T.C.
DOKUZ EYLÜL ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
EKONOMETRİ ANABİLİM DALI
EKONOMETRİ PROGRAMI
YÜKSEK LİSANS TEZİ

VERİ MADENCİLİĞİ VE BİR UYGULAMASI

Burhan GEMİCİ

Danışman
Prof. Dr. Kaan YARALIOĞLU

İZMİR - 2012

YÜKSEK LİSANS
TEZ/ PROJE ONAY SAYFASI

2007800340

Üniversite : Dokuz Eylül Üniversitesi
Enstitü : Sosyal Bilimler Enstitüsü
Adı ve Soyadı : Burhan GEMİCİ
Tez Başlığı : Veri Madenciliği ve Bir Uygulaması

Savunma Tarihi : 01.06.2012
Danışmanı : Prof.Dr.Kaan YARALIOĞLU

JÜRİ ÜYELERİ

Ünvanı, Adı, Soyadı	Üniversitesi	İmza
Prof.Dr.Kaan YARALIOĞLU	DOKUZ EYLÜL ÜNİVERSİTESİ	.....
Doç.Dr.İpek DEVECİ KOCAKOÇ	DOKUZ EYLÜL ÜNİVERSİTESİ	.....
Yrd.Doç.Dr.Yılmaz GÖKŞEN	DOKUZ EYLÜL ÜNİVERSİTESİ	.....

Oybirliği (X)
Oy Çokluğu ( )

Burhan GEMİCİ tarafından hazırlanmış ve sunulmuş "**Veri Madenciliği ve Bir Uygulaması**" başlıklı Tezi(X) / Projesi( ) kabul edilmiştir.

Prof.Dr. Utku UTKULU
Enstitü Müdürü

YEMİN METNİ

Yüksek Lisans Tezi olarak sunduğum **Veri Madenciliği Ve Bir Uygulaması**" adlı çalışmanın, tarafımdan, bilimsel ahlak ve geleneklere aykırı düşecek bir yardıma başvurmaksızın yazıldığını ve yararlandığım eserlerin kaynakçada gösterilenlerden oluştuğunu, bunlara atıf yapılarak yararlanılmış olduğunu belirtir ve bunu onurumla doğrularım.

Tarih

.../..../.....

Burhan GEMİCİ

İmza

ÖZET

Yüksek Lisans Tezi

Veri Madenciliği Ve Bir Uygulaması

Burhan Gemici

Dokuz Eylül Üniversitesi

Sosyal Bilimler Enstitüsü

Ekonometri Ana Bilim Dalı

Ekonometri Programı

Gelişen ve değişen teknolojiler sayesinde şirketler arasındaki rekabet hızlı bir artış göstermiştir. Bu rekabet ile birlikte şirketlerin bilgiye ulaşmaları büyük önem taşımaktadır. Bilgisayar teknolojilerinin gelişmesi ile veri tabanlarında çok büyük boyutlarda veri saklamak mümkün hale gelmiştir. Şirketler bu veriler ile kullanışlı bilgiye ulaşmayı hedeflemektedir. Bunun gereksinimi olarak veritabanlarında bilgi keşfi ve veri madenciliği kavramları ortaya atılmıştır. Bu kavramlarla amaçlanan, veritabanlarında saklanan veriler arasındaki gizli kalmış örüntüleri ortaya çıkarmaktır.

Bu tez kapsamında, veritabanlarında bilgi keşfi ve veri madenciliği tanımları üzerinde durulmuş, bu tanımlar doğrultusunda veri madenciliği süreci adımları incelenmiş ve veri madenciliği teknikleri ve algoritmaları anlatılmıştır. Veri madenciliği bileşenlerinden makine öğrenimi kavramına yer verilmiş ve makine öğrenimi için gerekli bilgisayar yazılımları önerilmiştir. Bu kavramlar ile İMKB (İstanbul Menkul Kıymetler Borsası)' de işlem gören 10 şirkete ait hisse senedi değerlerindeki değişmeler arasındaki birlikteliklerin ortaya çıkarılmasını amaçlayan bir uygulama gerçekleştirilmiştir. Bu uygulamada, veri madenciliği algoritmalarından biri olan apriori algoritması kullanılmış ve birliktelik kuralları ortaya çıkarılmıştır.

Anahtar Kelimeler: Veritabanlarında Bilgi Keşfi, Veri Madenciliği, Apriori Algoritması, Birliktelik Kuralları.

ABSTRACT

Master's Thesis

Data Mining And Its Application

Burhan Gemici

Dokuz Eylül University

Graduate School of Social Sciences

Department of Econometrics

Econometrics Program

Developing and changing technologies have shown a rapid increase in competition between companies. Companies to compete with this information is of great importance to reach. Databases with the development of computer technology has become possible to store very large volumes of data. Useful information with companies aiming to achieve these data. The concepts of data mining and knowledge discovery in databases as a requirement has been introduced. These concepts aim to reveal patterns hidden in data stored in databases.

This thesis focuses on the definitions of data mining and knowledge discovery in databases, data mining process steps in accordance with the definitions and data mining techniques and algorithms described were examined. Given to the concept of machine learning in data mining and machine learning components required for the proposed computer software. These concepts and the ISE (Istanbul Stock Exchange), belonging to 10 companies traded in the associations between changes in stock values was an application aimed at revealing. In this embodiment, one of an algorithm of data mining algorithms is apriori algorithm used, and association rules were uncovered.

Keywords: Knowledge discovery in databases, Data Mining, Apriori Algorithm, Association Rules.

VERİ MADENCİLİĞİ VE BİR UYGULAMASI

İÇİNDEKİLER

İKİNCİ BÖLÜM
VERİ MADENCİLİĞİ TEKNİKLERİ ve MODELLERİ

KISALTMALAR

A.B.D	Amerika Birleşik Devletleri
BIRCH	Balance Iterative Reducing and Clustering Using Hierarcihies
CART	Classification And Regression Trees
CRISP-DM	Cross - Industry Standard Process
CRM	Costumer Relationship Management
DBSCAN	Density-Based Spatial Clustering Method Based on Connected Regions with Sufficiently High Density
İMKB	İstanbul Menkul Kıymetler Borsası
SLIQ	Supervised Learning in Quest
SPRINT	Scalable Parallel Classifier for Data Mining

TABLOLAR LİSTESİ

ŞEKİLLER LİSTESİ

EK LİSTESİ

GİRİŞ

Veritabanı sistemlerinin gelişmesi ile çok sayıda veriyi bilgisayar ortamında uzun yıllar saklamak mümkün hale gelmiştir. Tera byte ile ölçülebilen bu veritabanları, kullanışlı bilgiler barındırmaktadır. Bu bilgilerin ortaya çıkarılması şirketler için büyük önem taşımaktadır. Bu bilginin ortaya çıkarılması için veri madenciliği teknikleri ve algoritmaları geliştirilmiştir. Veri madenciliği, istatistik, matematik ve bilgisayar bilimlerinin bir kesişimidir. Bundan dolayı 'disiplinlerarası' bir disiplin olarak nitelendirilir. Veri madenciliği, veritabanlarında bilgi keşfi olarak da adlandırılır.

