T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

VERİ MADENCİLİĞİ METOTLARINDAN OLAN KÜMELEME ALGORİTMALARININ UYGULAMALI ETKİNLİK ANALİZİ

YÜKSEK LİSANS TEZİ

End.Müh. Tamer ALTINTAŞ

Enstitü Anabilim Dalı : ENDÜSTRİ MÜHENDİSLİĞİ

Tez Danışmanı : Yrd.Doç.Dr. İbrahim ÇİL

Haziran 2006

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

VERİ MADENCİLİĞİ METOTLARINDAN OLAN KÜMELEME ALGORİTMALARININ UYGULAMALI ETKİNLİK ANALİZİ

YÜKSEK LİSANS TEZİ

End.Müh. Tamer ALTINTAŞ

Enstitü Anabilim Dalı : ENDÜSTRİ MÜHENDİSLİĞİ

Bu tez 26 / 06 /2006 tarihinde aşağıdaki jüri tarafından Oybirliği ile kabul edilmiştir.

Prof. Dr. Harun TAŞKIN	Doç. Dr. Nejat YUMUŞAK	Yrd. Doç. Dr. İbrahim ÇİL
Jüri Başkanı	**Üye**	**Üye**

TEŞEKKÜR

Bu çalışmanın hazırlanmasında yardımlarını bir an olsun esirgemeyen danışman hocam Yrd.Doç.Dr. İbrahim ÇİL'e teşekkürlerimi sunmayı bir borç bilirim. Ayrıca, hayatımın her safhasında bana büyük destek olan aileme ve teknik konulardaki yardımlarından dolayı ev arkadaşlarıma teşekkür ederim.

Tamer ALTINTAŞ

İÇİNDEKİLER

ŞEKİLLER LİSTESİ

ÖZET

Anahtar Kelimeler: Veri madenciliği, kümeleme, weka

Hızla gelişen bilgisayar teknolojileri ile artık verileri veri ambarlarında saklamak çok kolaylaştığı gibi verilerin boyutları da inanılmaz seviyelere gelmiştir. Böyle büyük boyutlardaki verilerden işimize yarayacak, geleceği daha iyi görebilmemizi ve tahminler yapabilmemizi sağlayacak bilgileri elde etmenin en kullanışlı yolarından biri de veri madenciliği yöntemleridir.

Bu çalışmada veri madenciliği yöntemlerinden olan kümeleme algoritmaları derinlemesine incelenmiştir. Bir bankanın müşteri bilgilerini barındıran bir veri tabanı üzerinde yapılan kümeleme çalışması ile bankanın müşterilerini kredilerini ödeme durumlarına göre kümelere ayırması sağlanmıştır. Bu işlem yapılırken kümeleme algoritmalarından K-means, EM (Expectation Maximization), Farthest-First, Cobweb ve Density Based algoritmaları kullanılmış ve bunların karşılaştırılmasına olanak sağlanmıştır.

EFFICIENCY ANALYSIS OF CLUSTERING ALGORITHMS USING IN DATA MINING

SUMMARY

Keywords: Data mining, clustering, weka

The major reason that data mining became one of the hottest current technologies of the information age is the wide availability of huge amounts of data and the need for turning such data into useful information and knowledge. As computer systems getting cheaper and computer power increases, the amount of data available to be collected and processed increases. Therefore using techniques that operates very well with large amounts of data becomes an obvious choice. The information and knowledge gained can be used for applications ranging from business management, production control, and market analysis, to engineering design and science exploration. In this study, clustering alghorythms were discussed with an application.

BÖLÜM 1. GİRİŞ

Günümüz ekonomilerinde işletmelerin yoğun teknoloji ve bilgisayar kullanımlarının artmasıyla birlikte müşteri verileri ve dataları da elektronik ortamda tutulmaya başlanmıştır. Müşteri verilerinin elektronik ortamda tutulması bu verilerin kolayca belli amaçlara yönelik olarak kullanılmasını gündeme getirmiş ve özellikle müşteri segmentlerinin geniş olduğu sektörlerde oluşturulan müşteri veri tabanlarının işletme amaçları ve politikaları çerçevesinde kullanılması oldukça önemli hale gelmiştir. Özellikle yeni teknolojilerden etkilenen ve dijital ekonomi olarak da adlandırılan bu yeni dünyada bilgi ve zaman boyutlarının önemi de gittikçe artmaktadır. Buna bağlı olarak işletmelerin doğru ve anlamlı bilgiye dayalı hızlı karar alma gereği her zamankinden daha fazla ön plana çıkmıştır.

İşte yöneticilerin var olan bu yoğun rekabet ortamında doğru kararı en hızlı şekilde verebilmeleri için işletmeyle ilgili taraf ve işletme süreçleri hakkında detaylı bilgiye sahip olmaları gerekmektedir. Elektronik ortamda mevcut müşteri hakkında tutulan veri sayısının fazla olması doğru karara en hızlı şekilde ulaşmayı kolaylaştırmaktadır. Fakat doğru karara en hızlı şekilde ulaşmak sadece verileri toplamakla değil, aynı zamanda toplanan bu veri yığınlarını analiz edip, yorumlayarak anlamlı raporlar oluşturmakla mümkündür.
İşte işletmelerin müşterileri hakkında tuttukları verileri analiz edip yorumlanması ve anlamlı raporlar haline getirilerek işletme karar süreçlerinde rol almasını sağlayan yöntem "veri madenciliği" olarak ifade edilmektedir
Çok büyük veri yığınları altında saklı olan bilgilere ulaşmak için uzun yıllar boyu yapıla gelen çalışmaların neticesinde bir dizi metodoloji geliştirilmiştir. İşte incelemeye çalışılan konu uzun yıllar, özellikle batı ülkelerinde üzerinde çalışılmış fakat, gerçek hayatta, yazılım endüstrisinin son yıllarda üretmiş olduğu ileri teknoloji ürünü yazılımlar ile kullanılmaya başlamıştır.

Bu yöntemin gelecek yıllar için üstlenmiş olduğu misyon hakkında dünyanın önde gelen araştırma ve danışmanlık firmalarından açıklanan rakamlar oldukça dikkat çekicidir. Örneğin, Gartner Group Araştırma şirketi, gelecek on yıl içinde, hedef pazarlarda veri madenciliği kullanımının % 80'lere ulaşacağı tahmininde bulunmaktadır. Diğer taraftan META Group ise, veri madenciliği pazarının bu yıl 800 milyon dolara yükseleceği yönünde tahminlerde bulunmaktadır [3].

Bu çalışmada, veri madenciliği teorik açıdan incelenmeye çalışılmış ve veri madenciliğinin işletmelerdeki kullanım alanları belirtilerek ne tür faydalar sağlanabileceği, bu sürecin nasıl işlediği, bu süreçte ne tür yöntemlerden faydalanıldığı ve veri madenciliğini ne tür gelişmelerin etkilediği ortaya konulmaya çalışılmıştır.

BÖLÜM 2. VERİ MADENCİLİĞİNİN TANIMI VE TARİHİ GELİŞİMİ

2.1. Veri Madenciliğinin Tanımı

Günümüzde veri tabanları artık tera byte'larla ölçülmektedir. Bu ölçekte büyük veriler, stratejik öneme sahip bilgileri gizlemektedir. Bu bağlamda veri madenciliği(VM), büyük veri tabanlarındaki gizli bilgi ve yapıyı açığa çıkarmak için çok sayıda veri analizi aracını kullanan bir süreçtir. VM'nin üç farklı bakış açısı bulunmaktadır; veri tabanı bakış açısı, makine öğrenim bakış açısı ve istatistiksel bakış açısı. Yazılan kitaplar ve geliştirilen bilgisayar programları da bu farklı bakış açılarına uygun olarak yapılmaktadır. Konunun önemi anlaşıldıkça bu alanla ilgili bilgisayar programları da hızla artmaya başlamıştır.

İşte, büyük miktarlarda ve oldukça hızlı toplanan verilerin çeşitli analizler sonucunda anlamlı bilgilere dönüştürülmesi noktasında "veri madenciliği" önemli bir rol oynamaktadır. VM tanımları incelendiğinde, bu tanımların ortak unsurlarının ilki "çok fazla" miktarlarda verinin veri ambarlarında tutulması, ikincisi ise bu verilerden "anlamlı" bilgiler elde edilmesidir [2].

VM ile ilgili yapılan tanımlardan bazıları aşağıda özetlenmiştir.
Konunun önde gelen uzmanlarından Piatetsky-Shapiro, verilerden daha önceden bilinmeyen, muhtemelen faydalı bilginin monoton olmayan bir süreçte çıkartılması işlemi olarak tanımlamaktadır. Bu süreç kümeleme (Clustering), veri özetleme (Data Summarization), sınıflama kurallarının öğrenilmesi (Classification Rules), bağımlılık ağlarının (Dependency Networks) bulunması, değişikliklerin analizi (Analysing Changes) ve anomali tespiti (Detecting Anomaly) gibi farklı bir çok teknik yaklaşımı kapsamaktadır [1].

Gartner Group tarafından ise VM, istatistik ve matematiksel tekniklerle birlikte örüntü tanıma (Pattern Recognition) teknolojilerini kullanarak, depolama ortamlarında saklanmış bulunan veri yığınlarının elenmesi ile anlamlı yeni korelasyon, örüntü ve eğilimlerin keşfedilmesi süreci olarak tanımlanmaktadır [12]. Bir başka tanımda; veri ambarlarındaki tutulan çok çeşitli verilere dayanarak daha önce keşfedilmemiş bilgileri ortaya çıkarmak, bunları karar vermek ve eylem planını gerçekleştirmek için kullanma sürecidir. Bu noktada kendi başına bir çözüm değil, ancak çözüme ulaşmak için verilecek karar sürecini destekleyen, problemi çözmek için gerekli olan bilgileri sağlamaya yarayan bir araçtır. VM, zeki yöntemler aracılığı ile büyük miktarda veriden anlamlı bilgilerin çıkarılması sürecidir. Daha sonra, çıkarılan örüntüler, içlerinden yararlı olanların belirlenmesi için değerlendirilir.

Alataş ve Akın tarafından yapılan tanım ise şöyledir; eldeki verilerden üstü kapalı, çok net olmayan, önceden bilinmeyen ancak potansiyel olarak kullanışlı bilginin çıkarılmasıdır. Diğer bir ifadeyle, verilerin içerisindeki desenlerin, ilişkilerin, değişimlerin, düzensizliklerin, kuralların ve istatistiksel olarak önemli olan yapıların yarı otomatik olarak keşfedilmesidir. Veri madenciliğinde keşfedilecek kurallar veritabanının özelliklerine ve kuralların kullanılışına göre farklı tekniklerle bulunur. Bunlardan bazıları sınıflama, kümeleme, birliktelik kuralları, ardışık örüntüler, zaman serisi analizi, tahmin etme, tanımlama ve görselleştirme gibi tekniklerdir [3].

Dünyada 1960'larda veri toplama sistemleri, 1970'lerde ise ilişkisel veri tabanları kullanılmaya başlanmış, 1980'lerde ise ilişkisel veri tabanları popüler olmaya başlamış, 1990 ve 2000'lerde ise bilgisayar sistemlerindeki teknolojik gelişmelere paralel ilişkisel veri tabanlarında tutulan veri depoları kullanılmaya başlanmıştır. Bugün, dünya gündeminde de veri madenciliğinin, veri ambarlarının, multimedya ve web veri tabanlarının yaygınlaşmaya başladığını görüyoruz. VM, son 10 yılda dünyada hızla yaygınlaşmaya başlayan bir disiplinler arası disiplin olarak göze çarpmaktadır. Günümüzde artan veri sayısı, bilgisayar kullanımının yaygınlaşması ve bilgi toplumu olma yolundaki adımlar bu disiplinin daha fazla gündeme gelmesine neden olmaktadır. Yurt dışında yaygın bir şekilde kullanılan veri madenciliği, ülkemizde daha yeni yeni tanınmaya ve kullanılmaya başlanmıştır.

BÖLÜM 3. VERİ MADENCİLİĞİNİN KULLANIM AMACI VE KULLANIM ALANLARI

3.1. Veri Madenciliğinin Kullanım Amaçları

İstatistiğin amacı nasıl ana kütle hakkında anlamlı bilgiler elde etmek ve yorum yapmaksa veri madenciliğinin amacı da anlamlı bilgiler elde etmek ve bunu eyleme dönüştürecek kararlar için kullanmaktır. Buradaki temel amaç, değişkenler arasındaki ilişkilerden çok, geleceğe yönelik sağlıklı öngörülerin üretilmesidir. Bu anlamda VM, özbilginin keşfedilmesi anlamında bir "kara kutu" bulma yaklaşımı olarak kabul edilmektedir ve bu doğrultuda yalnızca keşifsel veri analizi tekniklerini değil, sinir ağı tekniklerinden hareketle geçerli öngörüler yapmak ve öngörülen değişkenler arasındaki ilişkilerin belirlenmesi mümkün olduğu için aynı zamanda sinir ağı tekniklerini de kullanmaktadır.

Yöntemin işletmelerde kullanımı sonucunda sağlanabilecek faydalar aşağıdaki gibi özetlenebilir. Bir işletme kendi müşterisiyken rakibine giden müşterilerle ilgili analizler yaparak rakiplerini tercih eden müşterilerinin özelliklerini elde edebilir ve buradan hareketle gelecek dönemlerde kaybetme olasılığı olan müşterilerin kimler olabileceği yolunda tahminlerde bulunarak onları kaybetmemek, kaybettiklerini geri kazanmak için farklı stratejiler geliştirebilir.

- Mevcut müşterilerin işletme tarafından daha iyi tanınmasını sağlayabilir. Özellikle finans sektöründe mevcut müşterilerinin segmentlere ayrılarak çıkarılacak kredi risk davranış modellerinin yeni başvuruda bulunan müşterilere uygulanmasını sağlayarak riski minimize edebilir. Bir anlamda kredi risk skorlamasının altyapısının oluşturulmasında kullanılabilir.
- Mevcut müşterilerin ödeme performansları incelenerek kötü ödeme performansı gösteren müşterilerin ortak özellikleri belirlenerek, benzer

özelliklere sahip tüm müşteriler için yeni risk yönetim politikaları oluşturulabilir.

- En karlı mevcut müşteriler belirlenerek, potansiyel müşteriler arasından en karlı olabilecekler belirlenebilir. Karlı müşteriler tespit edilerek onlara özel kampanyalar uygulanabilir. En masraflı müşteriler daha masrafsız müşteri haline dönüştürülebilir. Örneğin en çok bankacılık işlemi yapanlar ortaya çıkarılıp bunlar şube bankacılığı yerine daha masrafsız internet bankacılığına yönlendirilebilir.
- Mevcut müşteriyi tanıyarak işletmelerin müşteri ilişkileri yönetimlerinde düzenleme ve geliştirmeler yapılabilir. Bu sayede firmanın müşterilerini daha iyi tanıyarak müşteri gibi düşünme kapasitelerinin arttırılması sağlanabilir. Bununda işletmelere pazarda avantaj sağlayacağı unutulmamalıdır.
- Geçmiş ve mevcut yapı analiz edilerek geleceğe yönelik tahminlerde bulunulabilir. Özellikle ciro, karlılık, Pazar payı gibi analizlerde veri madenciliği çok rahat kullanılabilir.
- Mevcut müşteriler üzerinde firma ürünlerinin çapraz satış kapasitesinin arttırılması sağlanabilir. Mesela firmanın X ürününü alan müşterilerin çok büyük bir bölümünün Y ürününü de aldıklarını biliyorsak, buna yönelik pazarlama stratejileri geliştirilebilir.
- Piyasada oluşabilecek değişikliklere mevcut müşteri portföyünün vereceği tepkinin firma üzerinde oluşturabileceği etkinin tespitinde kullanılabilir.
- Operasyonel süreçte oluşabilecek olası kayıpların veya suiistimallerin tespitinde kullanılabilir.
- Kurum teknik kaynaklarının en optimal şekilde kullanılmasını sağlamakta kullanılabilir.
- Firmanın finansal yapısının, makro ekonomik değişmeler karşısındaki duyarlılığı ve oluşabilecek risklerin tespitinde kullanabilir.
- Günümüzde var olan yoğun rekabet ortamında firmaların hızlı ve kendisi için en doğru kararı almalarını sağlayabilir [26].

3.2. Veri Madenciliğinin Kullanım Alanları

Ülkemizde son yıllarda yeni yeni tanınmaya başlayan VM kavramı, Avrupa ve

Kuzey Amerika ülkelerinde birbirinden çok farklı alanlarda kullanıldığı görülmektedir. Pazarlama ve satış alanında, hedef pazarların tespitinde, müşteri ilişkilerinin yönetiminde, sepet analizinde, çapraz satışlarda, Pazar segmentasyonlarında ve müşteri hatırlamada sık sık veri madenciliğinden yararlanılmaktadır. Veri kaynaklarını işlemek için müşteri kartı bilgilerinin kaydedilmesinde, müşteri şikayetlerinin incelenmesinde, e-ticarette oldukça büyük işlevlere sahiptir. Diğer taraftan satış kampanyalarının, verimlilik analizlerinin yapılması, reklamcılık, indirim kartları ve bonuslandırmaları, karlılığın artırılması gibi daha bir çok kullanım alanı bulunmaktadır.

Sayılan bu kullanım alanlarının yanında, astronomi, biyoloji, finans, sigorta, tıp gibi bir çok başka alanda da uygulanmaktadır. Son 20 yıldır Amerika Birleşik Devletleri'nde çeşitli veri madenciliği algoritmalarının gizli dinlemeden, vergi kaçakçılıklarının ortaya çıkarılmasına kadar çeşitli uygulamalarda da kullanıldığı görülmektedir.

