

Sang, Y., Qi, H., Li, K., Jin, Y., Yan, D., Gao, S. 2014. An effective discretization method for disposing high-dimensional data. Information Sciences, 270, 73-91.

Satchidananda, S. S., Simha, J. B. 2006. Comparing decision trees with logistic regression for credit risk analysis. International Institute of Information Technology, Bangalore, India.

Savaş, S., Topaloğlu, N., Yılmaz, M. 2012. Veri madenciliği ve Türkiye'deki uygulama örnekleri, İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi,11(21): 1-23.

Shaikhina, T., Lowe, D., Daga, S., Briggs, D., Higgins, R., Khovanova, N. 2015. Machine learning for predictive modelling based on small data in biomedical engineering. IFAC-PapersOnLine, 48(20), 469-474.

Shariaty, F., Davydov, V. V., Yushkova, V. V., Glinushkin, A. P., Rud, V. Y. 2019. Automated pulmonary nodule detection system in computed tomography images based on Active-contour and SVM classification algorithm. In Journal of Physics: Conference Series (Vol. 1410, No. 1, p. 012075). IOP Publishing.

Shankar, K., Lakshmanaprabu, S. K., Gupta, D., Maseleno, A., De Albuquerque, V. H. C. 2020. Optimal feature-based multi-kernel SVM approach for thyroid disease classification. The journal of supercomputing, 76(2), 1128-1143.

Shannon, C.E. 1948. A Mathematical Theory of Communication, Bell System Technical Journal. 27 (3): 379–423.

Shaw, B., Jebara, T.2009. Structure preserving embedding. In Proceedings of the 26th Annual International Conference on Machine Learning (pp. 937-944).

Shouman, M., Turner, T., Stocker, R. 2011. Using decision tree for diagnosing heart disease patients. In Proceedings of the Ninth Australasian Data Mining Conference-Volume 121 (pp. 23-30).

Shukla, A. K., Singh, P., Vardhan, M. 2019. A new hybrid wrapper TLBO and SA with SVM approach for gene expression data. Information Sciences, 503, 238-254.

Singh, A., Tiwari, V., Tentu, A. N. 2018. A Machine Vision Attack Model on Image Based CAPTCHAs Challenge: Large Scale Evaluation. In International Conference on Security, Privacy, and Applied Cryptography Engineering (pp. 52-64). Springer, Cham.

Sriwanna, K., Boongoen, T., Iam-On, N. 2017. Graph clustering-based discretization of splitting and merging methods (GraphS and GraphM). Human-centric Computing and Information Sciences, 7(1), 21.

Stehman, S.V, 1997. Selecting and interpreting measures of thematic classification accuracy, Remote Sensing of Environment, 62 (1), 77–89.

Stone, M. 1977. An asymptotic equivalence of choice of model by cross□validation and Akaike's criterion. Journal of the Royal Statistical Society: Series B (Methodological), 39(1), 44-47.

Su, C. T., Hsu, J. H. 2005. An extended chi2 algorithm for discretization of real value attributes. IEEE transactions on knowledge and data engineering, 17(3), 437-441.

Sweilam, N. H., Tharwat, A. A., Moniem, N. A. 2010. Support vector machine for diagnosis cancer disease: A comparative study. Egyptian Informatics Journal, 11(2), 81-92.

Tahan, M. H., Asadi, S. 2018. MEMOD: a novel multivariate evolutionary multi-objective discretization. Soft Computing, 22(1), 301-323.

Tahraoui, A., Kheddam, R., Bouakache, A.,Belhadj-Aissa, A. 2017. Multivariate alteration detection and ChiMerge thresholding method for change detection in bitemporal and multispectral images. 5th International Conference on Electrical Engineering-Boumerdes, IEEE, 1-6.

Tarabalka, Y., Fauvel, M., Chanussot, J., Benediktsson, J. A. 2010. SVM-and MRF-based method for accurate classification of hyperspectral images. IEEE Geoscience and Remote Sensing Letters, 7(4), 736-740.

Tay, F. E., Shen, L. 2002. A modified Chi2 algorithm for discretization. IEEE Transactions on knowledge and data engineering, 14(3), 666-670.

Thaiphan, R., Phetkaew, T. 2018. Comparative Analysis of Discretization Algorithms on Decision Tree. In 2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS) IEEE, 63-67.