Türkiye' de birliktelik kuralları ile pazar sepeti analizi uygulamaları mevcuttur. Bunlar süpermarketlerde hangi müşteri tipinin hangi ürünü ne kadar aldığını, ne sıklıkta aldığını tespit etmeyi amaçlamaktadır. Bunun yanında bir araştırmada birliktelik kuralları ortaya çıkarılarak bir süpermarkette reyon düzenlemesi yapılmıştır.

Türkiye' de sermaye piyasaları ile birçok veri madenciliği çalışması yapılmıştır. Bunlardan en önemlileri Dr. Ali Serhan Koyuncugil' in çalışmalarıdır. Bu çalışmalarda risk yönetimi amaçlanarak kümeleme ve sınıflandırma teknikleri ile tanımlama ve tahminleme yapılmıştır. Bunun yanında Dr. Engin Küçüksille, doktora çalışması olarak genetik algoritmaları kullanarak İMKB hisse senetleri piyasasında portföy performansı değerlendirmesini amaçlayan bir program geliştirmiştir. Bu konuda hisse senetlerinde birliktelik kurallarının tespiti için ASP.NET programlama diliyle geliştirilen basit programlar mevcuttur. Shu- Hsien ve diğerleri, Tayvan borsası için iki adımlı veri madenciliği çalışması yapmıştır. Bu adımlardan birincisi, apriori algoritması kullanarak hisse senetleri arasındaki birliktelikleri ortaya çıkarmış ve ikinci olarak kümeleme tekniklerinden k- ortalamalar algoritması ile hisse senetlerini kategorilere ayırmışlardır.

.

Bu çalışmanın birinci bölümünde, veri madenciliği tanımlarına yer verilmiş, veri madenciliğinin kullanım alanları ve uygulamaları aktarılmıştır. Veritabanlarında saklanan çok sayıda verinin kullanılabilir hale gelebilmesi için düzenlenmesi gerekmektedir. Bu konuya, veri önişleme kısmında geniş olarak yer verilmiştir. Makine öğreniminin ne olduğu tanımlanmış, makine öğrenimi için gerekli yazılımlar gösterilmiştir.

Çalışmanın ikinci bölümünde, veri madenciliği teknikleri anlatılmıştır. Her tekniğe ait algoritmaların işleyişine yer verilmiştir. Bu teknikler denetimli ve denetimsiz öğrenme olarak gruplandırılmıştır. Sınıflandırma teknikleri denetimli, birliktelik kuralları ve kümeleme teknikleri ise denetimsiz öğrenme ile uygulanır.

Çalışmanın üçüncü ve son bölümünde, birinci ve ikinci bölümde aktarılanlar ışığında bir uygulama çalışması sunulmuştur. Bu uygulama çalışması için İMKB' de işlem gören 10 şirket seçilmiştir. Uygulamada, bu şirketlere ait hisse senetlerindeki altmış iki günlük değişmeler arasındaki birliktelik kurallarını ortaya çıkarmak amaçlanmıştır. Birliktelik kurallarından apriori algoritması kullanılmıştır. Uygulamanın sonucunda bir model oluşturulmuş ve sonuçlar yorumlanmıştır.

BİRİNCİ BÖLÜM

VERİTABANLARINDA BİLGİ KEŞFİ VE VERİ MADENCİLİĞİ

1.1 Veri Madenciliği Tanımları

Her geçen gün, kullanışlı veya kullanışsız verilerin sayısı büyük artışlar göstermektedir. Her yerde var olan kişisel bilgisayarlar sayesinde, önceden işe yaramaz bulup sildiğimiz verileri saklamak çok kolay hale gelmiştir. Her yerde bulunabilen elektronik gereçler sayesinde, kararlarımız, süpermarketlerdeki seçimlerimiz, finansal alışkanlıklarımız kaydedilebilmektedir. Bu verilerin artması, insanların bu verilerden anlamlı çıkarımlarını azaltmıştır. Bu verilerin içinde gizlenen bilgi, kullanılabilir potansiyel bilgidir ve açık olarak bulunmamaktadır. Kullanılabilir potansiyel bilgiyi ortaya çıkarmak, verilerdeki örüntüleri keşfetmek veri madenciliği yöntemleriyle elde edilmektedir. Bu yeni bir anlayış değildir. İnsanlar, insanlığın başlangıcından beri verilerdeki örüntüleri araştırmaktadırlar. Avcılar, hayvanların göç davranışlarındaki, çiftçiler, ekinlerinin gelişmesindeki, politikacılar seçmenlerinin düşüncelerindeki örüntüleri araştırmışlardır. Günümüzde de girişimciler, fırsat yaratabilmek için kendilerine fayda sağlayacak iş örüntülerini araştırmaktadırlar (Witten ve Frank, 2005:4).

Bu veriler arasındaki anlamlı yeni korelasyonları ve örüntüleri keşfetme süreci veri madenciliği olarak tanımlanır. Diğer bir tanımla veri madenciliği, geniş gözlemli veri setlerindeki şüphe edilmeyen ilişkileri analiz etme ve yeni yollarla veri sahiplerine anlamlı ve kullanışlı bir şekilde özetleme yöntemidir (Hand ve diğerleri, 2001:14). Veri madenciliği, geniş veritabanlarından bilgiyi çıkarmak için; makine öğrenimi, örüntü tanıma, istatistik, veritabanı ve görüntüleme tekniklerini bir araya getiren disiplinler arası alan (Cabena ve diğerleri, 1998:13) olarak da tanımlanabilir.

Sumathi ve Sivanandam, veri madenciliği ve bilgi keşfi tanımlamalarını aşağıdaki gibi sıralamıştır;

- Veri madenciliği, Büyük veri setlerinden açık olmayan değerli bilgiye ulaşmak için etkili araştırmadır.
- Veri tabanlarında bilgi keşfi: Verilerdeki, kullanışlı ve anlaşılabilir geçerli potansiyel örüntüleri tanımlamak için kullanılan önemli bir süreçtir.
- Veri madenciliği: Değerli iş verilerindeki ilişkileri ve yeni durumları otomatik araştırmasıdır.
- Veri madenciliği, kullanılacak iş de, rekabet ortamında avantaj sağlayan bilginin keşfidir.
- Veri madenciliği: Veri tabanlarından anlaşılır model ve örüntüleri ortaya çıkaran tümevarımdır.
- Veri madenciliği: Geniş veri tabanlarından önceden bilinemeyen değerli ve kullanışlı bilginin ortaya çıkarılması ve kritik iş kararlarında kullanılması sürecidir.

Bunun yanında geleneksel veri analizi tekniklerini oluşturan regresyon analizi, kümeleme analizi, çok boyutlu analiz, diğer çok değişkenli istatistiksel yöntemleri, stokastik modelleme ve zaman serisi analizi birçok problemin çözümünde yaygın olarak kullanılır. Bu yöntemlerin öncelikli amacı sayısal çıkarımlar yapma ve istatistiksel veri özelliklerini ortaya çıkarmaktır (Larose, 2005:3).