Özellikle, son yıllarda, risk analizi ve yönetiminde de, doğru ve etkin kredi kararı verebilme, kredi geri ödemesi yapmamaya meyilli müşterileri belirleme, risk derecelendirme, finansal işlemlerde sahtekarlığa yönelik eğilimleri izleme, ekonomik ve finansal yatırımları kararlaştırma, iflas / başarısızlık tahmini gibi alanlarda da yaygın olarak kullanılmaya başlamıştır.

Görüldüğü gibi veri madenciliği teknikleri çok çeşitli alanlarda kullanılmaktadır. Bu uygulama alanları ana başlıklar altında aşağıdaki gibi özetlenebilir [12].

3.2.1. Pazarlama - perakendecilik

Mevcut müşterilerin elde tutulması için geliştirilecek pazarlama stratejilerinin oluşturulmasında, Müşterilerin demografik özellikleri arasındaki bağlantıların kurulmasında, Müşterilerin satın alma örüntülerinin belirlenmesinde, Posta kampanyalarında cevap verme oranının artırılmasında, Pazar sepeti çapraz satış analizlerinde, Müşteri ilişkileri yönetimi ve müşteri değerlendirme, Satış tahmini ve satış noktası veri analizlerinde, Tedarik ve mağaza yerleşim optimizasyonunda

kullanılmaktadır.

3.2.2. Bankacılık - sigortacılık - borsa

Farklı finansal göstergeler arasındaki gizli korelasyonların bulunmasında, Kredi kartı dolandırıcılıklarının tespitinde, Kredi kartı harcamalarına göre müşteri gruplarının belirlenmesinde, Kredi taleplerinin değerlendirilmesinde, Risk analizi ve risk yönetiminde, Yeni poliçe talep edecek müşterilerin tahmin edilmesinde, Sigorta dolandırıcılıklarının tespitinde, Hisse senedi fiyat tahmininde, Genel piyasa analizleri, Alım-satım stratejilerinin optimizasyonunda kullanılmaktadır.

3.2.3. Telekomünikasyon

Kalite ve iyileştirme analizlerinde, Hisse tespitlerinde, Hatların yoğunluk tahminlerinde kullanılabilir.

3.2.4. Sağlık ve ilaç

Test sonuçlarının tahmininde, Ürün geliştirmede, Tıbbi teşhiste, Tedavi sürecinin belirlenmesinde.

3.2.5. Endüstri – mühendislik

Kalite kontrol analizlerinde, Lojistik, Üretim süreçlerinin optimizasyonunda, Ampirik veriler üzerinde modeller kurarak bilimsel ve teknik problemlerin çözümlenmesinde kullanılabilmektedir.

BÖLÜM 4. VERİ MADENCİLİĞİ SÜRECİ

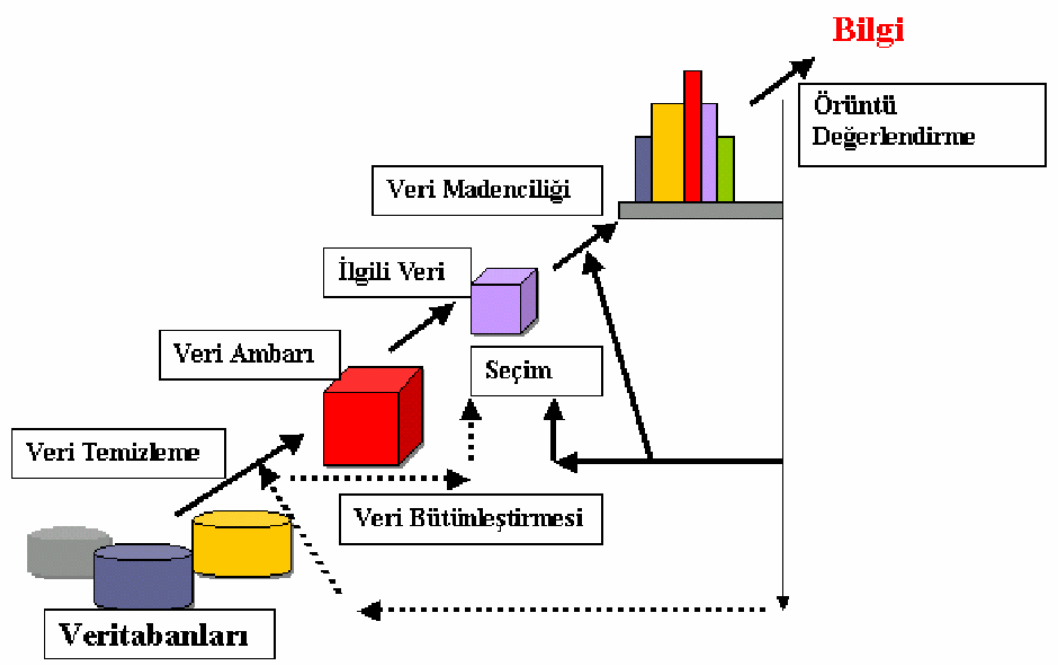
Ne kadar etkin olursa olsun hiçbir veri madenciliği algoritmasının üzerinde inceleme yapılan işin ve verilerin özelliklerinin bilinmemesi durumunda fayda sağlanması mümkün değildir. Bu nedenle aşağıda tanımlanan tüm aşamalardan önce, iş ve veri özelliklerinin öğrenilmesi başarının ilk ve temel şartı olacaktır.

Başarılı bir veri madenciliği projelerinde sırasıyla; Problemin Tanımlanması, Verilerin Hazırlanması, Modelin Kurulması ve Değerlendirilmesi, Modelin Kullanılması, Modelin izlenmesi adımları yer almaktadır. Aşağıdaki Şekil-1'de bu süreç gösterilmiştir [3].

4.1. Problemin Tanımlanması

Veri madenciliği çalışmalarında başarılı olmanın en önemli şartı, projenin hangi işletme amacı için yapılacağının açık bir şekilde tanımlanmasıdır. İlgili işletme amacı, işletme problemi üzerine odaklanmış ve açık bir dille ifade edilmiş olmalı, elde edilecek sonuçların başarı düzeylerinin nasıl ölçüleceği tanımlanmalıdır. Ayrıca yanlış tahminlerde katlanılacak olan maliyetlere ve doğru tahminlerde kazanılacak faydalara ilişkin tahminlere de bu aşamada yer verilmelidir.

Bu aşamada mevcut iş probleminin nasıl bir sonuç üretilmesi durumunda çözüleceğinin, üretilecek olan sonucun fayda-maliyet analizinin diğer bir ifadeyle üretilen bilginin işletme için değerinin doğru analiz edilmesi gerekmektedir. Analistin işletmede üretilen sayısal verilerin boyutlarını, proje için yeterlilik düzeyinin iyi analiz edilmesi gerekmektedir. Ayrıca analistin işletme konusu hakkındaki iş süreçlerinin de iyi analiz etmesi gerekmektedir [3].



Şekil 4.1 Veri Madenciliği Süreci

4.2. Verilerin Hazırlanması

Burada kullanılacak verinin kalitesi sonuçları da etkileyeceğinden kullanılacak verilerin öncelikle ön işlemden geçirilmesi büyük bir önem taşımaktadır. Sonuçta kaliteli verilerden ancak kaliteli çıktılar elde edilebilecektir. Bu nedenle verilerin kalitesini arttırmanın yolu, verilerin ön işlemden geçirilmesidir.

Modelin kurulması aşamasında ortaya çıkacak sorunlar, bu aşamaya sık sık geri dönülmesine ve verilerin yeniden düzenlenmesine neden olacaktır. Bu durum verilerin hazırlanması ve modelin kurulması aşamaları için, bir analistin veri keşfi sürecinin toplamı içerisinde enerji ve zamanının %50 - %85'ini harcamasına neden olmaktadır. Bu aşamada firmanın mevcut bilgi sistemleri üzerinde ürettiği sayısal bilginin iyi analiz edilmesi, veriler ile mevcut iş problemi arasında ilişki olması gerektiği de unutulmamalıdır. Proje kapsamında kullanılacak sayısal verilerin, hangi iş süreçleri ile elde edildiği de bu veriler kullanılmadan analiz edilmelidir. Bu sayede analist veri kalitesi hakkında fikir sahibi olabilir. Verilerin hazırlanması aşaması kendi içerisinde toplama, değer biçme, birleştirme ve temizleme, seçme ve dönüştürme adımlarından oluşmaktadır [3].

4.3. Modelin Kurulması ve Değerlendirilmesi

Tanımlanan problem için en uygun modelin bulunabilmesi, olabildiğince çok sayıda modelin kurularak denenmesi ile mümkündür. Bu nedenle veri hazırlama ve model kurma aşamaları, en iyi olduğu düşünülen modele varılıncaya kadar yenilenen bir süreçtir.

Bir modelin doğruluğunun test edilmesinde kullanılan en basit yöntem basit geçerlilik testidir. Bu yöntemde tipik olarak verilerin %5 ile %33 arasındaki bir kısmı test verileri olarak ayrılır ve kalan kısım üzerinde modelin öğrenimi gerçekleştirildikten sonra, bu veriler üzerinde test işlemi yapılır. Bir sınıflama modelinde yanlış olarak sınıflanan olay sayısının, tüm olay sayısına bölünmesi ile hata oranı, doğru olarak sınıflanan olay sayısının tüm olay sayısına bölünmesi ile ise doğruluk oranı hesaplanır. (Doğruluk Oranı = 1 – Hata Oranı)

Sınırlı miktarda veriye sahip olunması durumunda, kullanılabilecek diğer bir yöntem ise çapraz geçerlilik (Cross Validation) testidir. Bu yöntemde veri kümesi tesadüfi olarak iki eşit parçaya ayrılır.

Bir diğer önemli değerlendirme kriteri ise modelin anlaşılabilir olmasıdır. Bazı uygulamalarda doğruluk oranlarındaki küçük artışlar çok önemli olsa da, bir çok işletme uygulamasında ilgili kararın niçin verildiğinin yorumlanabilmesi çok daha büyük önem taşıyabilir [12].

4.4. Modelin Kullanılması

Kurulan ve geçerliliği kabul edilen model doğrudan bir uygulama olabileceği gibi, bir başka uygulamanın alt parçası olarak da kullanılabilir. Kurulan modeller risk analizi, kredi değerlendirme, dolandırıcılık tespiti gibi işletme uygulamalarında doğrudan kullanılabileceği gibi, promosyon planlaması simülasyonuna entegre edilebilir veya tahmin edilen envanter düzeyleri yeniden sipariş noktasının altına düştüğünde, otomatik olarak sipariş verilmesini sağlayacak bir uygulamanın içine de gömülebilir [12].

4.5. Modelin İzlenmesi

Zaman içerisinde bütün sistemlerin özelliklerinde ve dolayısıyla ürettikleri verilerde ortaya çıkan değişiklikler, kurulan modellerin sürekli olarak izlenmesini ve gerekiyorsa yeniden düzenlenmesini gerektirecektir. Tahmin edilen ve gözlenen değişkenler arasındaki farklılığı gösteren grafikler model sonuçlarının izlenmesinde kullanılan yararlı bir yöntemdir [3].

BÖLÜM 5. VERİ MADENCİLİĞİ MODELLERİ VE KULLANILAN ALGORİTMALAR

Veri madenciliginde kullanılan modeller, tahmin edici *(Predictive)* ve tanımlayıcı *(Descriptive)* olmak üzere iki ana baslık altında incelenmektedir. Tahmin edici modellerde, sonuçları bilinen verilerden hareket edilerek bir model gelistirilmesi ve kurulan bu modelden yararlanılarak sonuçlan bilinmeyen veri kümeleri için sonuç değerlerin tahmin edilmesi amaçlanmaktadır. Örneğin, bir banka önceki dönemlerde vermis olduğu kredilere iliskin gerekli tüm verilere sahip olabilir. Bu verilerde bağımsız değişkenler kredi alan müşterinin özellikleri, bağımlı değişken değeri ise kredinin geri ödenip ödenmediğidir. Bu verilere uygun olarak kurulan model, daha sonraki kredi taleplerinde müşteri özelliklerine göre verilecek olan kredinin geri ödenip ödenmeyeceğinin tahmininde kullanılmaktadır.

Tanımlayıcı modellerde ise karar vermeye rehberlik etmede kullanılabilecek mevcut verilerdeki örüntülerin tanımlanması sağlanmaktadır . X/Y aralığında geliri ve iki veya daha fazla arabası olan çocuklu aileler ile, çocugu olmayan ve geliri X/Y aralığından düşük olan ailelerin satın alma örüntülerinin birbirlerine benzerlik gösterdiğinin belirlenmesi tanımlayıcı modellere bir örnektir.

Veri madenciligi modellerini gördükleri işlevlere göre,
1- Sınıflama *(Classification)* ve Regresyon *(Regression)*
2- Kümeleme *(Clustering)*
3- Birliktelik Kuralları *(Association Rules)*
olmak üzere üç ana baslık altında incelemek mümkündür. Sınıflama ve regresyon modelleri tahmin edici, kümeleme ve birliktelik kuralları modelleri tanımlayıcı modellerdir [2].

Şimdi bu algoritmaların yapılarını inceleyelim.

5.1. Sınıflama ve Regresyon Algoritması

Sınıflama ve regresyon, önemli veri sınıflarını ortaya koyan veya gelecek veri eğilimlerini tahmin eden modelleri kurabilen iki veri analiz yöntemidir. Sınıflama kategorik değerleri tahmin ederken, regresyon süreklilik gösteren değerlerin tahmin edilmesinde kullanılır. Örneğin, bir sınıflama modeli banka kredi uygulamalarının güvenli veya riskli olmalarını kategorize etmek amacıyla kurulurken, regresyon modeli geliri ve mesleği verilen potansiyel müşterilerin bilgisayar ürünleri alırken yapacakları harcamaları tahmin etmek için kurulabilir.

Sınıflandırma algoritmaları 3 grupta toplanabilir. Bunlar:
a. Karar Ağaçları,
b. Yapay sinir ağı tabanlı sınıflandırma algoritmaları,
c. İstatistiksel sınıflandırma algoritmalarıdır.

Karar ağacı yöntemi karar verme problemlerinde sıkça kullanılan temel bir yöntem olduğundan ileride bir örnekle de üzerinde duracağız.

5.1.1. Karar ağacı gösterimi ile sınıflandırma

Karar ağaçları (decision trees*),* sınıflandırma, kümeleme ve tahmin modellerinde kullanılan bir tahmin tekniğidir. Sorunla ilgili araştırma alanını alt gruplara ayırmak için kullanılır. Karar ağaçlarında kök ve her düğüm bir soruyla etiketlenir. Düğümlerden ayrılan dallar ise ilgili sorunun olası yanıtlarını belirtir. Her dal düğümü de söz konusu sorunun çözümüne yönelik bir tahmini temsil eder. Karar ağaçları, üç bölümden oluşan bir modeldir.

1. Tanımdaki gibi bir karar ağacı,
2. Ağacı oluşturacak bir algoritma,
3. Ağacı veriye uygulayacak ve söz konusu sorunu çözecek bir algoritma .

Karar ağaçları, eğitici örnekteki veriyi sınayan bir algoritma aracılığıyla gerçekleştirilir yada alanın bir uzmanı tarafından oluşturulur. Karar ağacı tekniklerinin çoğu, birbirlerinden ağacın nasıl oluşturulduğuyla ayrılır [26].

5.1.1.1. Karar ağacı gösterimi için "Tenis Oynama Örneği"

Tenis oynama örneği, hava durumu, sıcaklık, nem ve rüzgar gibi değişkenlere (attributes) bağlı olarak tenis oynama kararının verilmesidir.

Tablo 5.1 Örnek veri seti

Hava	Sıcaklık	Nem	Rüzgar	Tenis Oynansın mı?
Güneşli	Sıcak	Yüksek	Rüzgarsız	Hayır
Güneşli	Sıcak	Yüksek	Rüzgarlı	Hayır
Bulutlu	Sıcak	Yüksek	Rüzgarsız	Evet
Yağmurlu	Ilıman	Yüksek	Rüzgarsız	Evet
Yağmurlu	Soğuk	Normal	Rüzgarsız	Evet
Yağmurlu	Soğuk	Normal	Rüzgarlı	Hayır
Bulutlu	Soğuk	Normal	Rüzgarlı	Evet
Güneşli	Ilıman	Yüksek	Rüzgarsız	Hayır
Güneşli	Soğuk	Normal	Rüzgarsız	Evet
Yağmurlu	Ilıman	Normal	Rüzgarsız	Evet
Güneşli	Ilıman	Normal	Rüzgarlı	Evet
Bulutlu	Ilıman	Yüksek	Rüzgarlı	Evet
Bulutlu	Sıcak	Normal	Rüzgarsız	Evet
Yağmurlu	Ilıman	Yüksek	Rüzgarlı	Hayır

Karar Ağacı Gösterimi

Karar ağacı gösteriminde amaç minimum dal sayısı elde etmektir. Bunun için geliştirilen üç metot bulunmaktadır.Bunlar:

1. Bilgi Kazanç Metodu (ID3/C45)

2. Bilgi Kazanç Oranı
3. Gini İndeksi

Bu metotlardan en yaygın olarak kullanılan Bilgi Kazanç Metodu(ID3) açıklanacaktır.

Bilgi Kazanç Metodu: Minimum dal sayısı için önce hangi değişkenin seçileceği belirlenir. Bunun için temel kriter en fazla bilgi kazanç değerine sahip olan dalın seçilmesidir. Bu işlemler diğer değişkenler için tekrar edilerek minimum dal sayısına sahip olan son karar ağacı elde edilir.