Thorndike, R, L.1953. Who belongs in the family?, Psychometrika, 18(4), 267-276,

Vejkanchana, N., Kucharoen, P. 2019. Continuous Variable Binning Algorithm to Maximize Information Value Using Genetic Algorithm. In International Conference on Applied Informatics (pp. 158-172). Springer, Cham.

Wang, K., Liu, B. 1998. Concurrent discretization of multiple attributes. In Pacific Rim International Conference on Artificial Intelligence (pp. 250-259). Springer, Berlin, Heidelberg.

Wang, Z. Q., Sun, X., Zhang, D. X., Li, X. 2006. An optimal SVM-based text classification algorithm. In 2006 International Conference on Machine Learning and Cybernetics (pp. 1378-1381). IEEE.

Wiggins, M. 2007. Evolving a Bayesian classifier for ECG-based age classification in medical applications, Applied Soft Computing. 8 (1). ss. 599-608.

Wójciak, M., Łupińska-Dubicka, A. 2018. Empirical comparison of methods of data discretization in learning probabilistic models, Advances in Computer Science Research, 14, 177-192.

Wu, X., Kumar, V., Quinlan, J. R., Ghosh, J., Yang, Q., Motoda, H., Steinberg, D. 2008. Top 10 algorithms in data mining. Knowledge and information systems, 14(1), 1-37.

Wu, G., Zhao, Z., Fu, G., Wang, H., Wang, Y., Wang, Z., Huang, L. 2019. A fast knn-based approach for time sensitive anomaly detection over data streams. In International Conference on Computational Science (pp. 59-74). Springer, Cham.

Zhang, H. 2004. The Optimality of Naive Bayes,". In Proc. Seventeenth Int. Florida Artif. Intell. Res. Soc. Conf. FLAIRS 2004 (Vol. 1, No. 2, pp. 1-6).

Zhang, H. 2005. Exploring conditions for the optimality of Naive Bayes. International Journal of Pattern Recognition and Artificial Intelligence, 19(02), 183-198.

Zhang, S. 2012. Nearest neighbor selection for iteratively kNN imputation. Journal of Systems and Software, 85(11), 2541-2552.

Zou, L., Yan, D., Karimi, H. R., Shi, P. 2013. An algorithm for discretization of real value attributes based on interval similarity. Journal of Applied Mathematics, 2013.

R Foundation for Statistical. (2016). https://www.R-project.org. Erişim Tarihi: 11.03.2021.

ÖZGEÇMİŞ

Adı Soyadı **:** Nuran PEKER

ÖĞRENİM DURUMU

Derece	Eğitim Birimi	Mezuniyet Yılı
Doktora	Sakarya Üniversitesi / Fen Bilimleri Enstitüsü / Endüstri Mühendisliği	Devam ediyor
Yüksek Lisans	Kocaeli Üniversitesi / Fen Bilimleri Enstitüsü / Bilgisayar Mühendisliği	2017
Lisans	Sakarya Üniversitesi / Mühendislik Fakültesi / Bilgisayar Mühendisliği	2019
Lisans	Kocaeli Üniversitesi / Teknik Eğitim Fakültesi / Bilgisayar Öğretmenliği	2014

İŞ DENEYİMİ

Yıl	Yer	Görev
2019-Halen	İstanbul	Bilişim Öğretmeni

YABANCI DİL

İngilizce

ESERLER

1. Peker, N., & Kubat, C. (2021). Application of Chi-square discretization algorithms to ensemble classification methods. *Expert Systems with Applications*, 115540.

2. Peker, N., & Kubat, C. (2021). A Hybrid Modified Deep Learning Data Imputation Method for Numeric Datasets. *International Journal of Intelligent Systems and Applications in Engineering*, *9*(1), 6-11.

3. Peker, N., & Kubat, C. (2020). Boyut Azaltmanın Bulanık C-Ortalama Kümeleme Teknikleri Üzerindeki Etkisi. *Veri Bilimi*, *4*(1), 1-7.

4. Peker, N., & Kubat, C. (2019). Using Chi-square based Discretization Algorithms for Ensemble Classification Methods, *10th International Symposium on Intelligent Manufacturing and Service Systems*.