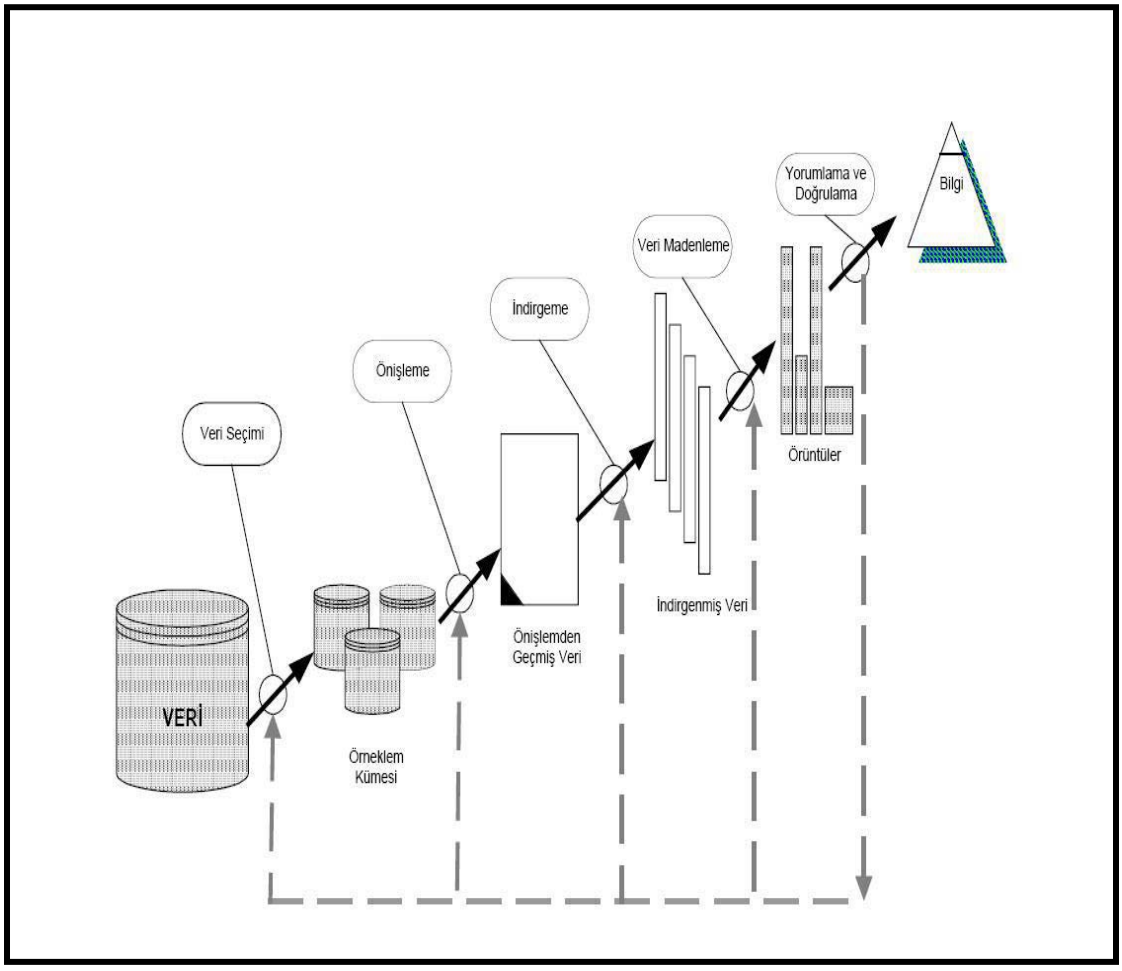
Örneğin istatistiksel analiz, veri setindeki değişkenler arasındaki korelasyona ve kovaryansına karar verebilir. Değişkenler arasındaki bağımlılıkları niteleyemeyebilir; bağımlılıkların neden oluştuğuna sıradan açıklamalar getirir. Merkezi eğilime ve belli faktörün varyansına, regresyon analizi ile de veri noktalar kümesine eğri uydurmaya karar verebilir ancak değişkenler arasındaki gizli kalmış örüntüleri ortaya çıkaramazlar. Bu durum da, gelişen bilgisayar teknolojisiyle kayıt altına alınan kullanışlı ve kullanışsız veriden bilgiyi elde etmeyi zorlaştırmış ve piyasa oyuncularının karar sürecini yavaşlatmıştır. Rekabet ortamının gelişmesi, karar sürecinde bilgiyi elde edememe ve ya yavaş elde etmeden dolayı piyasa oyuncularına önemli kayıplar yaşatmıştır (Sumathi ve Sivanandam, 2006:9).

Bundan dolayıdır ki uzun çalışmalardan sonra bilim adamları, yeni araştırma alanı olan *veri madenciliği ve bilgi keşfi* tanımlamalarını ortaya atmışlardır. Veri madenciliği bilgi keşfinin bir basamağı olarak da tanımlanmıştır.

1.2 Veritabanlarında Bilgi Keşfi Adımları

Veritabanlarında bilgi keşif süreci adımları Şekil 1‘ de gösterilmiştir.

Şekil 1: Veritabanlarında Bilgi Keşif Süreci Adımları



Kaynak: Özçakır, 2006, s. 2

Şekil 1’ de gösterilen bilgi keşif adımları ve bu adımların işlevleri aşağıda açıklanmıştır (Maimon ve Rokach, 2005:3).

- Veri seçimi ve araştırmaya uygun veri seti yaratma**:** Bilgi keşfi için amaca yönelik verilere karar verilmelidir. Bu, kullanışlı veriyi ortaya çıkarmayı, gerekli ek verileri elde etmeyi ve bilgi keşfi için kullanılacak bu verileri tek bir veri setine bütünleştirmeyi içerir. Bu süreç çok önemlidir çünkü veri madenciliği uygun verilerden öğrenir ve keşfeder. Bu model oluşturmanın temelidir. Eğer kullanılması gere bazı değişkenler eksik ise oluşturacağımız model hatalı olur

- Veri önişleme ve temizleme: Bilgi keşfinin bu basamağı verinin güvenilirliğini artırır. Veriler farlı kaynaklardan edinildiği için kodlama farklılıklarından, tarih farklılıklarından doğabilecek sonuçları engellemek için öncelikle veri değerlendirilir (Akpınar, 2000:5). Daha sonra veri temizleme aşamasına geçilir. Veri temizleme, kayıp verileri düzenleme ve gürültülü ve ya sapan verileri uzaklaştırma işlemleridir. Böylelikle veri, amaca uygun ve zaman kaybına neden olmadan işlenmiş olur.

- Veri dönüştürme veya indirgeme: Bu adım, hazır ve gelişmiş veri madenciliği için daha iyi veri üretmek için uygulanır. Bu iki adımla uygulanabilir; veri dönüştürme ve boyut indirgeme. Örneğin, yapay sinir ağı algoritması kullanılması halinde kategorik değişken değerlerinin evet/hayır olması, bir karar ağacı algoritmasının kullanılması durumunda ise örneğin gelir değişken değerlerinin yüksek/orta/düşük olarak gruplanması modelin etkinliğini arttırır. Bu adım bütün veri keşfi süreci için çok kritiktir (Akpınar, 2000:5).