Bilgi Değeri: Örneğin bilgi değeri için karar değişkeninin entropisi hesaplanır. Karar değişkeni iki sınıfa (Evet, Hayır) sahiptir. p ve n değerleri her sınıfa ait eleman sayısı olmak üzere örneğin bilgi değeri:

$$I(p,n) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$$

Evet = 9

Hayır = 5

$$I(9,5) = -[\frac{9}{14}\log 2(\frac{9}{14}) + \frac{5}{14}\log 2(\frac{5}{14})] = 0{,}940$$

Entropi: Her bir değişken için entropi değerleri hesaplanırken önce alt sınıfların bilgi değeri hesaplanır. pi ve ni değerleri her alt sınıfa ait eleman sayısıdır.

$$E(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p + n} I(p_i, n_i)$$

Hava = Güneşli, (2 ve 3) (Evet =2 , Hayır =3)

$$I([2,3]) = \text{entropy}(2/5,3/5) = -\frac{2}{5}\log(\frac{2}{5}) - \frac{3}{5}\log(\frac{3}{5}) = 0.971$$

Hava = Bulutlu, (4) (Evet =4)

Aynı sınıfa ait olduğu için bilgi değeri = 0 dır.

Hava = Yağmurlu, (3 ve 2) (Evet=3 , Hayır =2)

$$I([3,2]) = \text{entropy}(3/5,2/5) = -\frac{3}{5}\log(\frac{3}{5}) - \frac{2}{5}\log(\frac{2}{5}) = 0.971$$

Değişkenin entropi değeri:

$$E([3,2],[4,0],[3,2]) = (5/14)\times 0.971 + (4/14)\times 0 + (5/14)\times 0.971 = 0.693$$

E(Hava) = 0,693

Kazanç: Sınıfın bilgi kazanç değeri, bilgi değerinden entropi değeri çıkarılarak elde edilir:

$$Kazanç(A) = I(p,n) - E(A)$$

Kazanç (Hava) = I(9,5)- I([2,3],[4,0],[3,2])= 0,940- 0,693 = 0,247

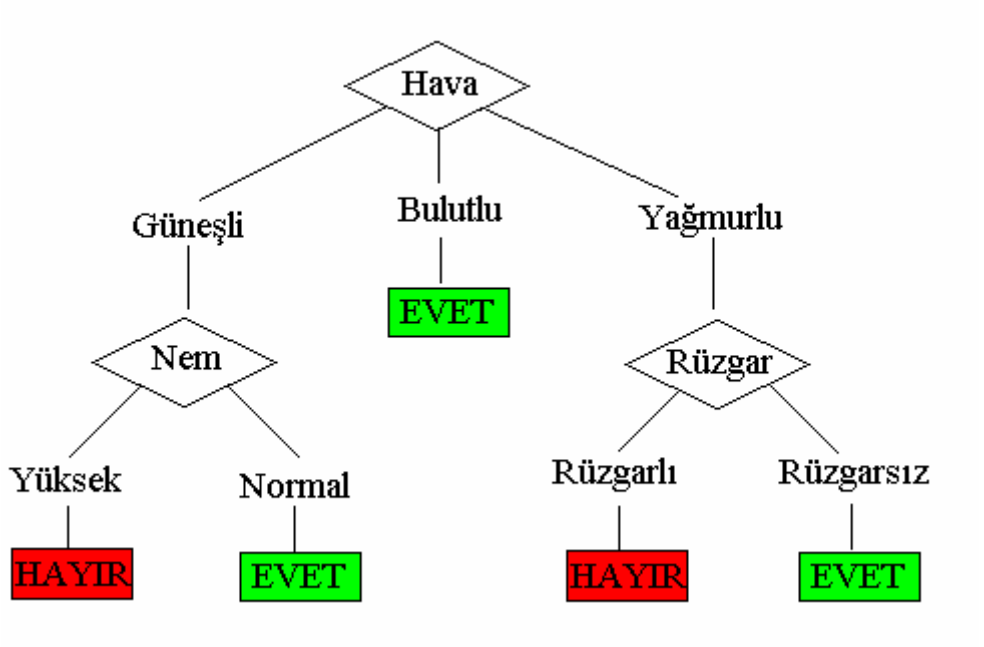
Diğer sınıfların kazanç değerleri şöyledir:

Kazanç (Sıcaklık) = 0,029

Kazanç (Nem) = 0.152

Kazanç (Rüzgar) = 0,048

En yüksek kazanç değerine sahip olan (0,247) Hava değişkeni karar ağacına yerleştirilerek ilk karar ağacı çizilir. Bu işlem minimum dal sayısına ulaşılıncaya kadar diğer değişkenler için de uygulanarak son karar ağacı elde edilir.



Şekil 5.1 Karar Ağacı

Karar ağacından kural çıkarmanın en basit yolu her bir dalı takip ederek "Eğer-…..ise o zaman ….. " kural kalıpları çıkarmaktır. Her kural kökten uca kadar giden yolu takip eder. Uç noktada bulunan değer tahmin yada karar vermede kullanılır. Kurallar kolay anlaşılabilirdir. Tenis oynama örneği için şu kurallar çıkarılabilir:

- Eğer *Hava* = Güneşli ve *Nem* = Yüksek ise o zaman tenis oynama.
- Eğer *Hava* = Güneşli ve *Nem* = Normal ise o zaman tenis oyna.
- Eğer *Hava* = Bulutlu ise o zaman tenis oyna.
- Eğer *Hava* = Yağmurlu ve *Rüzgar* = Rüzgarlı ise o zaman tenis oynama.
- Eğer *Hava* = Yağmurlu ve *Rüzgar* = Rüzgarsız ise o zaman tenis oyna.

5.2. Kümeleme (*Clustering*) Algoritması

Kümeleme analizi, nesnelerin altdizinlere gruplanmasını yapan bir işlemdir. Böylece nesneler, örneklenen kitle özelliklerini iyi yansıtan etkili bir temsil gücüne sahip olmuş olur. Sınıflamanın aksine, yeniden tanımlanmış sınıflara dayalı değildir. Kümeleme, bir denetimsiz öğrenme (unsupervised learning) yöntemidir.

Hiyerarşik olmayan kümelemede, kümeler arasında ilişki bulunmamaktadır. Örneğin, ‘k-ortalamalar’, hiyerarşik olmayan bir kümeleme algoritmasıdır.

Hiyerarşik kümelemede, her kümede veri nesnelerini içerecek bir bağlantı kurulur. Hangi yöntem olursa olsun kümeler birbirine benzer özellik gösteren nesnelerden oluşturulur. Böylece kümeler kendi içinde aynı özelliği taşıyan nesneleri içermiş olur. Manhattan ve Euclid uzaklık fonksiyonları çoğunlukla benzerliklerin bulunmasında kullanılır. Uzaklık fonksiyonunun sonucu yüksek bir değer ise az benzerlik, düşük bir değer ise çok benzerlik olduğunu ifade eder. P-boyutlu veri nesneleri i:(x_{i1},x_{i2},.....,x_{ip}), j:(x_{j1},x_{j2},.....,x_{jp}) için aşağıda verilen uzaklık fonksiyonları tanımlanabilir.

Euclid uzaklık fonksiyonu:

$$d_{ij} = \sqrt{\left|x_{i1}-x_{j1}\right|^2 + \left|x_{i2}-x_{j2}\right|^2 + ... + \left|x_{ip}-x_{jp}\right|^2}$$

Manhattan uzaklık fonksiyonu:

$$d_{ij} = \left|x_{i1}-x_{j1}\right| + \left|x_{i2}-x_{j2}\right| + ... + \left|x_{ip}-x_{jp}\right|$$

Uzaklıkları karşılaştırmak için Euclid uzaklık fonksiyonu kullanıldığında, denklemin sağ tarafındaki kare kökü hesaplamak gereksizdir. Çünkü uzaklıklar her zaman pozitif sayılardır ve bundan dolayı, d_1 ve d_2 gibi iki uzaklık için, $\sqrt{d_1} > \sqrt{d_2}$ iken $d_1 > d_2$’dir. Bir nesnenin bazı özellikleri farklı ölçeklerde ölçülüyorsa, Euclid uzaklık fonksiyonu kullanılarak büyük ölçeklerle ölçülen nitelikler küçük bir ölçekte ölçülen niteliklere baskın gelebilir. Böyle bir sorundan kaçınabilmek için, nitelik değerleri çoğunlukla 0 ile 1 arasında normalleştirilir.

Veri kümeleri için uygulanacak uzaklık fonksiyonlarının verimleri farklı olabilir, bundan dolayı Euclide ve Manhattan’ın haricindeki uzaklık fonksiyonları bazı veri kümeleri için daha uygun olabilir [27].

VM'de kullanılmakta olan bir çok kümeleme algoritması vardır ve bunlar analiz edilecek olan verinin yapısına göre belirlenir. Kümeleme metotları genel olarak şunlardır:

- Bölümlendirme Metodu: n tane nesnenin olduğu veritabanında, nesneler mantıksal gruplara ayrılarak analiz edilir. Küçük ve orta boyutlu veritabanlarında birkaç grup olabilirken, veritabanının büyüklüğü arttığında daha çok grup oluşabilir. Gruplandırma yapılırken değişik kriterler değerlendirilebilir. Yapılan gruplandırma analizin kalitesine etki eder.
- Hiyerarşik Metot: Analiz etmeden önce nesneler, hiyerarşik bir yapıya göre düzenlenir. Veriyi hiyerarşik bir yapıya çevirmek için değişik yöntemler kullanılır. Bunların arasında BIRCH ve CURE yöntemleri bulunur.
- Yoğunluk Bazlı Metot: Birçok kümeleme yöntemi nesnelerin birbirleri arasındaki farklılıklarına göre kümeleme yaparken, bu metot nesnelerin yoğunluğuna göre gruplama yapar. Yoğunluktan kasıt, analiz edilen nesnelerin sayısıdır. Yoğunluk bazlı metotlara örnek olarak DBscan verilebilir.
- Grid Bazlı Metot: Nesneleri grid yapısı oluşturacak şekilde sayılarına göre sınıflandırır. Temel avantajı hızlı tamamlanması ve nesnelerin sayısından bağımsız olmasıdır. Bu tipteki metotlara örnek olarak Sting verilebilir.
- Model Bazlı Metot: Her küme için bir model belirlenir ve bu modele uyan veriler uygun kümeye yerleştirilir.

5.2.1. Bölümlendirme metodu

n tane nesnesi olan ve k sayıda küme tanımlanmış bir veritabanı düşünelim. Bu durumda bölümlendirme metodu tüm nesneleri k adet kümeye ayıracaktır. Kümeler, nesneler arasındaki benzersizliklere göre oluşturulur. En çok bilinen bölümlendirme algoritmaları k-means, k-medoids ve EM (Expectation Maximization) algoritmalarıdır.

5.2.1.1. K-means algoritması

d boyutlu metrik uzayda verilen n adet nesnenin aynı kümelerdeki nesneler diğer

kümelerdekine kıyasla daha benzer olacak şekilde k adet kümeye yerleştirilerek bölünmesinin yapılmasıdır.

Bu algoritma şu parametreleri alır:
k: kaç küme olacak
d: kaç nesne olacak

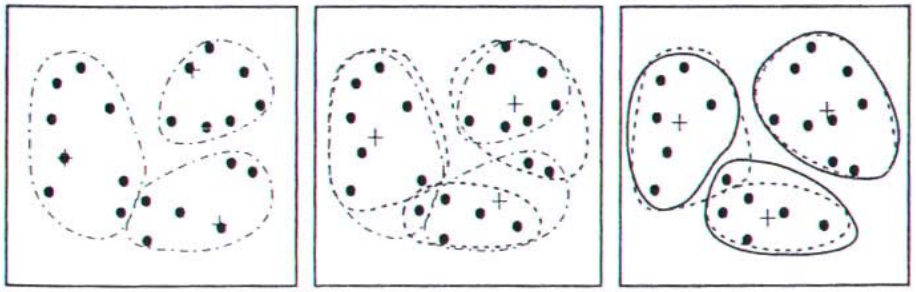
Bu nesneler benzersizliklerine göre kümeleme yapılıp geri verilir. Bu algoritmada kümeler arasındaki benzerlik düşük olur.

Bu algoritma öncelikle rasgele şekilde k tane nesne seçer. Bunların her birinin orta değeri kendisidir. Kalan nesnelerin tümünü bu seçilen nesnelere yakın olanlara göre kümelere dahil eder ve her defasında yeni mean (orta değer) hesaplar.
Her nesnenin bir hata kriter değeri (E) vardır.
Algoritma: k-orta değer : k sayıda kümelendirme algoritması
Girişler: nesne sayısı (n) ve küme sayısı (k)
Çıkış: k sayıdaki minimum hata ile oluşturulmuş kümeler

Algoritma:

1. Kabaca n tane nesne seç,
2. Tekrarla,
3. Değişken benzerliklerine göre grupları oluştur ve her grup için bir ortalama değer hesapla bu ortalama değeri uygun olan kümelere yerleştir,
4. Yerleştirme bittikten sonra ortalama değerleri güncelle,
5. Bir değişiklik olmayana dek tekrarla.

Bu metot ölçeklendirilebilir bir metottur ve çok geniş veritabanları üzerinde de uygulanabilir. Çünkü karmaşıklığı oldukça azdır.

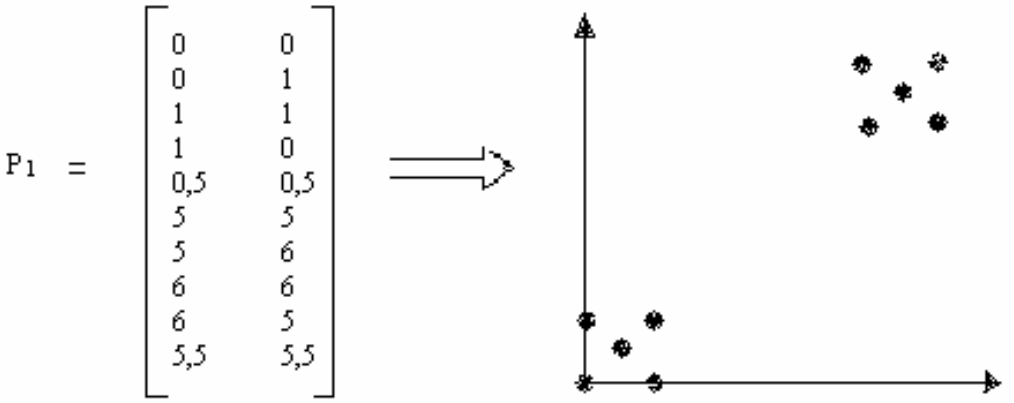


Şekil 5.2 Bir Nesne Setinin K-Means Metodu İle Kümelenmesi.

Her bir kümenin orta değeri "+" ile işaretlenmiştir.

Örnek: Girdi olarak veri kümesi Şekil 5.3'teki gibi verilmiş, k=2 seçilmiş ve uzaklık fonksiyonlarından Manhattan uzaklık fonksiyonu $|x_2-x_1|+|y_2-y_1|$ olarak belirlenmiştir.

Bu bilgilere göre hesaplama şöyledir:



Şekil 5.3 Örnek Veri

Adım1. İlk olarak k bölüm oluşturulur. İlk bölüm k başlangıç noktası, seçilerek oluşturulur. Bu k tohum (başlangıç) noktası ilk k nesne olabileceği gibi rasgele seçilen k kayıt da olabilir. Burada ilk iki nesne seçilir ve işlem başlatılır. Bizim örneğimiz için kümeler (bölümler) C_1 = {(0,0) } ve C_2={(0,1)} olur.

Adım2. Her kümede henüz sadece bir nokta olduğu için bu nokta kümenin merkezidir.

Adım3a. Her bir nesne ve küme merkezi için aralarındaki uzaklığı hesapla, nesneyi en yakın kümeye ata.

Örneğin, üçüncü nesne için:

Uzaklık(1,3)=$|1-0|+|1-0|=2$ ve Uzaklık(2,3)= $|1-0|+|1-1|=1$

bu nedenle nesne C_2'ye atanır.

Beşinci nesne her iki kümeden eşit uzaklıkta olduğu için, beşinci nesne rasgele bir kümeye yani C_1'e atandı. Her bir nokta için uzaklıklar hesaplandıktan sonra, kümeler aşağıdaki nesneleri içerir:

C_1={(0,0), (1,0), (0.5, 0.5)} ve

C_2={(0,1),(1,1),(5,5),(5,6),(6,6),(6,5),(5,5),(5.5, 5.5)}

Adım3b. Her bir küme için yeni küme merkezlerini hesapla.

C_1 için yeni merkez C_1=(0,5, 0,16), (0+1+0,5)/3=0,5, (0+0+0,5)/3=0,16

C_2 için yeni merkez C_2=(4,1, 4,2) için yeni merkez, (0+1+5+5+6+6+5,5)/7=4,1

(1+1+5+5+6+6+5,5)/7=4,2

Adım3a*. Yeni merkezler C_1=(0,5, 0,16) ve C_2=(4,1, 4,2), eski merkezler C_1=(0,0) ve C_2=(0,1)'den farklılık gösterir, bu nedenle döngü tekrarlanır. On nesne en yakın küme merkezine yeniden atanır, sonuç:

C_1={(0,0),(0,1),(1,1),(1,0),(0,5, 0,5)}

C_2={(5,5),(5,6),(6,6),(6,5),(5,5, 5,5)}

Adım3b*. Her bir küme için yeni küme merkezleri hesapla

C_1=(0,5, 0,5) için yeni merkez

C_2=(5,5, 5,5) için yeni merkez

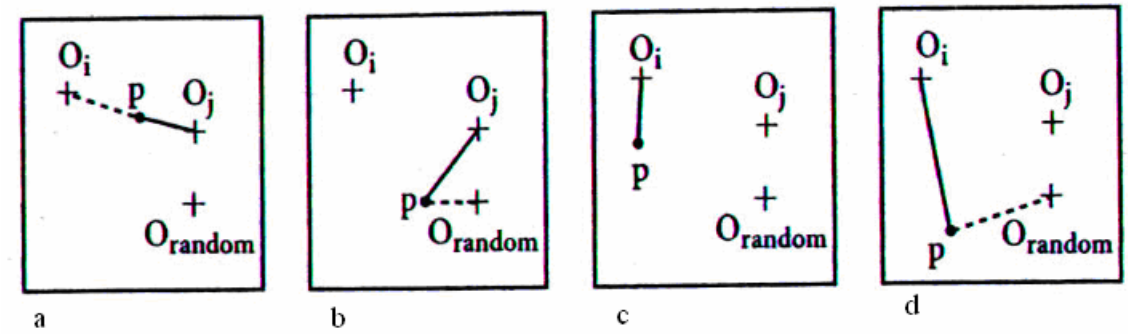
Adım3a**. Yeni merkezler C_1=(0,5, 0,5) ve C_2=(5,5, 5,5) eski merkezler C_1=(0,5, 0,16) ve C_2=(4,1, 4,2)'den farklılık gösterir, dolayısıyla döngü tekrarlanır. On nesneyi en yakın küme merkezine yeniden ata.