- Veri madenciliği: Bu adım üç basamakta gerçekleşir.

a. Uygun veri madenciliği tekniğini seçmek: Bu adımda, kullanılacak veri madenciliği tekniğine karar verilir. Bu tekniklere örnek olarak sınıflandırma, regresyon ve kümeleme verilir. Bu, veri tabanlarından veri keşfinin amacına bağlıdır. Veri madenciliğinde iki amaç vardır. Bunlar tahminleme ve tanımlamadır.

Tahminleme için kullanılan veri madenciliği, denetimli veri madenciliği olarak adlandırılır. Tanımlama için kullanılan veri madenciliği ise denetimsiz veri madenciliği ve veri görüntüleme olarak adlandırılır.

b. Veri madenciliği algoritmasını seçmek: Kullanılacak olan teknik belirlendikten sonra bu teknikteki hangi algoritmanın kullanılacağına karar vermek gerekir.

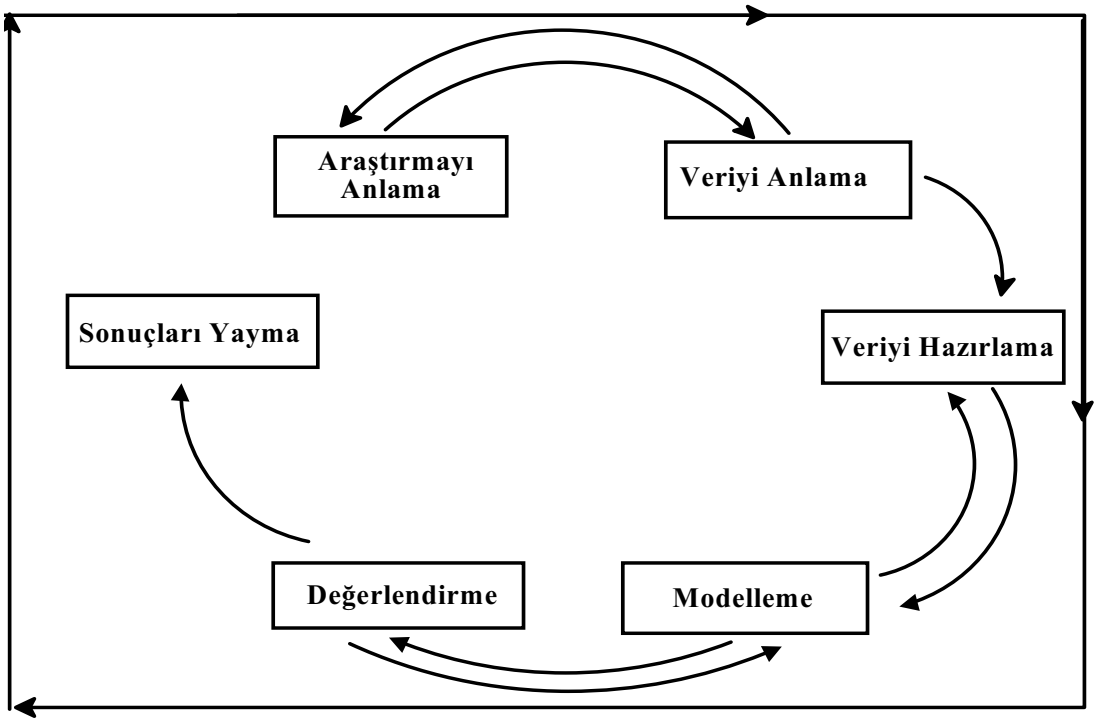
c. Veri madenciliği algoritmasını uygulamak: Algoritmaya karar verildikten sonra algoritma, kararlaştırılmış parametrelere ulaşana kadar değişkenler üzerinde denenmelidir. Tek yapraklı karar ağaçlarında, örneklerdeki en küçük sayıya ulaşana kadar denenmesi buna örnek gösterilebilir.

- Değerlendirme: Bu adımda, çıkarılmış örüntüler ilk adımda belirlenen amaca uygun olarak değerlendirilir ve yorumlanır. Elde edilen modelin anlaşılabilirliği ve kullanışlılığı araştırılır. Keşfedilen bilginin daha sonraki kullanımları için belgelendirilir. Özetle, veri madenciliği ile elde edilen örüntülerin ve keşiflerin kullanımı ve etraflı bir geri beslemesidir.

1.3 CRISP-DM

Veri tabanlarında bilgi keşfi süreci adımlarına benzer olarak bazı kurumsal şirketler, departmanları arasındaki bölünmüşlük ve birbirleriyle etkileşimlerinin olmaması nedeniyle veri madenciliğine yeni bir standart getirmiştir. CRİSP-DM, 1996 yılında DaimlerChrysler, SPSS ve NCR.CRISP şirketlerinden analistler tarafından kişiye özel olmayan ve özgürce kullanılabilir standart süreç olarak geliştirilmiştir. CRISP-DM ‘e göre, verilen veri madenciliği projesi Şekil 2‘ de gösterilen altı basamaklı bir yaşam döngüsüne sahiptir (Larose, 2005:5).

Şekil 2: CRİSP-DM Döngüsü



Kaynak: Larose, 2005, s.6

CRISP-DM yaklaşımı büyük veri madenciliği projelerinin daha hızlı, etkili, güvenilir, yönetilebilir ve az maliyetle sonuçlandırılmasını sağlar.

CRISP-DM basamakları ve bu basmakların tanımları aşağıda gösterilmiştir (Larose, 2005:6).

1.Yapılacak işi veya araştırmayı anlama

a. Projenin amaçları ve gereksinimleri açıkça ifade edilmelidir.

b. Bu amaç ve kısıtlamalar veri madenciliğinin problem tanımına uygun olarak formüle edilmelidir.

c. Bu amaçları gerçekleştirmek için ön hazırlığa başlanmalıdır

2. Veriyi anlama

a. Veri toplanır.

b. Verileri hakkında ön bilgiye sahip olmak için keşfedici veri analizi kullanılır.

c. Verinin yapısı değerlendirilir.

3. Veriyi hazırlama

a. Amaç, ilk işlenmemiş veriyi kullanılacak olan veri setine hazırlamaktır.

b. Analiz için uygun değişkenleri seçilmelidir.

c. Kullanılacak olan değişkenlerde ihtiyaca göre dönüşümler yapılmalıdır.

d. Kayıp veriler düzenlenmelidir.

4. Modelleme

a. Neyi amaçladığımıza bağlı olarak uygun veri madenciliği tekniği seçilir.

b. Eğer elde edilen modelle bulunan sonuçların yanlış ve tutarsız olduğu düşünülürse, ilk basamağa dönülür ve mevcut olan başka değişkenler modele eklenir (Berthold ve diğerleri, 2010:10).

5. Değerlendirme

a. Ortaya çıkan sonuçlar, problemin veya işin sahiplerinin bakış açılarından tartışılır ve uygun olup olmadığı analiz edilir.

b. Model uygun ise bir sonraki adıma geçilir.

c. Bu adımda, yeterli olmayan sonuçlar nedeniyle proje durdurulabilir ve analiz için kullanılan nesneler tekrar gözden geçirilebilir.