Adım3b**. Yeni küme merkezleri hesapla. Merkezler Adım3b*'dekiyle aynıdır. Bu nedenle algoritma sonlandırılır. Sonuç, 3b*'dekinin aynısıdır.

5.2.1.2. K-medoids algoritması

Çok yüksek değerdeki nesneler, küme dağılımını olumsuz etkiler. Çünkü k-means tüm değerlere karşı duyarlıdır. k-medoid de, k-means gibi tek tek hesaplamak yerine;

1) Her bir küme için kabaca bir temsilci nesne belirlenir (medoid)
2) Kalan her nesneyi bu medoid le karşılaştırır ve benzerliğine göre o nesne kümeye dahil edilir.
3) Bir kümedeki nesneyi alarak, daha yüksek kaliteyi elde edene dek kümeler arasında iteratif olarak yer değiştirme yapılır.



Şekil 5.4 k-medoids algoritması ile kümeleme

\+ küme merkezi

__ yer değiştirmeden önce

--- yer değiştirmeden sonra

Algoritma:

1. k tane nesne seç (medoid)
2. Tekrarla
3. Nesneleri onlara en yakın medoidlere at
4. Medoid olmayan rasgele bir nesne seçilir
5. Bu nesne bir medoidmiş gibi ele alınıp toplam performans hesaplanır
6. Eğer daha performanslı sonuç elde ediliyorsa diğeri yerine yeni medoid bu nesne olur (yer değiştirilir) (örneğin a kümesinden bir nesne seçerek b ve a kümeleriyle karşılaştır ve eğer daha kaliteli bir duruma gelecekse yer değiştir.)
7. Bir değişiklik olmayana dek tekrarla.

k-medoids, k-means'e göre çöp veriden daha az etkilenir [2].

5.2.1.3. EM algoritması

Em algoritması çok detaylı algoritmalardan biridir. Bu algoritma, bir veri setindeki parametre dağılımının maksimum olabilirlik tahminini yapan bir algoritmadır. Daha çok, tamamlanmamış veya kayıp veri içeren veri setlerine uygulanmaktadır.

w_i: kümeye dahil olma olasılığı olarak gösterilirse, A kümesinin ortalaması ve varyansı aşağıdaki gibidir.

$$\mu_A = \frac{w_1 x_1 + w_2 x_2 + ... + w_n x_n}{w_1 + w_2 + ... + w_n}$$ μ_A: A'nın ortalaması σ_A^2

$$\sigma_A^2 = \frac{w_1 (x_1 - \mu)^2 + w_2 (x_2 - \mu)^2 + ... + w_n (x_n - \mu)^2}{w_1 + w_2 + ... + w_n}$$ σ_A^2 : A'nın varyansı

k-ortalama algoritmasında, kayıtların kümeleri sabit olunca yineleme durur, Ancak, EM'de sabitleme yoktur, yaklaşma olması nedeniyle belli bir yere istenen aralıkta yaklaşılmaya başlanınca yinelemenin durması durumu söz konusudur. Genel olabilirliği (overall likelihood) hesaplayarak (5 tane parametre değeriyle veri kümesindeki veri değerlendirilerek) durma noktası bulunur. Genel olabilirlik her i nesnenin olasılıklarıyla çarpılarak hesaplanır.

$\prod_i \left(P_A P_r [x_i / A] + P_B P_r [x_i / B] \right)$ P_A: A kümesindeki eleman sayısının tüm eleman sayısına oranı.

$\Pr[A \mid x] = f(x; \mu_A, \sigma_A)\, p_A / \Pr[x]$: Verilen x nesnesinin A kümesinde olması olasılığıdır.

Burada olasılıklar A ve B kümeleri için verilmiştir. Bu olasılıklar normal dağılım fonksiyonundan hesaplanır (f(x ; μ, σ)).

Normal dağılım için olasılık yoğunluk fonksiyonu f(x) olarak verilmiştir.

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Genel olabilirlik kümelemenin başarısını belirtir ve EM'nin her yinelenmesinde artar. Tekrar f(x ; μ, σ) ile x'in belli değerlerindeki olasılıklarda eşitleme zorlukları vardır. Bundan ötürü, olasılık normalizasyon işlemleri uygulanır [28].

5.2.2. Hiyerarşik kümeleme algoritması

Hiyerarşik kümeleme nesnelerin yakınlık ilişkisine göre oluşturulan kümelerden bir ağaç inşa eder. Hiyerarşik kümeleme aşağıdaki özelliklere sahiptir:

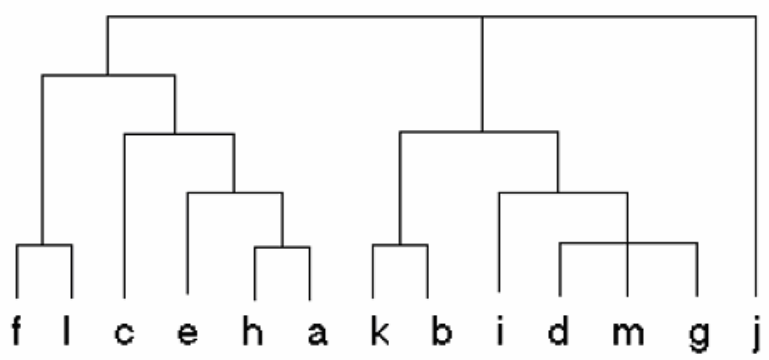
- Bir veri tabanını bir kaç kümeye ayrıştırır.
- Bu ayrıştırma dendogram adı verilen bir ağaç sayesinde yapılır.
- Bu ağaç, yapraklardan gövdeye doğru veya gövdeden yapraklara doğru kurulabilir.
- Aşağıdan-yukarıya yaklaşım (toplayıcı (agglomerative)) hiyerarşik kümeleme şu şekildedir:
- Her bir nesne için farklı bir grup oluşturarak başla,
- Bazı kurallara göre grupları birleştir: örn.; merkezler arasındaki uzaklık,
- Bir sonlandırma durumuna ulaşılıncaya kadar devam et.
- Yukarıdan aşağıya yaklaşımı (bölücü (divisive)):
- Aynı kümedeki bütün nesnelerle başla,
- Bir kümeyi daha küçük kümelere böl,
- Bir sonlandırma durumuna ulaşılıncaya kadar devam et.

Hiyerarşik kümelemeye örnek; BIRCH, CHAMELEON verilebilir. Hiyerarşik kümeleme algoritmalarından, Toplayıcı (Agglomerative) Hiyerarşik algoritmasına izleyen kesimde yer verilmiştir [16].

5.2.2.1. Toplayıcı *(Agglomerative)* hiyerarşik algoritması

Hiyerarşik algoritmalar toplayıcı (agglomerative) veya bölücü (divisive) olabilir. Toplayıcı hiyerarşik kümeleme algoritmaları her bir nesneyle ayrı bir grup olarak başlar. Bu gruplar, yalnızca bir grup kalana kadar veya belirli bir sonlandırma koşulu sağlanıncaya kadar, benzerliklere dayalı olarak birbirini takip edecek şekilde birleştirilir. n nesne için, n-1 birleştirme yapılır. Hiyerarşik algoritmalarla bir birleştirme yapıldığında geriye dönüş yoktur. Toplayıcı hiyerarşik kümeleme algoritmalarının az hesaplama maliyeti olmasına rağmen, yanlış birleştirme yapılması sorunlara yol açar. Bu yüzden, birleştirme noktalarının dikkatlice seçilmesi

gerekir. Hiyerarşik algoritmaların kümeleme kalitesini arttırabilmek amacıyla BRICH ve CURE gibi çok daha karmaşık algoritmalar geliştirilmiştir.



Şekil 5.5 Örnek Hiyerarşi Grafiği

Şekil 5.5, hiyerarşik bir kümeleme algoritmasından oluşturulabilen örnek bir dendogramı (hiyerarşik çizge) göstermektedir. Hiyerarşik kümelemede k-ortalamalar algoritmasından farklı olarak küme sayısı (k) belirtilmez. Hiyerarşi kurulduktan sonra, kullanıcı gerekli kümelerin sayılarını (1'den n'e) belirleyebilir. Hiyerarşinin en üst seviyesi bir kümeyi temsil etmektedir, veya k=1'dir. Daha fazla sayıda küme incelemek için hiyerarşinin alt kademelerine doğru inilmesi gerekir. Bir toplayıcı hiyerarşik algoritması Şekil 5.6'da verilmiştir.

Verilen:
Nesnelerin bir X kümesi $\{x_1,x_2,.....,x_n\}$ ve bir uzaklık fonksiyonu uzaklık(c_1,c_2)

1) SAYARAK YİNELE (i=1;i<=n;i++) {
  $C_i=\{x_i\}$
}
2. $C= \{ c_1,c_2,....,c_b\}$
3. *l*=n+1
4. Tüm (C.büyüklük>1) için {
a) Tüm C'deki c_i,c_j'ler için (c_{min1}, c_{min2})=*min* uzaklık(c_i,c_j)
b) C'den C_{min1} ve c_{min2}'yi çıkar
c) $\{c_{min1}, c_{min2}\}$'yi C'ye ekle
d) *l*=*l*+1
}

Şekil 5.6 Toplayıcı Hiyerarşik Algoritması

Bu algoritmadaki uzaklık fonksiyonu tek bir bağlantı (link) ve grup-ortalamasını içerirken, birçok yöntemle kümelerin benzerliğini belirleyebilir. Tek bağlantı, kümelerde yer alan herhangi iki nesne arasındaki en kısa uzaklık olacak şekilde, iki küme arasındaki uzaklığı hesaplayabilir. Grup-ortalama öncelikle grupta yer alan tüm nesnelerin ortalama değerlerini bulur (örn; küme) ve ortalama değerler arasındaki uzaklık gibi kümeler arasındaki uzaklığı hesaplar.

X'de (X: nesne kümesi) bulunan her bir nesne öncelikle tek bir nesneyi içeren bir küme oluşturulmasında kullanılır. C kümelerine eklenen bu kümeler başarılı bir şekilde yeni kümelerle birleştirilir. Kümelerin bir parçası birleştirildiğinde kümeler çifti, orijinal kümeler C'den çıkarılır. Dolayısıyla, X'teki tüm nesneleri içeren C'deki kümelerin sayısı bir küme kalıncaya kadar azalır. Kümelerdeki hiyerarşi C'nin iç içe veri kümelerini temsil edilir.

Örnek: Yukarıda tanımlanmış olan basit Toplayıcı Hiyerarşik Kümeleme Algoritması için girdi X veri kümesinin matrisi ve grafik gösterimi Şekil 5.3'teki gibi olsun. Kümeler arasındaki uzaklıkları hesaplayabilmek için Manhattan uzaklık fonksiyonu ve tek bağlantı yöntemi kullanılmıştır. X veri kümesi x_1'den x_{10}'a (n=10) eleman içermektedir. x_1=(0,0) dır.

Adım1. Öncelikle, X'in her bir x_i elemanı c_i kümesine yerleştirilir. c_i:C kümelerinin bir dizisinin bir üyesidir.

C= $\{\{x_1\},\{x_2\},\{x_3\},\{x_4\},\{x_5\},\{x_6\},\{x_7\},\{x_8\},\{x_9\},\{x_{10}\}\}$

Adım2.*I*=11 olsun

Adım3. (while loop döngüsünün birinci yinelemesi) C.büyüklük=10

- İki küme arasındaki minimum tek link uzaklığı 1 'dir. Bu iki yerde ortaya çıkar, c_2 ile c_{10} arasında ve c_3 ile c_{10} arasında.

Minimum fonksiyonlarımızın nasıl çalıştığına göre küme çiftlerini seçebiliriz. Burada keyfi olarak birincisi seçildi.

(c_{min1}, c_{min2})= (c_2, c_{10})

- *I* =10, $c_{11} = c_2 \cup c_{10} = \{\{x_2\}\{x_{10}\}\}$olduğunda
- C'den c_2 ve c_{10}'u çıkar,

- C'ye c_{11}'i ekle,

C = {{x_1},{x_3}, {x_4},{x_5},{x_6},{x_7}, {x_8},{x_9},{{x_2}, {x_{10}}}}

- $I = I + 1 = 12$

Adım3. (ikinci yineleme) C. büyüklük=9

- İki küme arasındaki minimum tek bağlantı uzaklığı 1 dir. Bu, c_3 ile c_{10} arasında meydana gelir, çünkü x_3 ile x_{11} arasındaki uzaklık 1'dir, burada x_{10} c_{11}'in içindedir,

(c_{min1}, c_{min2})= (c_3, c_{11})

- $c_{12} = c_3 \cup c_{11}$ = {{{x_2},{x_{10}}},{x_3}} olduğunda,

- C'den c_3 ve c_{11}'i çıkar,

- C'ye c_{12}'yi ekle,

C = {{x_1}, {x_4},{x_5},{x_6},{x_7}, {x_8},{x_9},{{{x_2}, {x_{10}}}, {x_3}}}

- $I = I + 1 = 13$

Adım3. (üçüncü yineleme) C. büyüklük=8

- (c_{min1},c_{min2}) = (c_1,c_{12})

- C = {{x_4},{x_5},{x_6},{x_7}, {x_8},{x_9},{{{{x_2}, {x_{10}}}, {x_3}},{x_1}}}

Adım3. (dördüncü yineleme) C. büyüklük=7

- (c_{min1},c_{min2}) = (c_4,c_8)

- C = {{x_5},{x_6},{x_7}, {x_9},{{{{x_2}, {x_{10}}}, {x_3}},{x_1}},{{x_4},{x_8}}}

Adım3. (beşinci yineleme) C. büyüklük=6

- (c_{min1},c_{min2}) = (c_5,c_7)

- C = {{x_6}, {x_9},{{{{x_2}, {x_{10}}}, {x_3}},{x_1}},{{x_4},{x_8}}, {{x_5},{x_7}}}

Adım3. (altıncı yineleme) C. büyüklük=5

- (c_{min1},c_{min2}) = (c_9,c_{13})

- C = {{x_6}, {{x_4},{x_8}}, {{x_5},{x_7}},{{{{{x_2}, {x_{10}}}, {x_3}},{x_1}},{x_9}}}

Adım3. (yedinci yineleme) C. büyüklük=4

- (c_{min1},c_{min2}) = (c_6,c_{15}) }}, {x_3}},{x_1}},{x_9}},{{x_6}, {{x_5},{x_7}}}}

Adım3. (sekizinci yineleme) C. büyüklük=3

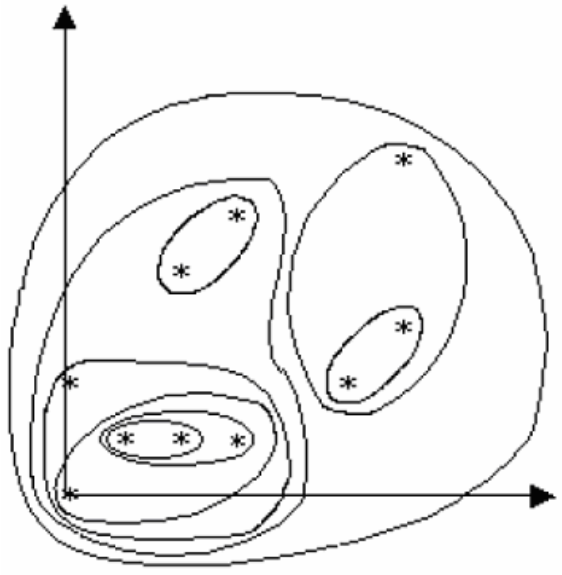
- $(c_{min1},c_{min2}) = (c_{14},c_{16})$
- C = { $\{\{x_6\}, \{\{x_5\},\{x_7\}\}\}, \{\{\{x_4\},\{x_8\}\}, \{\{\{\{x_2\}, \{x_{10}\}\}, \{x_3\}\},\{x_1\}\},\{x_9\}\}\}$

Adım3. (dokuzuncu yineleme) C. büyüklük=2

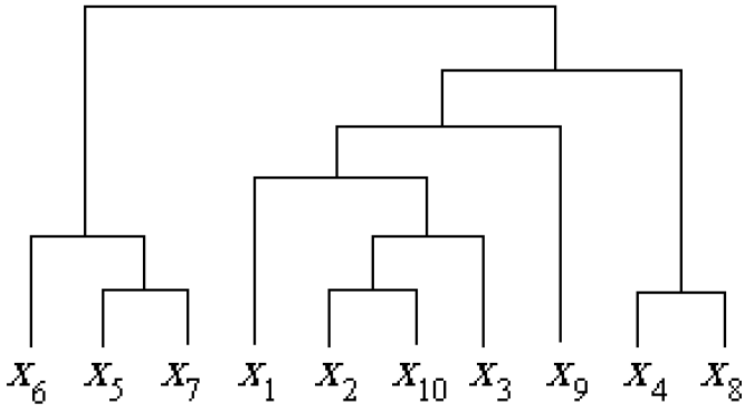
- $(c_{min1},c_{min2}) = (c_{17},c_{18})$
- C = $\{\{\{\{x_4\},\{x_8\}\}, \{\{\{\{x_2\}, \{x_{10}\}\}, \{x_3\}\},\{x_1\}\},\{x_9\}\}\}, \{\{x_6\}, \{\{x_5\},\{x_7\}\}\}\}$

Adım 3. (onuncu yineleme) C. büyüklük=1. Algoritma tamamlandı.