6.Sonuçları yayma

a. Eğer proje sonuçları sürekli olarak kullanılacak ise bulduğumuz model raporlanır (Berthold ve diğerleri, 2010:10).

b. Modeli kullanacak olan şirketin, kurumun vb. departmanları modelden haberdar edilir.

1.4 Veri Madenciliği Ne Değildir ?

Toplanan verilerden yapılacak sorgulamalar ve detaylı analizler ile elde edilen sonuçlar veri madenciliği olarak değerlendirilmemelidir. Örneğin; bir süper market zincirinde, şubelerin cirolarını ve hangi ürünlerin hangi şubede daha fazla satıldığını sorgulamak, bir satış şirketinde hangi müşterilerin süreklilik gösterdiğini belirlemek tam bir veri madenciliği olarak değerlendirilemez. Aynı şekilde yalnızca regresyon analizi yaparak gelir ile cinsiyet arasındaki ilişkiyi modellemek de veri madenciliği değildir (Argüden ve Erşahin, 2008:17).

Veri madenciliğinin ne olmadığı ve ne olması gerektiği Tablo 1‘ de birkaç örnekle gösterilmiştir.

Tablo 1: Veri Madenciliği Ne Değildir ?

NE DEĞİLDİR	NE OLMALIDIR
İnternetten ayrıntılı bilgi araştırmak	İnternette aynı içerikteki benzer bilgileri gruplamak
Aynı hastalığa sahip hasta kayıtlarını sorgulamak	Benzer semptomlar görülen aynı hastalığa sahip hastaları gruplamak
Yer listesinden termal otellerin yerini sorgulamak	Termal otelleri, hangi hastalığın tedavisi ile ilgili olduğuna göre gruplamak
Şirketlerin finansal raporlarından tabloları analiz etmek	Şirketlerin satış ile ilgili veri tabanlarından müşteri profillerini ortaya çıkarmak

Kaynak: Gorunescu, 2011, s.4

1.5 Veri Madenciliğinin Kullanım Alanları

Veri madenciliği teknikleri, bilgisayar teknolojilerinin gelişmesiyle iş, bilim ve spor alanlarında sıklıkla kullanılmaktadır. Bir araştırma şirketi tarafından yapılan bir araştırmanın sonuçları A.B.D 'de veri madenciliği pazar hacminin 3 milyar dolar olduğunu göstermektedir (Akpınar, 2000:2).

- Pazarlama yönetimi: Hedef pazarlama, müşteri ilişkileri yönetimi, Pazar sepeti analizi, çapraz satış analizi, müşteri değerlendirme, mevcut müşterilerin elde tutulması için yapılacak pazarlama strateji analizleri (Sumathi ve Sivanandam, 2006:27).
- Bankacılık ve finans: Risk yönetimi, rekabet analizleri, karlılık analizleri, müşteri kaybını engelleme, kredi onayı değerlendirmeleri, kredi kartı harcamalarına göre müşteri gruplarının belirlenmesi, dolandırıcılık tespiti, genel piyasa analizleri, hisse senedi fiyat analizleri, farklı finansal göstergeler arasındaki örüntülerin bulunması (Kalikov, 2006:10).
- Telekominikasyon ve medya: Pazarlama kampanyaları yönetimi, müşteri bölünmeleri, karlılık analizleri, telekominikasyon hatlarında yoğunluk tahminleri (Sumathi ve Sivanandam, 2006:27).
- Sağlık: Ürün geliştirme, tedavi sürecinin belirlenmesi, önceki verilerden faydalanarak hastalık tahmini (Kalikov, 2006:10).
- Endüstri: Kalite kontrol çalışmaları, lojistik, üretim süreçleri optimizasyonu (Kalikov, 2006:10).

Tablo 2' de veri madenciliğinin kullanım alanlarının sektörel oranları verilmiştir. Bu tabloya göre veri madenciliği, en çok CRM/Müşteri analitiği alanında kullanılmaktadır. Bunu %24.4 oranıyla bankacılık sektörü takip etmektedir. Tablo 2' de görülüğü gibi hemen hemen her sektörde veri madenciliği kullanılmaktadır.

Tablo 2: Veri Madenciliği Kullanım Alanları

Kullanım Alanı	Kullanım Oranı (%)
CRM/Müşteri Analitiği	32.8
Bankacılık	24.4
Direk Pazarlama	16.1
Kredi Puanlama	15.6
Telekominikasyon	14.4
Dolandırıcılık Tespiti	13.9
Satış	11.7
Sağlık	11.7
Finans	11.1
Bilim	10.6
Reklamcılık	10.6
E-Ticaret	10.0
Sigortacılık	10.0
Web Madenciliği	8.3
Sosyal Ağlar	7.8
İlaç	7.8
Bioteknoloji	7.8

Kaynak: Gorunescu, 2011, s.41

1.6 Veri Madenciliği Uygulamaları

Veri madenciliği süreci, verilerin kaydedilip saklanabildiği birçok yerde uygulanmaktadır. Bu uygulamalar, birliktelik kuralları, kaçakçılık tespiti, astronomik veriler, genomik veriler, doküman verileri, üretim verileri, eğitim verileri, spor verileri, müzik çalışmaları, yatırım kararları olarak sıralanır.

1.6.1 Birliktelik Kuralları

Süpermarketlerden alışveriş yapan müşteriler, belirli zamanlarda farklı ürünler satın alırlar. Satın alınan ürünlerin kim tarafından ne zaman alındığı barkod okuyucular sayesinde veritabanına aktarılmakta ve geniş veri tabanlarında saklanmaktadır. Temel problem, hangi ürünün birlikte satın alınma eğilimi olduğudur. Bu üstü kapalı bir birliktelik problemidir. Bu birliktelikleri ortaya çıkarmak için birçok birliktelik algoritması geliştirilmiştir (Sumathi ve Sivanandam, 2006:29). Bunun yanında, piyasaya yeni çıkmış bir ilacın yan etkilerinin hangi durumlarda

ortaya çıktığının araştırılmasında, telekominikasyon ağlarındaki sorunları tahminlemede birliktelik kuralları kullanılır (Larose, 2005:17).

1.6.2 Kaçakçılık tespiti

Diğer sahtekârlıklara nazaran kredi kartı işlemlerinde yapılan sahtekârlıklar az olsa da, bu yolla her yıl 500 milyon dolar kayıp yaşanmaktadır. Bunun için kullanılan veri madenciliği teknikleri geliştirilmiştir (Sumathi ve Sivanandam, 2006:29). Müşteriler hakkında her türlü veriyi veritabanında saklayan bankalar, yapılan kredi kartı işlemlerinin müşteriler için kurulan modele uygun olup olmadığını bu tekniklerle tahmin edebilirler.

1.6.3 Astronomik Veriler

Astronotlar tarafından yeni galaksi, yıldız ve gökcisimleri fotoğraflanarak incelenmektedir. Son zamanlarda ise yeni astronomik keşif sürecini makineleştirmek için sınıflandırma algoritmaları kullanılmaktadır. Sınıflandırma algoritmaları, gökyüzü nesnelerinin, parlaklığı, alanı ve şekli gibi görüntü işleme kanalı ile üretilen değişkenleri türetmek için uygulanır. Bu yaklaşım, geleneksel hesaplama teknikleri ve elle yapılan analizlerle tespit edilen zayıf gözlemlerin aksine daha kullanışlıdır. Bu yaklaşım, Palomar Gözlemevi' nin gökyüzü haritasındaki gökcisimlerini 3' e katlamıştır (Sumathi ve Sivanandam, 2006:30).

1.6.4 Genomik Veriler

Genomik veriler bütün dünyada, farklı formatlarda ve farklı uygulama yönetimleriyle kaydedilmektedir. Yeni sistemler, gen karşılaştırmalarına, gen tanımlamalarında ve bütün gen işlevinin yorumlanmasına ve analizine olana sağlamaktadır (Sumathi ve Sivanandam, 2006:30).

1.6.5 Doküman verileri

Doküman veri madenciliğinde (text mining) ana amaç dokümanlar arasında ayrıca elle bir tasnif gerektirmeden benzerlikleri ortaya çıkarabilmektir. Bu genelde, otomatik olarak çıkarılan anahtar sözcüklerin tekrar sayısı sayesinde yapılır. Polis kayıtlarında mevcut rapora benzer kaç adet ve hangi raporlar var, ürün tasarım dokümanları ve internet dokümanları arasında mevcut tasarım için kullanılabilecek ne tür dosyalar var gibi sorulara bu yöntemle yanıt bulunabilir (Akgöbek ve Çakır, 2009:803). Günümüzde Google arama motorunun işleyişi bu şekildedir. Kullanıcı aramak istediği şeyin sadece bir kelimesini yazsa bile, yazdığı kelimeye uygun sonuçları ekranında görebilir.

1.6.6 Üretim Verileri

General Electric ve SNECMA ortaklığı ile geliştirilen CASSIOPEE hata bulma sistemi üç önemli Avrupa havayolu şirketine ait Boeing 737 tipi uçaklardaki problemleri tespit ve tahmin etmek için kullanılmıştır. Hataların ortaya çıkarılması için kümeleme yöntemleri uygulanmıştır. CASSIOPEE, Avrupa'nın ilk yenilik uygulaması ödülü alan sistemidir (Fayyad ve diğerleri, 1996:38).

1.6.7 Yatırım Kararları

Çok sayıda şirket yatırım kararları için veri madenciliğinden faydalanır ama çoğu kullandıkları sistemi açıklamaz. LBS şirketi sistemini açıklayanlardan biridir. Onun sistemi, 600 milyon dolarlık portföyü yönetmek için, uzman sistemleri, yapay ağları ve genetik algoritmaları kullanır (Fayyad ve diğerleri, 1996:38).

1.6.8 Spor Verileri

Spor dünyasında, her takımdan, oyuncudan, oyundan ve sezondan çok fazla sayıda veri biriktirilebilmektedir. Örneğin, basketbolda her oyuncunun rebound, asist, top çalma, blok ve turnike istatistikleri her oyunda kaydedilir. Bu durum da birçok

üstü kapalı bilgi demektir. Bu yüzden veri madenciliğinin sporda kullanımı idealdir. Son zamanlarda, birçok takım sporu organizasyonu, birçok yeteneği keşfetmek ve var olan oyuncularının eksikliklerini tespit etmek ve rakip takımı analiz edip karşılaşmalarda yeni takım stratejileri ortaya koymak amacıyla istatistikçi ve analist çalıştırmaktadır (Solieman, 2006:4). Amerikan Basketbol Ligi' nde takım stratejileri veri madenciliği yöntemleriyle hazırlanmaktadır.

1.6.9 Müzik Çalışmaları

Müzik veritabanı fonksiyonlarında 'k' en yakın komşuluk yöntemi kolayca kullanılır. Burada amaç, müzik araştırmalarını geliştirmek ve çözümlemektir. (Jensen, 2006:3) Belli bir müzik parçası verildiğinde buna benzeyen diğer müzik parçalarını binlerce parça arasından veri madenciliği yöntemleri ile tespit edilebilir. Buradaki değişkenler, tempo, tarz, artikülasyon olarak tanımlanabilir (Sevinç, 2005).

1.6.10 Eğitim Sektörü Verileri

Öğrenci işlerinde veriler analiz edilerek öğrencilerin başarı ve başarısızlık nedenleri, başarının arttırılması için hangi konulara ağırlık verilmesi gerektiği, üniversiteye giriş puanları ile okul başarısı arasında bir ilişkinin var olup olmadığı gibi sorulara yanıt arayarak eğitim kalitesini arttırmak ve eğitim politikalarını belirlemek amacıyla veri madenciliği kullanılır (Akgöbek ve Çakır, 2009:802).

1.7 Makine Öğrenimi

1.7.1 Öğrenme Nedir ?

İnsanlar, yaşayabilmeleri ve çevreye adapte olabilmeleri için öğrenmeye ihtiyaç duyarlar. İnsanların ne öğrendikleri tanımlanamaz ancak öğrendikleri şeylere nasıl karar verdikleri tanımlanabilir. Öğrenmeyi tanımlayabilmek için iki ana kavram vardır. Bunlar, iyi ya da kötü gerçekleştirilmesi gereken görev ve bu görevi gerçekleştirmek için kullanılacak bir tema ya da konudur. Kısaca öğrenme, insanların,

belirli bir görevin gerçekleşmediği bir durumdan aynı görevin aynı şartlar altında nasıl gerçekleştiğini kavrama durumudur (Adriaans ve Zantinge, 1996:12).

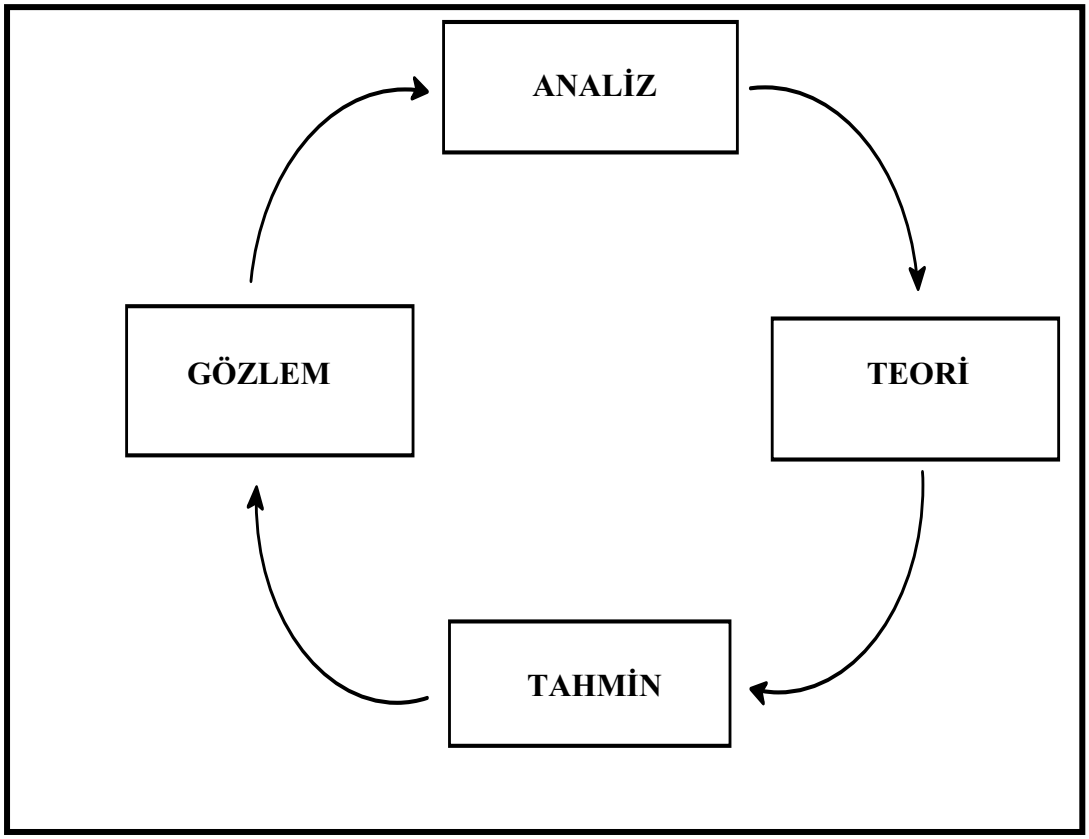
1.7.2 Kendi Kendine Öğrenen Bilgisayar Sistemleri