Bu algoritmadan oluşturulan küme Şekil 5.7'de görülmektedir. C'deki hiyerarşiden düzenlenen dendogram Şekil 5.8'de gösterilmiştir. Şekil 5.7'de girdi verisinin grafiği üzerinde birbirine yakın beliren noktalar hiyerarşide birbirine çok yakın gruplanmıştır [16].



Şekil 5.7 Toplayıcı Hiyerarşik Algoritması İçin Örnek Kümelerin Grafiği



Şekil 5.8 Toplayıcı Hiyerarşik Algoritması İçin Örnek Dendogram

5.2.3. Model tabanlı kümeleme metodları

Model tabanlı kümeleme metodları, verilen veri ile bazı matematiksel modeller arasında uygunluğu optimize etmeye çalışır. Bu metodlar genelde olasılık dağılımlarına göre varsayımlar üretir. Model tabanlı kümeleme metodlarının 2 ana yaklaşımı vardır: İstatiksel Yaklaşım veya nöral network yaklaşımı. Konumuzla alakalı olduğundan istatistiksel yaklaşımın üzerinde duracağız.

5.2.3.1. İstatiksel yaklaşım ve COBWEB

Kavramsal kümeleme, verilen etiketlenmemiş nesne kümeleri ve nesnelerdeki sınıf şemaların meydana gelmesi ile oluşan makine öğrenimli bir formdur. Benzer nesne gruplarını tanımlayan geleneksel kümelemenin tersine, kavramsal kümeleme bir adım öne geçip her grup için karakteristik tanımlamalar bulur. Kavramsal kümeleme iki adım işlemden oluşur: ilk kümeleme yapılır, sonra nitelendirme. Kümeleme kalitesi nesneler için tek fonksiyon değildir. Bununla beraber kavramların genellik ve basitlik gibi etkenleri de bir faktördür.

Tüm kavramsal kümeleme metodları istatiksel yaklaşımı benimsemektedir. İstatiksel yaklaşım, olasılık ölçütlerini kullanarak konseptleri veya kümeleri belirler. Artışsal konsept kümelenin popüler ve basit bir metodudur. Giriş değişkenleri kategorik

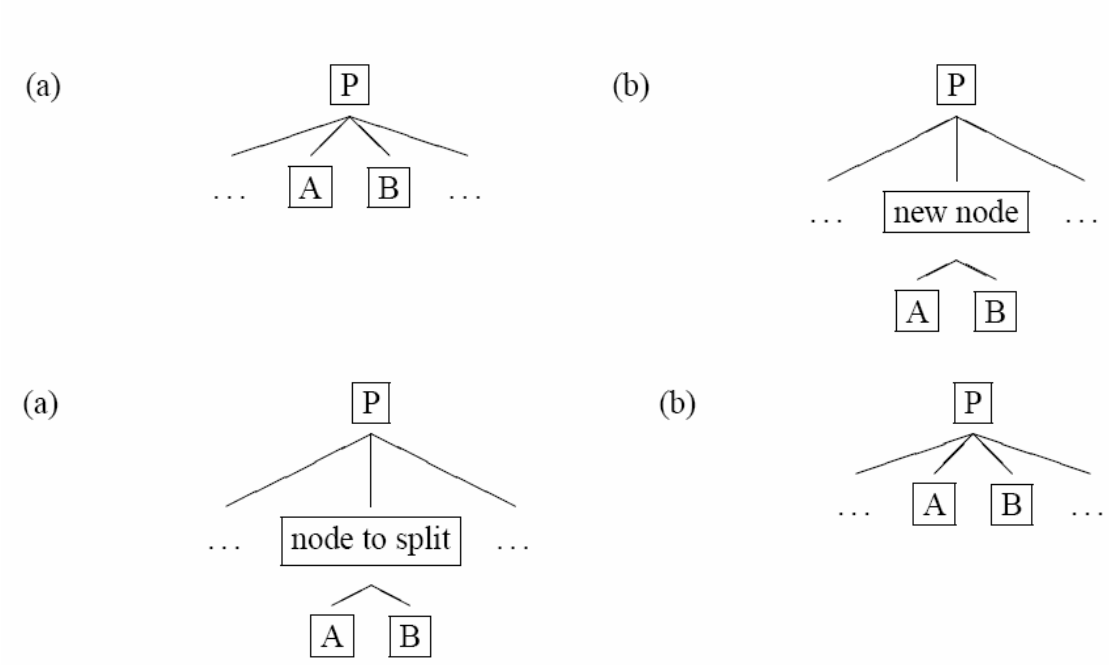
nitelik-değer çiftleri tarafından tanımlanır. COBWEB, sınıflandırma ağacı formu içinde hiyerarşik kümeleme oluşturur.

"Sınıflandırma Ağacı nedir? Karar ağacı ile aynı mıdır?" Sınıflandırma ağacı karar ağacından farklıdır. Sınıflandırmadaki bütün düğümler konsepte başvurur ve konseptin olasılık değerlerini ve koşullu olasılıkları içerir. Koşullu olasılık formülü P(Ai = Vij | C) Ai= Vij nitelik-değer çiftidir, Ck konsept sınıfıdır.(Sayım, her düğümde olasılıkların hesaplanması için biriktirilir ve kaydedilir). Karar ağaçlarında ise düğüm yerine etiket dallar, olasılık değerleri yerine lojik vardır. Verilen seviyedeki sınıf ağacındaki kardeş düğümler, bölünmüş formdur.

COBWEB, "category utility- katagori yararı" diye adlandırılan heuristik bir değerlendirme ölçümü kullanır. CU (category utility) şu şekilde tanımlanır:

$$\frac{\sum_{k=1}^{n} P(C_k)\left[\sum_i \sum_j P(A_i = V_{ij} | C_k)^2 - \sum_i \sum_j P(A_i = V_{ij}{}^2\right]}{n}$$

n, düğüm sayısı; [C1 , C2 ,. . ., Cn] konsept veya kategori ; CU, verilen bölmeden tahmin edilebilen beklenen sayıda nitelik değerinde yükselme .
CU, sınıf içi benzerlik ve farklılık hakkında bilgi verir:



Şekil 5.9 COBWEB'de Düğüm Ayrıştırma İşlemi

- Sınıf içi benzerlik olasılığı P(Ai = Vij | Ck). Büyük değer olması halinde, sınıf üyelerinin nitelik-değer çiftini paylaşma oranı büyük ve daha fazla tahmin edilebilir sınıf üyeleri çifti.
- Sınıf içi farklılık olasılığı P(Ck | Ai = Vij). Büyük değer olması halinde, nitelik-değer çiftini paylaşan sınıf üyeleri içinde az sayıda nesne; daha fazla tahmin edilebilir sınıf çifti.

COBWEB nasıl çalışır ona bakalım. COBWEB artışsal olarak nesneleri sınıflandırma ağacının içine dahil eder.
"Yeni bir nesne verildiğinde , COBWEB nesneyi sınıflandırma ağacı içinde nereye ekler?" COBWEB en iyi düğüm veya host'u bulana kadar ağaçta aşağı doğru ilerler ve ilerlerken değerleri günceller. Karar verme işinde, geçici olarak nesneyi bütün düğümlere yerleştirir,yerleştirilen bölüm için CU hesaplanır. En büyük değerde CU sonucu nesnenin yerini belirler.

Eğer nesne, ağaçta her bir konsepte uzaksa? Verilen nesne için yeni bir düğüm yaratmak daha iyiyse? COBWEB, yeni bir düğüme oluşturulması için CU'yu ölçmektedir. Bu diğer var olan düğümlerle karşılaştırılır. En yüksekteki değerle beraber yeni bir sınıf yaratılır,kaydedilir veya var olan sınıfın içine kaydedilir. COBWEB bölüm içindeki sınıf sayılarını otomatik olarak ayarlar.

İki operatör, nesnelerirn giriş sıralarına göre yüksek hassasiyet gösterir. COBWEB , bu hassasiyeti düşürmek için 2 tane daha operatör kullanır. Bunlar birleştirme (merging) ve ayrıştırma (splitting) dır. Nesne dahil edildiğinde iki en iyi düğüm bir sınıf içine yerleştirilir. Sonra COBWEB, en iyi düğümü ayrıştırır. Bu kararlar CU'ya dayanır. Bu operatörler COBWEB'in direkt(bidirectional) arama yapmasını sağlar.

"COBWEB'in sınırları nelerdir?" COBWEB'in sınır değeri vardır. İlk olarak , ayrık nitelikteki olasılık dağılımları istatiksel olarak birbirinden bağımsız varsayılır. Bu varsayım her zaman doğru değildir, nitelikler arasında genelde karşılıklı bir ilişki vardır. Kümelerin olasılık dağılımlarını temsil edilmesinden dolayı , güncelleme saklamak pahalı bir iştir. Sınıflandırma ağacı dengeli bir ağaç olmadığı için zaman ve alan karmaşıklığı giriş verilere bağlı olarak düşebilir. [20]

5.2.4. Grid temelli metodlar

Grid temelli kümeleme yaklaşımı çok çözümlü grid veri yapısını kullanır.Kümeleme yapılacak alanın sonlu sayıda hücrelere bölünmesiyle oluşur. Ana avantajı genelde birbirinden bağımsız sayıda veri nesnelerinde hızlı işlem zamanıdır.

Grid temelli yaklaşımın bazı genel örnekleri: STING, grid hücrelerindeki istatiksel bilgiyi araştırır; Wavecluster,wavelet dönüşüm metodunu kullanan nesneleri kümeler; CLIQUE, yüksek boyutlu veri alanlarını kümelemek için grid ve yoğunluk temelli yaklaşımı temsi eder [17].

5.2.4. STING (Statistical Information Grid- İstatiksel bilgi Grid)

STING uzayı dikdörtgensel hücrelere bölen bir tekniktir. Bu hücreler hiyerarşik yapıdadır. Üst seviyedeki bütün hücreler bir sonraki alt seviyede parçalanmış hücrelerden oluşur. Her bir grid hücresindeki niteliklerle ilişkili istatiksel bilgi (örneğin mean, maksimum veya minumum değerler) önişlenir veya tutulur.

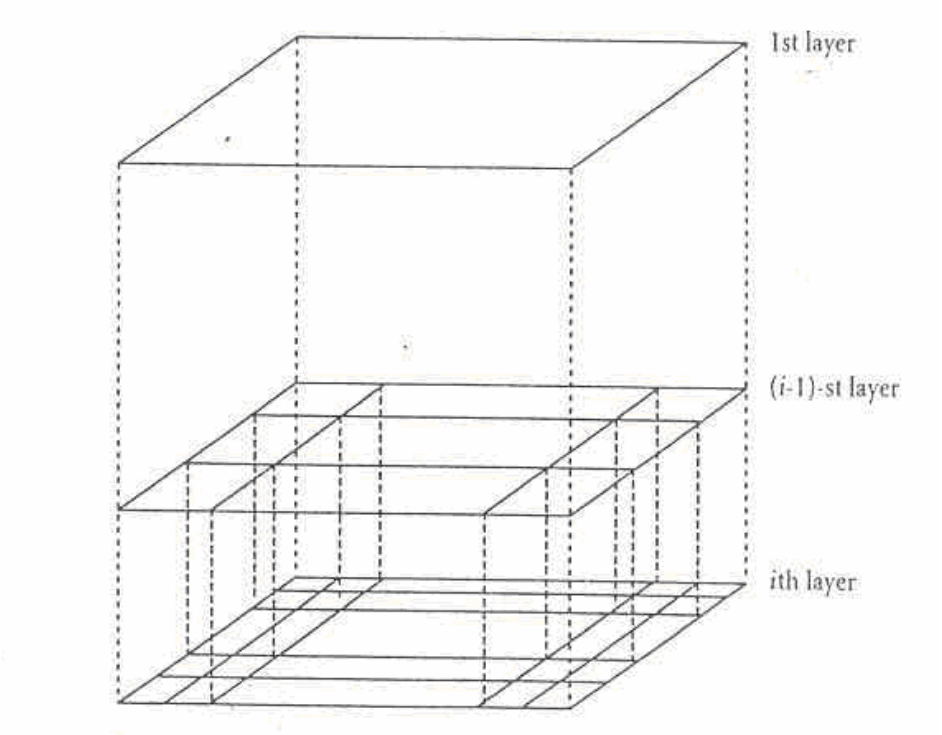
Üst seviyede hücre istatiksel parametreleri, alt seviyedeki hücre istatiksel parametrelerden kolayca hesaplanabilir. Bu parametreler şu şekildedir: bağımsız parametre, sayı (count); nitelik-bağımlı parametreler, m (mean), s (standart sapma), min (minimum), max (maksimum) ve hücrelerin nitel dağılımlarındaki dağılım tipi: normal, tek biçimli, üstel, veya hiçibiri(dağılım bilinmiyor) gibi. Veri , veritabanına kaydedilirken, en alt seviyede hücrelerdeki count, m, s, min ve max parametreleri direkt olarak hesaplanır. Dağılım değeri eğer dağılım tipi biliniyorsa kullanıcı tarafından önceden hesaplanabilir veya X^2 testi gibi hipotez testleri tanımlanabilir. Üst seviyedeki dağılım tipi , aynı alt seviyedeki hücrelerin birleştirilerek eşik filtreleme işleminden geçirilerek bulunabilir. Eğer alt seviye hücrelerdeki dağılım birbiriyle uyuşmuyorsa, eşik testi boşa gider ve üst seviyedeki dağılım tipi hiçbiri olur.

"İstatiksel bilgi, sorgu cevabı için nasıl yarar sağlar?" İlk olarak, sorgu-cevap işleminin başlayacağı hiyerarşik yapıdaki seviye belirlenir. Bu katman genelde küçük

sayıda hücre içerir. Katmandaki bütün hücreler için, verilen sorguya ilgisine göre güven aralığı hesaplarız. İlgisiz hücrele ilerki adımlar için silinir. Bu işlem en alt seviyeye ulaşılana kadar devam eder. Sorgu şartı sağlanırsa hücrelerdeki ilgili bölgeler döndürülür. İlgili veri, sorgunun gereklerini yerine getirene kadar yeniden düzeltilir ve işlenir.

"STING diğer kümeleme metodlarına göre ne gibi avantajlar sağlar?" STING'in bazı avantajları:

- Grid tabanlı hesaplama sorgu bağımsızdır, tüm hücrelerdeki istatiksel bilgi grid hücredeki özet bilgileri içerir, sorguya bağlı değildir;
- Grid yapısı, paralel işleme ve güncelleştirmelere uygundur.
- Metodun verimi asıl avantajıdır: STING, hücrelerdeki istatiksel parametreleri hesaplamak için veritabanına bir kere gider, kümeleri oluşturma zaman karmaşıklığı O(n)'dir, n nesnelerin toplam sayısıdır. Hiyerarşik yapıyı oluşturduktan sonra, sorgu işleme zamanı O(g), g en alt seviyedeki hücre sayısıdır. (genellikle n>g olur).



Şekil 5.10 Sting Kümelemenin Hiyerarşik Yapısı

STING'in kalitesi grid yapısındaki en alt seviye taneciğe bağlıdır. Tanecikler hassas ise işlem maliyeti artar, bununla beraber, en alt seviyedeki grid yapısının kalınlığı kümeleme analiz kalitesini azaltabilir. STING, ana hücrenin çocuk ve komşuları ile ilişkilerini göz önünde bulundurmaz. Kümeleme sınırları düşey veya yataydır, diagonal sınır yoktur. Bu kalite ve doğruluğu düşürür [32].

5.2.5. Yoğunluk temelli metotlar

Birçok kümeleme yöntemi nesnelerin birbirleri arasındaki farklılıklarına göre kümeleme yaparken, bu metot nesnelerin yoğunluğuna göre gruplama yapar. Yoğunluktan kasıt, analiz edilen nesnelerin sayısıdır. Yoğunluk bazlı metotlara örnek olarak DBscan verilebilir.

5.3. Birliktelik Kuralları

Birliktelik kuralları, büyük veri kümeleri arasında birliktelik ilişkileri bulurlar.Toplanan ve depolanan verinin her geçen gün gittikçe büyümesi yüzünden, şirketler veritabanlarındaki birliktelik kurallarını ortaya çıkarmak istemektedirler. Büyük miktardaki mesleki işlem kayıtlarından ilginç birliktelik ilişkilerini keşfetmek, şirketlerin
karar alma işlemlerini daha verimli hale getirmektedir. Birliktelik kurallarının kullanıldığı en tipik örnek market sepeti uygulamasıdır. Bu işlem, müşterilerin yaptıkları alışverişlerdeki ürünler arasındaki birliktelikleri bularak müşterilerin satın alma alışkanlıklarını analiz eder. Bu tip birlikteliklerin keşfedilmesi, müşterilerin hangi ürünleri bir arada aldıkları bilgisini ortaya çıkarır ve market yöneticileri de bu bilgi ışığında daha etki satış stratejileri geliştirebilirler.

Örneğin bir müşteri süt satın alıyorsa, aynı alışverişte sütün yanında ekmek alma olasılığı nedir? Bu tip bir bilgi ışığında rafları düzenleyen market yöneticileri ürünlerindeki satış oranını arttırabilirler. Örneğin bir marketin müşterilerinin süt ile birlikte ekmek satın alan oranı yüksekse, market yöneticileri süt ile ekmek raflarını yan yana koyarak ekmek satışlarını arttırabilirler.