Bilgisayarlar, tanıma göre öğrenmeye yeteneklidirler. Bir bilgisayar kendi başına bir görevi veya işi gerçekleştiremeyebilir ancak insan eğer doğru komutları verirse öğrenmeyi gerçekleştirebilir. Örneğin kullanıcı bilgisayarın diferansiyel denklemi çözmesini istiyorsa doğru bir program geliştirerek bilgisayara bunu çözdürebilir. Bu örnek, kendi kendine öğrenen bilgisayarlar için eksik kalmaktadır. Kendi kendine öğrenen bilgisayarlar, kendi programlarını üretip yeni görevleri gerçekleştirmeye olanak sağlarlar. Kullanıcının program geliştirmesi ile bilgisayarın probleme çözüm üretmesi, onun tek başına öğrendiği anlamına gelmemektedir. Bilgisayarlar, insanlara göre problemleri çok hızlı ve çok doğru çözseler de, yaratıcılık kullanamazlar. Örneğin, bilgisayarlar bir bulmacayı çözebilirler ya da milyonlarca kayıt içeren pazarlama veritabanından örüntüleri kolayca bulabilirler ancak bir cinayeti çözemezler, bir pazarlama planı ortaya koyamazlar (Adriaans ve Zantinge, 1996:13). Bu yüzden günümüzde robot teknolojileri ve yapay zeka çalışmaları bilgisayarlara yaratıcılık kazandırmayı ve karar verme kapasitelerini artırmayı hedeflemektedir.

1.7.3 Bilim Metodolojisi

Modern bilim adamlarının temel görevi var olan bir şeyi açıklama ve olmayan bir şeyi tahminlemektir. Bilimsel araştırmaların ideal döngüsü Şekil 3’ de gösterilmiştir.

Şekil 3: Bilimsel Araştırmaların Yaşam Döngüsü



Kaynak: Adriaans, Zantinge, 1996, s.14

Bu döngüde yer alan düğümlerin açıklamaları aşağıda verilmiştir.

- Gözlem: Araştırmaya gözlemlerle başlanır.
- Analiz: Bu gözlemlerdeki örüntüler bulunmaya çalışılır.
- Teori: Eğer belirli düzenlilik elde edilirse, analiz formüle edilir ve teorileştirilir. Teori bir hipotezdir.
- Tahmin: Kurulmuş olan teori, yeni gözlemler tarafından doğrulanabilen yeni olguları tahminler.

Bu yüzyılda deneysel gözlemleri açıklamak için teoriler formüle edilebilir ancak bunların kesinliği hiçbir zaman kanıtlanamaz. Bilimin keşfettiği her şey geçici değerlere sahiptir. Örneğin, kuğuların renkleri formüle edilmek istensin. Gözlemlenen bir kaç kuğunun beyaz renkte olduğu kaydedildi. Buna bağlı olarak " bütün kuğular

beyazdır" hipotezi ortaya atıldı. Bu teorinin doğrulanması için sonsuz sayıda gözleme ihtiyaç vardır. Bütün gözlemlere kaydetmek neredeyse imkânsızdır. Diğer taraftan eğer bir tane bile siyah kuğu gözlemlenirse teorinin doğruluğu kalmayacaktır. Filozof Karl Popper' e göre *genel kanunlar, sınırlı sayıda gözlemle doğrulanamazlar ancak, bir tek gözlemle reddedilebilirler.* Popper bu kuralı, doğrulama ve reddetme arasındaki asimetri olarak adlandırmaktadır. Bu nedenle eldeki verilere uygun teoriler geliştirirken, teorinin reddedildiği durumlar da formüle edilmelidir. Hipotezleri doğrulamak her zaman kolay değildir. Örneğin, bir ilacın etkinliği üzerinde çalışan bir araştırmacı yüz hastadan oluşan bir örneklem seçsin. İlk elli hastaya ilaç, diğer elli hastaya plasebo versin. Eğer ilk gruptaki hastaların hepsi iyileşir ve diğer gruptaki hastaların hiç biri iyileşmez ise ilacın etkin olduğu söylenebilir. Yine ilk gruptaki kırk kişi iyileşir ve diğer gruptaki sadece 10 işi iyileşme eğilimi gösterirse ilacın yine etkin olduğu söylenebilir. Bunun yanında ilk gruptan otuz hasta iyileşir ve diğer taraftan on hasta iyileşirse ilacın az etkili veya etkisiz olduğu söylenemez. Bu durum istatistiksel olarak anlamsız diye adlandırılır (Adriaans ve Zantinge, 1996:17).

1.7.4 Makine Öğrenimi Tanımları

Bilgisayar yardımıyla bir problem çözümlenmek istenirse, probleme uygun algoritmalar geliştirmek gereklidir. Günümüzdeki teknolojik gelişmeler sayesinde, veritabanlarında milyarlarca veri kaydedilmekte ve bu verilerden çıkarsamalar yapılmaktadır. Bu verilerdeki örüntüleri ve düzenlilikleri araştırmak için birçok algoritma geliştirilmiştir. Bu algoritmalar programlanarak makine öğreniminin bir parçasını oluştururlar. Makine öğrenimi, bilgisayarların, algılayıcı verisi ya da veritabanı gibi veri türlerine dayalı öğrenimini olanaklı kılan algoritmaların tasarım ve geliştirme süreçlerini konu edinen bir bilim dalıdır. Ancak makine öğrenimi sadece veritabanı problemi değil aynı zamanda yapay zekânın bir parçasıdır. Aynı zamanda makine öğrenimi, robot teknolojilerinde ve görüntü ve ses tanıma sistemlerinde birçok probleme çözüm üretirler (Alpaydın, 2010:3).