Örneğin bir A ürününü satın alan müşteriler aynı zamanda B ürününü da satın alıyorlarsa, bu durum (2.1)'deki Birliktelik Kuralı ile gösterilir;

A => B [destek = %2, güven = %60] (2.1)

Buradaki destek ve güven ifadeleri, kuralın ilginçlik ölçüleridir. Sırasıyla, kesfedilen kuralın kullanışlılığını ve doğruluğunu gösterirler. (2. l)'deki Birliktelik Kuralı için %2 oranındaki bir destek değeri, analiz edilen tüm alışverişlerden %2'sinde A ile B ürünlerinin birlikte satıldığını belirtir. %60 oranındaki güven değeri ise A ürününü satın alan müşterilerinin %60'ının aynı alışverişte B ürününü de satın aldığını ortaya koyar. Kullanıcı tarafından minimum destek eşik değeri ve minimum güven eşik değeri belirlenir ve bu değerleri asan birliktelik kuralları dikkate alınır.

Büyük veri tabanlarında birliktelik kuralları bulunurken, şu iki işlem basamağı takip edilir;

1- Sık tekrarlanan öğeler bulunur: Bu öğelerin her biri en az, önceden belirlenen minimum destek sayısı kadar sık tekrarlanırlar.

2- Sık tekrarlanan Öğelerden güçlü birliktelik kuralları oluşturulur: Bu kurallar minimum destek ve minimum güven değerlerini karşılamalıdır.

Sık tekrarlanan öğeleri bulmak için kullanılan en temel yöntem Apriori Algoritmasıdır. Aşağıda Apriori algoritması bir örnekle anlatılmaktadır.

Tablo 5.2'de bir marketten yapılan alışverişlerin bilgilerini içeren *D* veritabanı görülmektedir. Bu veritabanında yapılan alışverişlerin numaraları ANO sütununda görülmektedir. Her alışverişte satın alınan ürünler de Ürün No sütununda görülmektedir. Apriori algoritmasında takip edilen basamaklar Şekil 5.11'de gösterilmektedir.

1- Algoritmanın ilk adımında, her ürün tek basına bulunduğu *C1* kümesinin elemanıdır. Algoritma, her ürünün sayısını bulmak için tüm alışverişleri tarar ve elde edilen sonuçlar Şekil 5.11'de Destek Sayısı sütununda görülmektedir. Tablo 5.2'de görülebileceği gibi *D*'de I1 ürününden 6 adet, 12 ürününden 7 adet, 13 ürününden 6 adet, 14 ürününden 2 adet ve 15 ürününden de 2 adet satıldığı görülmektedir.

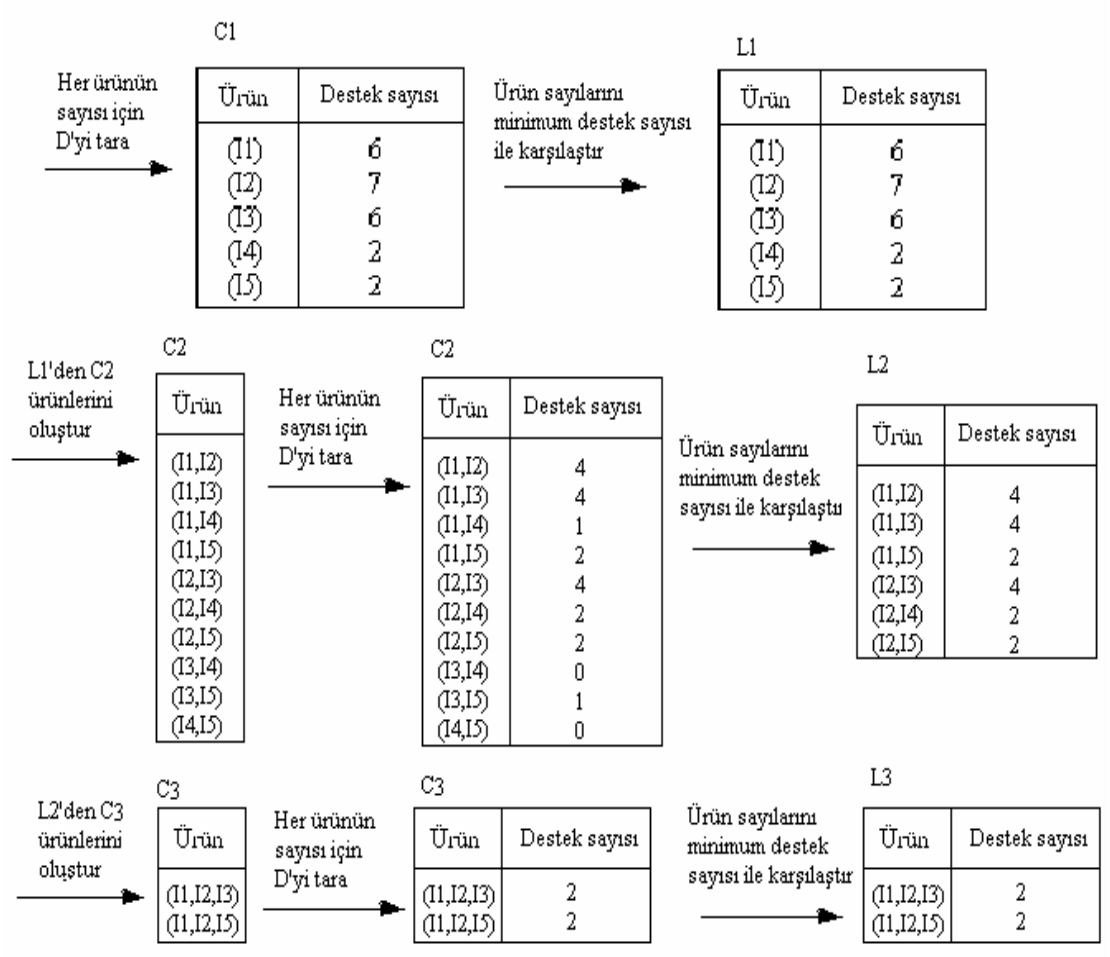
Tablo 5.2. Marketten Yapılan Alışveriş Bilgilerini içeren *D* Veritabanı

ANO	Ürün NO
A100	I1,I2,I5
A200	I2,I4
A300	I2,I3
A400	I1,I2,I4
A500	I1,I3
A600	I2,I3
A700	I1,I3
A800	I1,I2,I3,I5
A900	I1,I2,I3

2- Minimum alışveriş destek sayısının 2 olduğu varsayılırsa, tek baslarına sık tekrar lanan ürünler *L1* kümesinde görülmektedir. *C1* kümesindeki tüm ürünlerin destek sayısı, minimum destek eşik değeri olan 2'den fazla olduğu için *C1* tüm ürünler sık tekrarlanan ürün olarak değerlendirilir ve *L1* kümesine aktarılır.

3- Hangi ürünlerin ikili olarak sık tekrarlandığını belirlemek için *L1* kümesindeki ürünlerin ikili kombinasyonları bulunarak *C2* kümesi oluşturulur.

4- *C2* kümesindeki ürünlerin destek sayılarını bulmak amacıyla *D* taranır ve bulunan değerler destek sayısı sütununda belirtilir.



Şekil 5.11 Apriori Algoritmasının Gösterimi

5- *C2* kümesindeki ürünlerden minimum destek eşik değerini asan ürünler *L2* kümesine aktarılır.

6- Hangi ürünlerin üçlü olarak sık tekrarlandığını belirlemek için *L2* kümesindeki ürünlerin üçlü kombinasyonları bulunarak *C3* kümesi oluşturulur. Bu durumda *C3* = {{I1,I2,I3}, {I1,I2,I5}, {I1,I3,I5}, {I2,I3,I4}, {I2,I3,I5}} olması beklenir. Ancak Apriori algoritmasına göre, sık tekrarlanan öğelerin alt kümeleri de sık tekrarlanan öğe olması gerekmektedir. Buna göre yukarıdaki *C3* kümesindeki elemanlar sık tekrarlanan olmadığı için, yeni *C3* kümesi *C3* = {{I1,I2,I3}}, {I1,I2,I5}} olur.

7- *C3* kümesindeki ürünlerin destek sayılarını bulmak amacıyla *D* taranır ve bulunan değerler destek sayısı sütununda belirtilir.

8- *C3* kümesindeki ürünlerden minimum destek eşik değerini asan ürünler *L3* kümesine aktarılır.

9- Hangi ürünlerin dörtlü olarak sık tekrarlandığını belirlemek için *L3* kümesindeki ürünlerin dörtlü tek kombinasyonu {II, I2, I3, I5} olarak belirlenir. Ancak bu kümenin alt kümelerinin tamamı sık tekrarlanan öğe olmadığı için *C4* kümesi boş küme olur ve Apriori tüm sık tekrarlanan öğeleri bularak sonlanmıs olur. Sık tekrarlanan öğeleri bulduktan sonra , sıra birliktelik kurallarını oluşturmaya gelir.

Örneğin sık tekrarlanan bir öğe olan I1 için, boşolmayan tüm alt kümeler şunlardır [11]: {I1, I2}, {I2, I5}, {I2, I5}, {I1}, {I2},{I5}. Bu durumda Tablo 5.2'deki veritabanına bakarak şu birliktelik kuralları çıkartılabilir:

1- I1 Λ I2 → I5 güven = 2 / 4 = %50

2- I1 Λ I5 → I2 güven = 2 / 2 = %100

3- I2 Λ I5 → I1 güven = 2 / 2 = %100

4- I1 → I2 Λ I5 güven = 2 / 6 = %33

5- I2 → I1 Λ I5 güven = 2 / 7 = %29

6- I5 → I1 Λ I2 güven = 2 / 2 = %100

Eğer minimum güven eşik değeri %70 olarak belirlenmişse, ikinci, üçüncü ve altıncı kurallar dikkate alınır. Çünkü diğer kurallar eşik değerini aşamamış olurlar [18].

BÖLÜM 6. WEKA VERİ MADENCİLİĞİ YAZILIMI VE BİR UYGULAMA

6.1. Genel Bilgiler

WEKA(Waikato Environment for Knowledge Analysis), Yeni Zelanda Waikato Üniversitesi' nde geliştirilen bir veri madenciliği ve makine öğrenmesi (machine learning) yazılımıdır. WEKA yazılımı nesneye yönelik programlama dillerinden olan Java ile geliştirilmiştir. Java birçok değişik öğrenme algoritmaları için düzenli bir platform sağlamaktadır. WEKA' nın en güçlü özelliği birçok sınıflandırma tekniklerini içermesidir. Diğer bir özelliği de uygulamaların komut girilerek gerçekleştirilmesine imkan tanımasıdır [40].

WEKA dört temel modülden oluşmaktadır. Bu modüllere GUI(General User Interface) genel kullanıcı arayüzünden ulaşılabilir. Bunlar:

Simple CLI: Basit komut satırı arayüzü olarak adlandırılan bu modül, WEKA komutlarının direkt olarak çalıştırılmasını sağlar.

Explorer: WEKA ile veri keşfi yapılmasına imkan sağlayan bir platformdur. Veri madenciliği tekniklerini kullanmayı ve görselleştirme yapmayı sağlayan modüldür. Bu modül veri madenciliği açısından detaylı olarak incelenecektir.

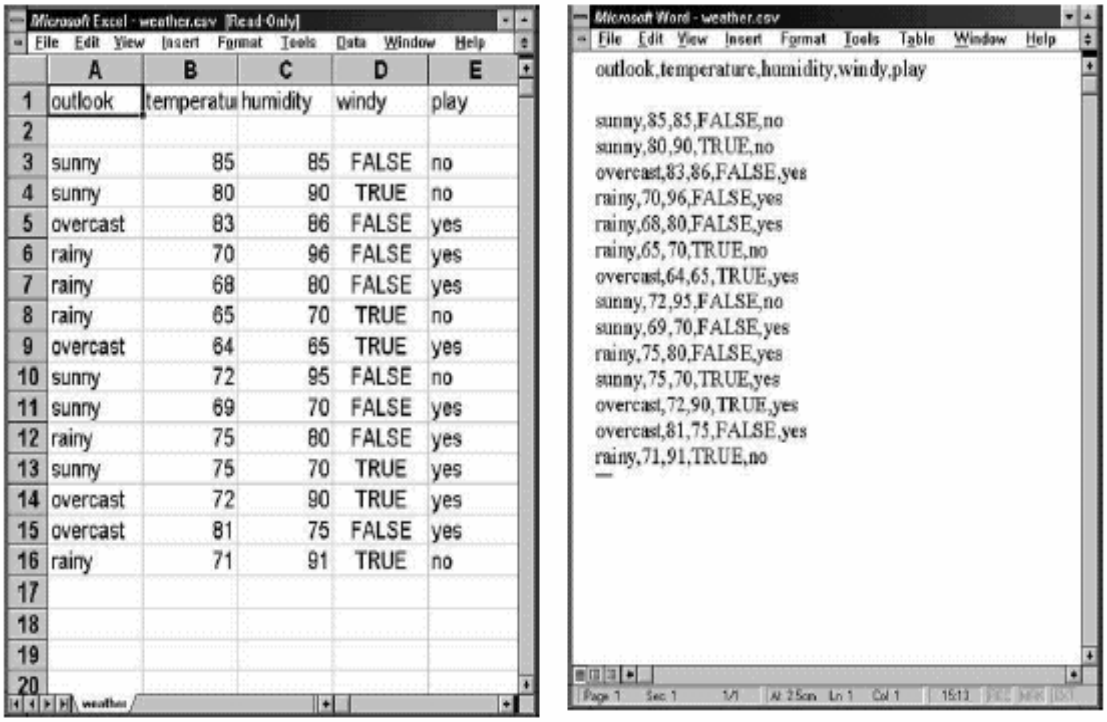
Experimenter: Öğrenme setlerinin denemelerinin ve aralarındaki istatistiksel testlerin yapılmasını sağlayan modüldür(Deneme Modülü).

Knowledge Flow: Explorer modülünün grafik olarak temsil edilmesidir.Bilgi akışının modellenmesini sağlayan bir modüldür.

Şekil 6.1 WEKA Genel Kullanıcı Arayüzü

Explorer ilk çalıştırıldığında preprocess menüsü dışındakiler aktif değildir. Bunun sebebi öncelikle bir veri seti seçilmesini sağlamaktır. Veri seti bir dosyadan, veritabanından yada URL adresi girilerek seçilebilir. WEKA, veri setleri için *.arff dosya formatını kabul etmektedir. Veri setini bu formata dönüştürmek için şu yol izlenir.

1. Excel gibi bir tablo düzenleyicide veriler girilir. (a)
2. Düzenlenen tablo farklı kaydet menüsüyle *.csv (virgülle ayrılmış) formatında kaydedilir. (b)



Microsoft Excel - weather.csv [Read-Only]

File Edit View Insert Format Tools Data Window Help

	A	B	C	D	E
1	outlook	temperatu	humidity	windy	play
2					
3	sunny	85	85	FALSE	no
4	sunny	80	90	TRUE	no
5	overcast	83	86	FALSE	yes
6	rainy	70	96	FALSE	yes
7	rainy	68	80	FALSE	yes
8	rainy	65	70	TRUE	no
9	overcast	64	65	TRUE	yes
10	sunny	72	95	FALSE	no
11	sunny	69	70	FALSE	yes
12	rainy	75	80	FALSE	yes
13	sunny	75	70	TRUE	yes
14	overcast	72	90	TRUE	yes
15	overcast	81	75	FALSE	yes
16	rainy	71	91	TRUE	no
17					
18					
19					
20					

Microsoft Word - weather.csv

File Edit View Insert Format Tools Table Window Help

outlook,temperature,humidity,windy,play

sunny,85,85,FALSE,no
sunny,80,90,TRUE,no
overcast,83,86,FALSE,yes
rainy,70,96,FALSE,yes
rainy,68,80,FALSE,yes
rainy,65,70,TRUE,no
overcast,64,65,TRUE,yes
sunny,72,95,FALSE,no
sunny,69,70,FALSE,yes
rainy,75,80,FALSE,yes
sunny,75,70,TRUE,yes
overcast,72,90,TRUE,yes
overcast,81,75,FALSE,yes
rainy,71,91,TRUE,no

(a) **(b)**

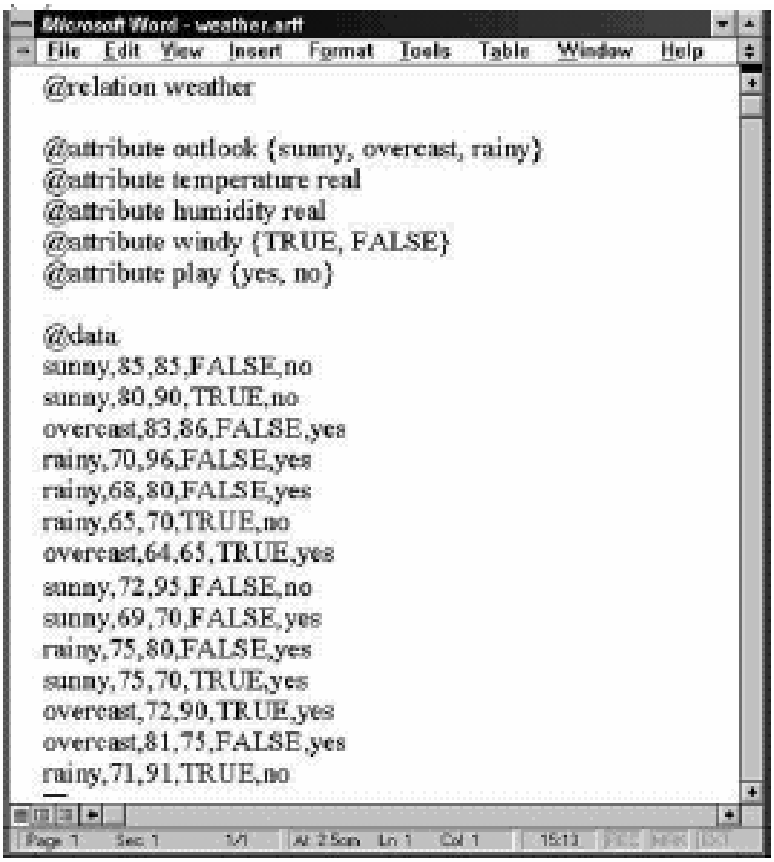
Şekil 6.2 Örnek Bir Veri Seti Düzeni

3. *.csv formatında kaydedilen bu belge herhangi bir metin düzenleyicide açılarak çeşitli düzenlemeler yapılır. Girilen veri setindeki sütun isimleri bir nitelik (attribute) olarak temsil edilir. Her nitelik için bir tanımlama yapılır. Nitelik değerleri sayısal (real yada integer), nominal, string yada tarih(date) olabilir(c).

@relation weather

@attribute outlook {sunny, overcast, rainy}
@attribute temperature real
@attribute humidity real
@attribute windy {TRUE, FALSE}
@attribute play {yes, no}

Nominal değerler aralarına virgül konularak {} aralığında yazılır.



Şekil 6.3 Arff Veri Formatı

Bu şekilde düzenlendikten sonra metin düzenleyicide *.txt olarak kaydedilip kapatılarak uzantısı *.arff olarak değiştirilir. Eğer WEKA yazılımı bilgisayarda kurulu ise bu işlemden sonra belge WEKA sembolünü alacaktır.

WEKA içerisinde var olan weather.arff örnek veri setini açtığımızda diğer tüm alt modüller (Clssify, Cluster, Associate, Select Attribute, Visualize) aktif hale gelir [40].

6.2. Banka Örneği

Aşağıda bir bankanın müşterileri ile ilgili bir veri seti görülmektedir. Banka müşterileri kümelere ayırarak hangi tip müşterinin kredi ödemesini zamanında yaptığını, hangisinin ödemelerde geciktiğini ve hangisinin de hiç ödemediğini görmek istemektedir. Bizde bu işlemi yaparken kümeleme algoritmalarını karşılaştırma fırsatı bulacağız.

@relation banka

@attribute yaş numeric
@attribute cinsiyet {BAYAN,ERKEK}
@attribute bölge {ŞEHİR_MERKEZİ,İLÇE,KASABA,KÖY}
@attribute kazanç numeric
@attribute evli {HAYIR,EVET}
@attribute çocuk {0,1,2,3}
@attribute araba {HAYIR,EVET}
@attribute mevduat_hesabı {HAYIR,EVET}
@attribute cari_hesap {HAYIR,EVET}
@attribute ev {HAYIR,EVET}
@attribute fazla_kredi {EVET,HAYIR}
@attribute ödemiş {EVET,GECİKMİŞ,HAYIR}

@data

48,BAYAN,ŞEHİR_MERKEZİ,17546,HAYIR,1,HAYIR,HAYIR,HAYIR,HAYIR,EVET,GECİKMİ
40,ERKEK,İLÇE,30085.1,EVET,3,EVET,HAYIR,EVET,EVET,HAYIR,EVET
51,BAYAN,ŞEHİR_MERKEZİ,16575.4,EVET,0,EVET,EVET,EVET,HAYIR,HAYIR,GECİKMİŞ
23,BAYAN,İLÇE,20375.4,EVET,3,HAYIR,HAYIR,EVET,HAYIR,HAYIR,GECİKMİŞ
57,BAYAN,KASABA,50576.3,EVET,0,HAYIR,EVET,HAYIR,HAYIR,HAYIR,EVET
57,BAYAN,İLÇE,37869.6,EVET,2,HAYIR,EVET,EVET,HAYIR,EVET,EVET
22,ERKEK,KASABA,8877.07,HAYIR,0,HAYIR,HAYIR,EVET,HAYIR,EVET,HAYIR
58,ERKEK,İLÇE,24946.6,EVET,0,EVET,EVET,EVET,HAYIR,HAYIR,GECİKMİŞ
37,BAYAN,KÖY,25304.3,EVET,2,EVET,HAYIR,HAYIR,HAYIR,HAYIR,GECİKMİŞ
54,ERKEK,İLÇE,24212.1,EVET,2,EVET,EVET,EVET,HAYIR,HAYIR,GECİKMİŞ
66,BAYAN,İLÇE,59803.9,EVET,0,HAYIR,EVET,EVET,HAYIR,HAYIR,EVET
52,BAYAN,ŞEHİR_MERKEZİ,26658.8,HAYIR,0,EVET,EVET,EVET,EVET,HAYIR,GECİKMİŞ
44,BAYAN,İLÇE,15735.8,EVET,1,HAYIR,EVET,EVET,EVET,EVET,GECİKMİŞ
66,BAYAN,İLÇE,55204.7,EVET,1,EVET,EVET,EVET,EVET,EVET,EVET
36,ERKEK,KASABA,19474.6,EVET,0,HAYIR,EVET,EVET,EVET,HAYIR,GECİKMİŞ
38,BAYAN,ŞEHİR_MERKEZİ,22342.1,EVET,0,EVET,EVET,EVET,EVET,HAYIR,GECİKMİŞ
37,BAYAN,İLÇE,17729.8,EVET,2,HAYIR,HAYIR,HAYIR,EVET,HAYIR,GECİKMİŞ
46,BAYAN,KÖY,41016,EVET,0,HAYIR,EVET,HAYIR,EVET,HAYIR,EVET
62,BAYAN,ŞEHİR_MERKEZİ,26909.2,EVET,0,HAYIR,EVET,HAYIR,HAYIR,EVET,GECİKMİŞ
31,ERKEK,İLÇE,22522.8,EVET,0,EVET,EVET,EVET,HAYIR,HAYIR,GECİKMİŞ
61,ERKEK,ŞEHİR_MERKEZİ,57880.7,EVET,2,HAYIR,EVET,HAYIR,HAYIR,EVET,EVET
50,ERKEK,İLÇE,16497.3,EVET,2,HAYIR,EVET,EVET,HAYIR,HAYIR,GECİKMİŞ

54,ERKEK,ŞEHİR_MERKEZİ,38446.6,EVET,0,HAYIR,EVET,EVET,HAYIR,HAYIR,EVET

27,BAYAN,İLÇE,15538.8,HAYIR,0,EVET,EVET,EVET,EVET,HAYIR,GECİKMİŞ

22,ERKEK,ŞEHİR_MERKEZİ,12640.3,HAYIR,2,EVET,EVET,EVET,HAYIR,

.

.

.

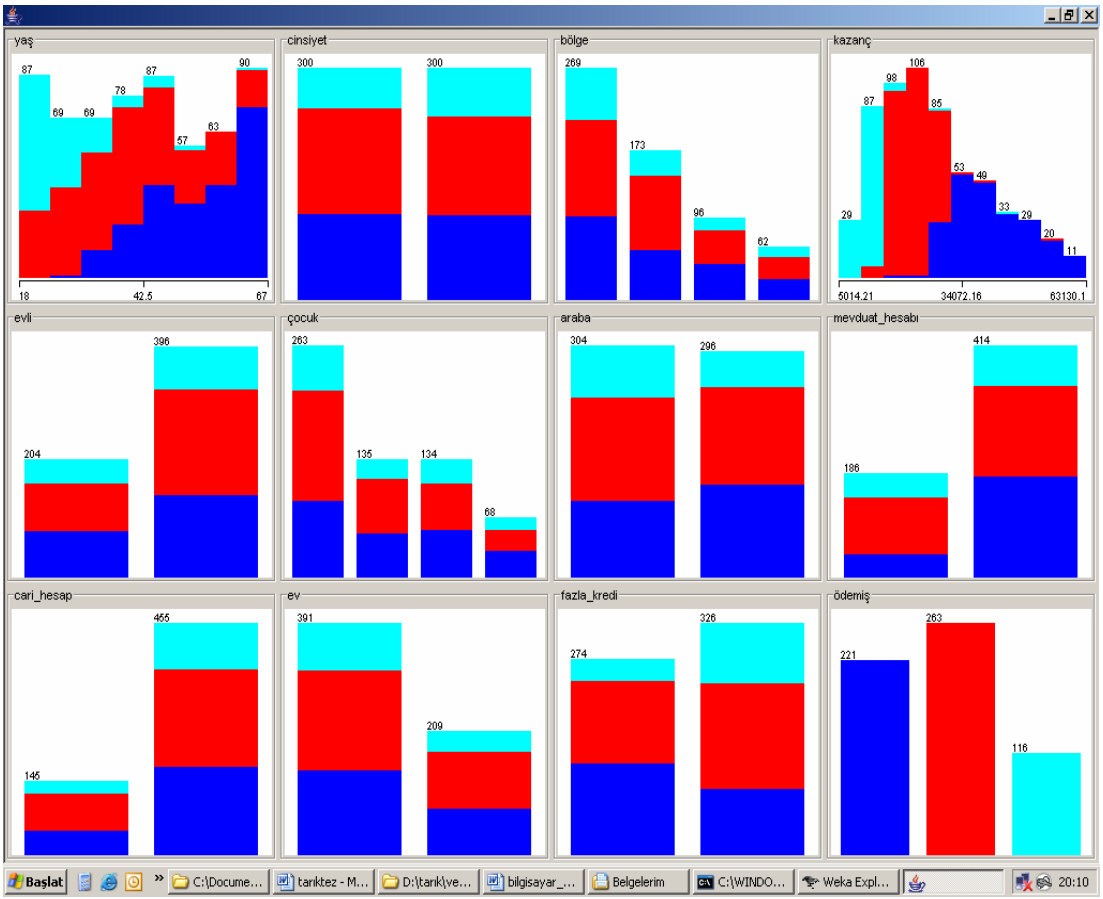
.

.

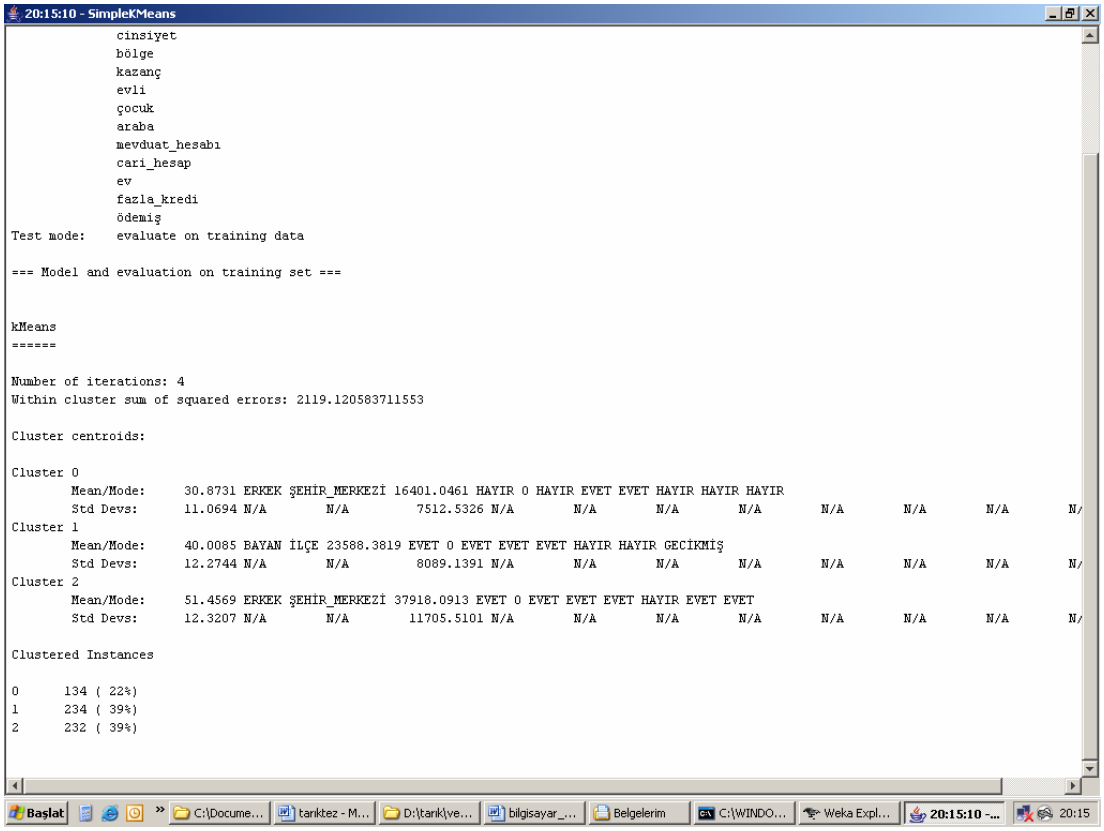
.

.

Veri setinde 12 değişken ve 600 kayıt bulunmaktadır.

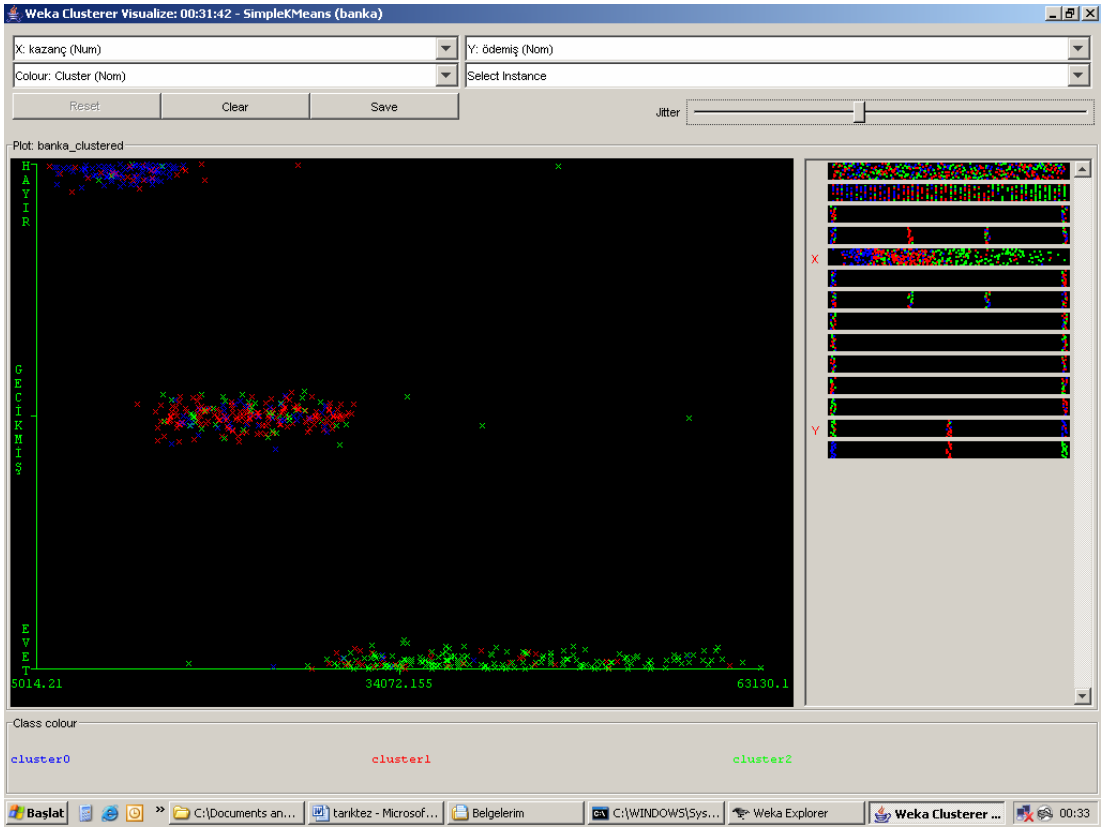


Şekil 6.4 Weka'da veri setinin grafiksel gösterimi



Şekil 6.5 K-Means Weka Çıktısı

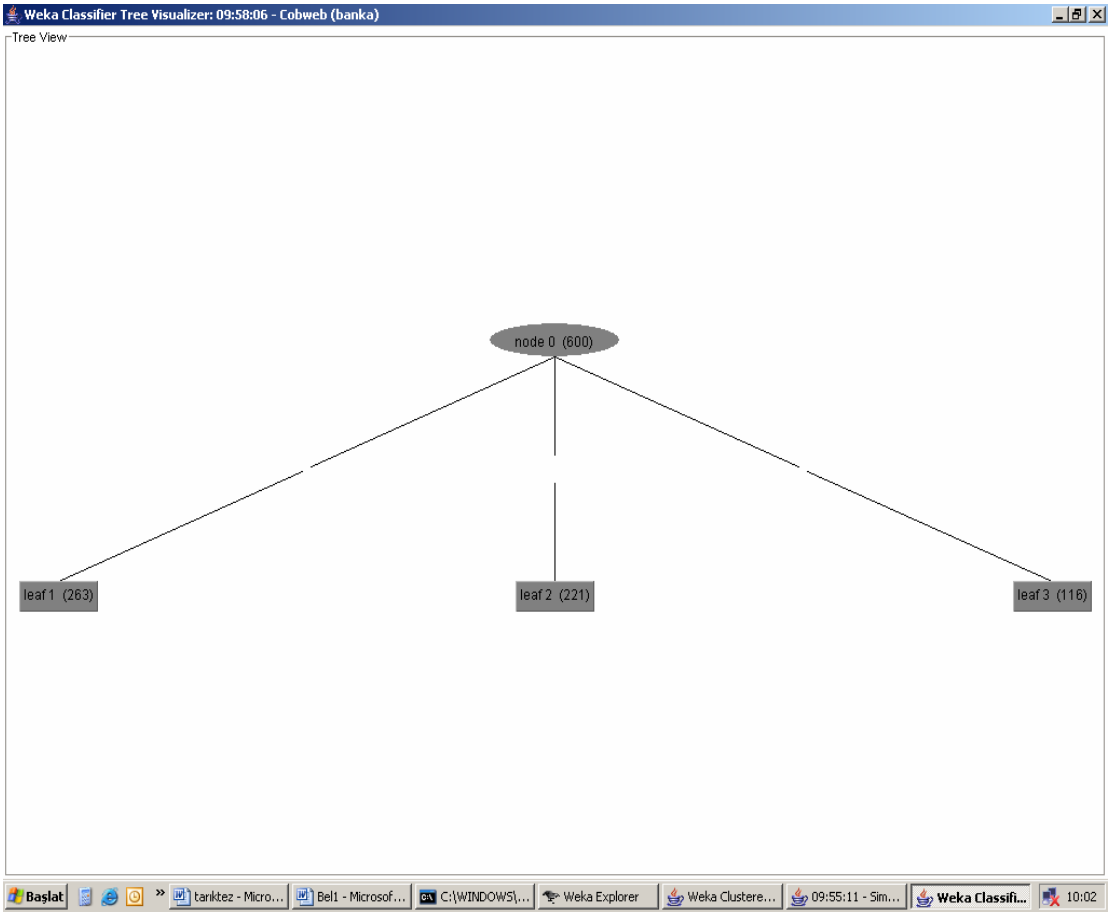
K-means algoritması; müşterileri zamanında ödeyenler, gecikenler ve ödemeyenler olarak 3 ana kümeye ayırdı. Şekil 6.5'te de görülebileceği gibi kazanç ve ödeme durumu arasında çok yakın bir ilişki vardır. Yıllık kazancı yaklaşık olarak 15000'den düşük olanların ödeme yapamadığını, 15000 ile 30000 arasında olanların geciktiğini ve 30000'den fazla olanların ise ödemelerini zamanında yapmış olduğunu görmekteyiz. Kazanç dışındaki değişkenler ile ödeme durumu arasında ise pek bir ilişki kurulamamıştır. Bu yüzden diğer algoritmalarda kazanç ve ödeme değişkeni arasında ilişki kurulacaktır.



Şekil 6.6 K-Means te Kümelerin Grafiksel Gösterimi

Grafikteki mavi noktalar birinci kümeye, kırmızı noktalar ikinci kümeye, yeşil noktalar ise üçüncü kümeye aittir. Grafiğin x ekseni yıllık kazanç miktarını, y ekseni ise ödeme durumunu göstermektedir. Birinci küme ödeyemeyenleri, ikinci küme gecikenleri, üçüncü küme de vaktinde ödemesini yapanları göstermektedir.

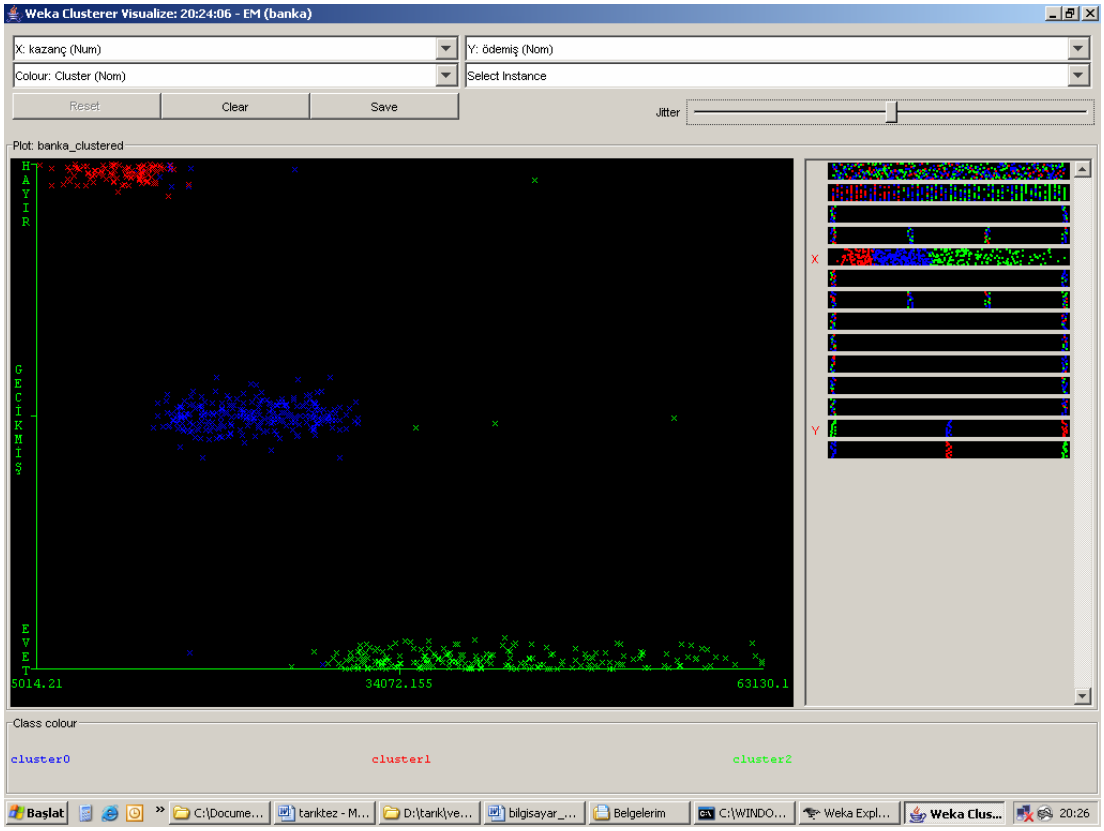
Grafik dikkatli bir şekilde incelendiğinde, bankamızın verilerine yönelik olarak uyguladığımız k-means algoritmasının müşterileri ödeme durumlarına göre çok kesin hatlarla birbirinden ayırmadığını görüyoruz. Çünkü grafikteki renk grupları birbirlerinden tamamen bağımsız değil. Bunun en büyük sebebi bu algoritmanın sert (hard) bir algoritma oluşudur.



Şekil 6.7 COBWEB Algoritması Düğüm Gösterimi

COBWEB algoritmasını banka verilerine uyguladığımızda karşımıza bu şekilde bir ağaç yapısı çıkmaktadır. Bu algoritma karar ağacı mantığı ile çalıştığından verileri bu şekilde düğümlerle ifade etmektedir.

Sadece kazanç ve ödeme durumu değişkenlerini göz önüne aldığımız için böyle basit bir ağaç yapısı meydana gelmiştir. Diğer değişkenler algoritmaya katıldığında yaklaşık 800 tane küme elde edilmektedir. Böyle bir sonuç ise elbette ki bankamız için yararlı olmamaktadır.



Şekil 6.8 Em Algoritması Grafik Gösterimi

EM algoritması banka verilerine uygulandığında yukarıdaki grafik oluşmuş ve oldukça etkili bir sonuç elde edilmiştir. Çünkü kümeler tam olarak birbirinden ayrılmış ve hangi müşterinin hangi kümede yer aldığı açık bir şekilde gösterilmiştir.

Grafikte kırmızı noktalar ödemelerini yapmayan müşterileri, mavi noktalar gecikenleri, yeşil noktalar ise zamanında ödeme yapanları göstermektedir.

Böylece bankamız kredi talebinde bulunan müşterilere karşı nasıl hareket edeceğini belirlemiş olacaktır. Kazancı 15000 ytl dendüşük olanlara kredi vermeyecek, kazancı 15000 ile 30000 ytl arasında olanlara ancak özel şartlar uygulayarak kredi verecek ve kazancı 30000 ytl nin üzerinde olan müşterilerine ise rahatça kredi verebilecektir.

BÖLÜM 7. SONUÇLAR VE ÖNERİLER

1- Bu çalışmada veri madenciliği modelleri Sınıflama ve Regresyon, Kümeleme ve Birliktelik Kuralları başlıkları altında incelenmiş ve kümeleme algoritmaları üzerinde detaylı bir şekilde durulmuştur. Kümeleme algoritmasının veri madenciliğinde nasıl kullanıldığı bir uygulama ile anlatılmıştır. Uygulamada bir bankanın müşteri kayıtları ele alınmış ve bu kayıtlar kümeleme algoritmaları ile veri madenciliğine tabi tutularak müşterilerin kredilerini ödeme durumlarına göre gruplanması sağlanmıştır. Böylece banka bir dahaki sefere kredi talebinde bulunan müşterilere buradaki sonuçlara göre muamele yapacaktır. Bizim örneğimizde müşterilerin yıllık gelirlerinin kredilerini zamanında ödemeleri noktasından kesinlikle belirleyici bir özellik olduğu ortaya çıkmıştır.

2- Bir veri madenciliği uygulaması gerçekleştirileceği zaman eldeki verinin ve problemin çok iyi bir şekilde analiz edilmesi gerekir. Bu işlem yapıldıktan sonra veri madenciliği tekniklerinden amaca en uygun olanı seçilmelidir.

3- K-means algoritması sert bir algoritma olduğundan sayasal verilerde çok hassas davranmaktadır. Bu yüzden kümeler kesin hatlarla birbirinden ayrılmamıştır. Karar verme sürecinde mühim bir eksikliktir.

4- EM algoritması çok başarılı bir şekilde kümeler oluşturmuş ve küme elemanlarını kesin olarak birbirinden ayırmıştır. Nominal ve nümerik değişkenlerin birlikte bulunduğu veri setlerinde başarılı olduğu görülmüştür.

5- COBWEB algoritması nominal ve nümerik değişkenlerin beraber olduğu veri setlerinde anlamsız sonuçlar çıkarabilmektedir. Ayrıca değişken sayısının çok fazla olması küme sayısını istenmeyecek şekilde arttırmaktadır.

KAYNAKLAR

[1] ACKNOSOFT ; Introduction To Data Mining And Case Based Reasoning, Http://Www.Acknosoft.Com/Technology.Html (26.10.200)

[2] AKPINAR, Haldun; Veri Tabanlarında Bilgi Keşfi Ve Veri Madenciliği, İ.Ü. İşletme Fakültesi Dergisi, C:29, S: 1/Nisan 2000.

[3] ALATAŞ, Bilal – Akın, Erhan; Veri Madenciliğinde Yeni Yaklaşımlar, Ya/Em-2004- Yöneylem Araştırması/Endüstri Mühendisliği Xxıv Ulusal Kongresi, 15-18 Haziran 2004, Gaziantep-Adana.

[4] ALPAYDIN, Ethem; Zeki Veri Madenciliği: Ham Veriden Altın Bilgiye Ulaşma Yöntemleri, Bilişim 2000 Veri Madenciliği Eğitim Semineri, Http://Www.Cmpe.Boun.Edu.Tr/~Ethem/Files/Papers/Veri-Maden-2k-Notlar.Doc(24.12.2004).

[5] AZMY, Ashraf. (18/05/1998). Superquery;Data Mining For Everyone

[6] BAYKASOĞLU, A – Öztaş, A – Erdoğan, E.T. ; Veri Madenciliği Tekniklerinin İhale Tenzilat Miktarı Karar Süreçlerinde Kullanımı, Ya/Em-2004-Yöneylem Araştırması /Endüstri Mühendisliği Xxıv Ulusal Kongresi, 15-18 Haziran 2004, Gaziantep-Adana.

[7] BOAR, B., 2000, December 25, Understanding Data Warehousing Strategically, Www.Carleton.Com.Au/Understanding%20data%20warehousing%20strategically.Htm

[8] CAMBAZOĞLU, T., 2000, Veri Ambarı [Data Warehousing) Temelleri, Www.Bilisimrehber.Com.Tr/Arastirma/Tr_Arastirma_Veriambari_Temelleri.Phtml

[9] CLEMENTINE, Tutorial/Practical, Data Mining-An Introduction Tutorial/Practical, Qub, 2003, Http://Www.Pcc.Qub.Ac.Uk /Tec /Courses/Datamining/Ohp/Dm-Ohp-Final_2.Html

[10] Data Mining Softwares And Datas, 2003, Http://Www.Yake.Ecn.Purdue.Edu/_~Brodley/Software/Lmdt.Html

[11] DILLY, Ruth; Data Mining: An Introduction, Http://Www.Pcc.Qub.Ac.Uk/Tec/Courses/Datamining/Stu_Notes/Dm_Book_1.Html(24.12.2004).

[12] EKER, Hakan, (A); Veri Madenciliği Veya Bilgi Keşfi, Http://Www.Bilgiyonetimi.Org/Cm/Pages/Mkl Gos.Php?Nt=538(25.01.2005).

[13] EKER, Hakan, (B); İşletmelerde Tutulan Müşteri Verilerinin Anlamlı Hale Getirilmesi Ve Etkin Kullanılması, Http://Www.Danismend.Com/Konular/Bilgiveteknoyon/Bilgi_Veri_Madencil igi.Htm (10.04.2005).

[14] ELDER Iv, J. F., Abbott, D. W., 1998, August 28, A Comparison Of Leading Data Mining Tools", Fourth International Conference On Knowledge Discovery & Data Mining Friday, New York E. Knorr And R. Ng. Algorithms For Mining Distance-Based Outliers In Large Datasets. Vldb'98.

[15] E. Schikuta. Grid Clustering: An Efficient Hierarchical Clustering Method For Very Large Data Sets. Proc. 1996 Int. Conf. On Pattern Recognition, 101-105.

[16] ETHEM Alpaydın (2000); Zeki Veri Madenciliği; Ham Veriden Altın Bilgiye Ulaşmaın Yöntemleri,

[17] G. J. Mclachlan And K.E. Bkasford. Mixture Models: Inference And Applications To Clustering. John Wiley And Sons, 1988

[18] G. Sheikholeslami, S. Chatterjee, And A. Zhang. Wavecluster: A Multi-Resolution Clustering Approach For Very Large Spatial Databases. Vldb'98

[19] GOEBEL, M. – Gruenwald, L.; A Survey Of Data Mining And Knowledge Discovery Software Tools, Sıgkdd Explorations, Usa, 1999.

[20] GÜRASAKAL, Nemci, Vd. ; Değişen Veri Kavramı Ve Yeni Alanlar, İstatistik Araştırma Sempozyumu, Bildiriler Kitabı, 27-29 Kasım, 2000.

[21] HOLTE, R., 2003, Data Mining Tutorial, Simon Fraser University, Machine Learning, Http://Www.Csi.Uottawa.Ca/~Holte/Learning/ Other-Sites.Html.

[22] KARAKAŞ, Melikşah; Veri Madenciliği Üzerine, Http://Www.Bilgiyonetimi.Org/Cm/Pages/Mkl-Gos.Php?Nt=132(25.01.2005).

[23] L. Kaufman And P. J. Rousseeuw. Finding Groups In Data: An Introduction To Cluster Analysis. John Wiley & Sons, 1990.

[24] MURTY, L., Kasif, M. L. Ve Salzberg, N., 20 August 2000, Http://Www.Cs.Jhu.Edu/ ~Salzberg/ Announce-Oc1.Html.

[25] OĞUZLAR, Ayşe; Veri Ön İşleme, Erciyes Üniversitesi İibf Dergisi, Sayı:21, Temmuz-Aralık 2003.

[26] ÖZMEN, Şule; İş Hayatı Veri Madenciliği İle İstatistik Uygulamalarını Yeniden Keşfediyor, Http://Www.İdari.Cu.Edu.Tr/Sempozyum/Bil38htm (24.01.2005).

[27] P. Michaud. Clustering Techniques. Future Generation Computer Systems, 13, 1997

[28] PIRAMUTHU, S.; Evaluating Feature Selection Methods For Learning İn Data Mining Applications, European Journal Of Operational Research, Article In Pres, 2003.

[29] R. Ng And J. Han. Efficient And Effective Clustering Method For Spatial Data Mining. Vldb'94.

[30] SAS Institute Inc. Using Data Mining Techniques For Fraud Detection. 1999

[31] SAS Institute Inc. Finding The Solution To Data Mining. 1998

[32] SAS Institute Inc. Data Mining And The Case For Sampling. 1998

[33] SEİDMAN, C., 2000, Data Mining With Microsoft Sql Server 2000, Microsoft Press

[34] SPSS Inc, More On What Data Mining İs – And İsn't. Www.Spss.Com/Datamine/What2.Htm (24.01.2005).

[35] TOKTAŞ, Peral – Demirhan, M.Başak; Risk Analizinde Veri Madenciliği Uygulamaları, Ya/Em-2004- Yöneylem Araştırması/Endüstri Mühendisliği Xxıv Ulusal Kongresi, 15-18 Haziran 2004, Gaziantep-Adana.

[36] T. Zhang, R. Ramakrishnan, And M. Livny. Birch : An Efficient Data Clustering Method For Very Large Databases. Sigmod'96vahaplar, Alper – İnceoğlu, M.Murat; Veri Madenciliği Ve Elektronik Ticaret Http://Www. Bayar.Edu.Tr/Bid/Dokumanlar/İnceoglu.Doc./24.01.2005).

[37] W. Wang, Yang, R. Muntz, Sting: A Statistical Information Grid Approach To Spatial Data Mining, Vldb'97.

[38] YARIMAĞAN, Ü., 2000, Veri Tabanı Sistemleri, Akademi & Türkiye Bilişim Vakfı, Ankara

[39] ZHOU, Z. ; Tree Perspectives Of Data Mining, Artificial Intelligence,143.

[40] WEKA 3, 2002, Machine Learning Software İn Java, Http://Www.Cs.Waikato.Ac.Nz /Ml/Weka.

ÖZGEÇMİŞ

1980'de İstanbul'da doğdu. İlk, orta ve lise öğrenimini doğduğu şehirde tamamladı. 1997 yılında girdiği Sakarya Üniversitesi Endüstri Mühendisliği Bölümü'nden 2002 yılında mezun oldu. Aynı yıl Sakarya Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Bölümü'nde yüksek lisansa başladı ve halen devam etmektedir.