Makine öğrenimi, örnek verileri ve geçmiş tecrübeleri kullanırken, performans ölçütlerini en uygun şekilde kullanmak için bilgisayar programları geliştirmektir. Örneğin, birkaç parametreli model tanımlansın, "Öğrenme", bilgisayar programını uygulamaya geçirecek ve program eğitim verilerini ya da geçmiş tecrübeleri kullanarak model parametrelerini en uygun hale getirecektir. Makine öğrenimi, matematiksel model kurabilmek için istatistik teorilerinden faydalanır çünkü ana görev örneklemden tahminlemeler yapmaktır. Makine öğrenimi aynı zamanda bilgisayar bilimlerinden, hem eğitim için hem de öğrenilmiş modelin gösterimi ve tahminlemelere algoritmik çözümler sunabilmek için faydalanır (Alpaydın, 2010:4).

Makine öğrenimi, tecrübelerden elde edilen bilgileri makineleştirerek hesaplama yöntemlerinde performansı artırmak için kullanılan bir çalışmadır. Makine öğrenimi, bilgi mühendisliği sürecinde otomasyon düzeyini artırmayı, eğitim verilerindeki örüntülerin keşfedilmesi sürecinde etkinliği arttıran otomatik tekniklerin, çok fazla zaman kaybına neden olan insan gücünün yerine geçmesini amaçlamaktadır (Jackson, 2002:272).

Makine öğrenimin başlıca uygulamaları, makine algılaması, bilgisayarlı görme, doğal dil işleme, sözdizimsel örüntü tanıma, arama motorları, tıbbi tanı, beyin-makine arayüzleri, kredi kartı dolandırıcılığı denetimi, borsa çözümlemeleri vb. olarak sıralanabilir.

1.7.5 Makine Öğreniminde Kullanılan Programlar

Makine öğrenimi uygulamalarında kullanılan programlar, her hangi bir lisans ücreti ödemeden edinebildiğimiz özgür yazılımlar ve lisans ücreti ödeyerek elde edebildiğimiz yazılımlar olarak ikiye ayrılır. Lisans ücreti ödeyerek elde edilen programlar Tablo 3' de gösterilmiştir.

Tablo 3: Makine Öğreniminde Kullanılan Lisanslı Yazılımlar

IBM-SPSS (Clementine)	Angoos software (Knowledge seker)
IBM (Data Warehouse)	Knowledge Builder Rules Authoring Std.
MICROSOFT (SQL Server)	SAP- Business intelligence solution
STATISTICA	CART 6.0 ProEx
MATLAB (ARMADA Data mining too)	Cloud1305
SAS Data mining-Enterprise Miner	Data Applied (Data Mining tools)
ORACLE Data Mining	Excel (XLMiner)

Kaynak: Gorunescu, 2011, s.39

Herhangi bir lisans ücreti ödemeden elde edilebilen özgür yazılımlar Tablo 4' de gösterilmiştir.

Tablo 4: Özgür Yazılımlar

ADaM	KEEL	RapidMiner
AlphaMiner	KNIME	Rattle
CRAN Task View	Machine Learning in Java (MLJ)	StarProbe
Databionic ESOM Tools	MiningMart	TANAGRA
ELKI	MLC++	Weka
Gnome Data Mining Tools	Orange	YALE

Bütün bu yazılımlar kendi resmi sitelerinden ücretsiz olarak elde edilebilir.

1.8 Veri Önişleme

Veri madenciliği çalışmalarında karşılaşılan en önemli sorunlar verilerdeki eksiklikler, araştırmaya uygun verilerin seçilmemesi, seçilen verilerin arasında yüksek korelasyon bulunması, sapan değerler gibi sorunlardır. Bu sorunlar, yanlış bulgular elde edilmesine ve çalışmaların uzamasına neden olmaktadır. Bunun için veri madenciliği adımlarından olan veri önişleme başka bir deyişle veri hazırlama bütün veri madenciliği sürecinin en önemli adımıdır.

1.8.1 Veri seçimi

Bir veri ambarı, birbirinden çok farklı veriler içerir ve bu verilerin hepsi, her veri madenciliği çalışmasının amacını gerçekleştirmek için kullanılamaz. Bu durumda amaca uygun verileri seçmek gerekir. Örneğin, market veritabanları, müşterilerin satın aldıkları malların, demografik özelliklerinin, tercihlerinin verilerini içerir. Market yönetimleri, hangi müşterilerin ne tür mallar satın aldıklarını tanımlamak isterse, demografik ve satın alınan mal verilerini kullanmaları gerekir. Oysa veri ambarında yukarıda belirtildiği gibi farklı birçok veri bulunmaktadır (Sumathi ve Sivanandam, 2006:197).

Veri seçimi aşamasında yapılması gerekenler;

a. Farklı ortamlardaki verilerin mevcut yapılarının incelenmesi ve tablo yapılarının incelenmesi,

b. Hedeflenen sonuca ulaşmak için gerekli verilerin, veri madenciliği uygulamak için belirlenen veri depolama ortamına transfer edilmesidir (Özçakır, 2006:12).

1.8.2 Kayıp Verilerin Düzenlenmesi

Verilerdeki bazı değerler çeşitli nedenlerle kaybolmuş, silinmiş, girilmemiş olabilir. Bu veriler kayıp veri olarak adlandırılır. Veri madenciliği çalışmalarında en sıkıntı veren ve zaman kaybına neden olan sorun veritabanında kayıp değerlerin bulunmasıdır. Kayıp veriler ya veritabanından çıkarılmalı ya da bunların yerine kullanıcı tarafından uygun teknikler kullanılarak yeni veriler girilmelidir. Yeni veri girişinde kullanılabilecek uygun teknikler aşağıda açıklanmıştır (Silahtaroğlu, 2008:21).

a. Tüm kayıp verilere aynı bilgiyi girmek: Örneğin, medeni hal verilerinde boş olan yerlere boş anlamına gelen “B” harfini girmek. Ancak bu durumda medenin halin “B” olması anlamlı bir sonuçmuş gibi çıkabilir. Kullanımı çok yaygın değildir.

b. Kayıp verilerin yerine tüm verilerin ortalama değerinin verilmesi: Örneğin kayıp ağırlık verilerinin yerine bütün ağırlık verilerinin ortalamasının verilmesi

c. Regresyon yöntemi kullanılarak diğer değişkenlerin yardımı ile kayıp verilerin tahminlenmesi: Eksik olmayan veriler kullanılarak regresyon denklemi elde edilebilir ve böylelikle kayıp veriler tahminlenebilir. Aynı şekilde Bayesyen sınıflandırma, karar ağaçları gibi teknikler kullanılarak da tahminlemeler yapılabilir.

1.8.3 Sapan Veriler

Sapan değerler, veri genişliğinin sınırlarını aşırı derecede aşan değerlerdir. Bu değerler değişkendeki diğer verilerin eğilimlerini de etkilerler. Bu değerler yanlış girilmiş olabilir. Yanlış girilmemiş olsa dahi veri madenciliği sonuçlarını etkileyeceğinden tespit edilmelidir (Larose, 2005:34).

Sapan değerlerden kurtulmak için;

a. Eğer gözlem yanlış girilmişse doğru bilgiye ulaşılmalı,

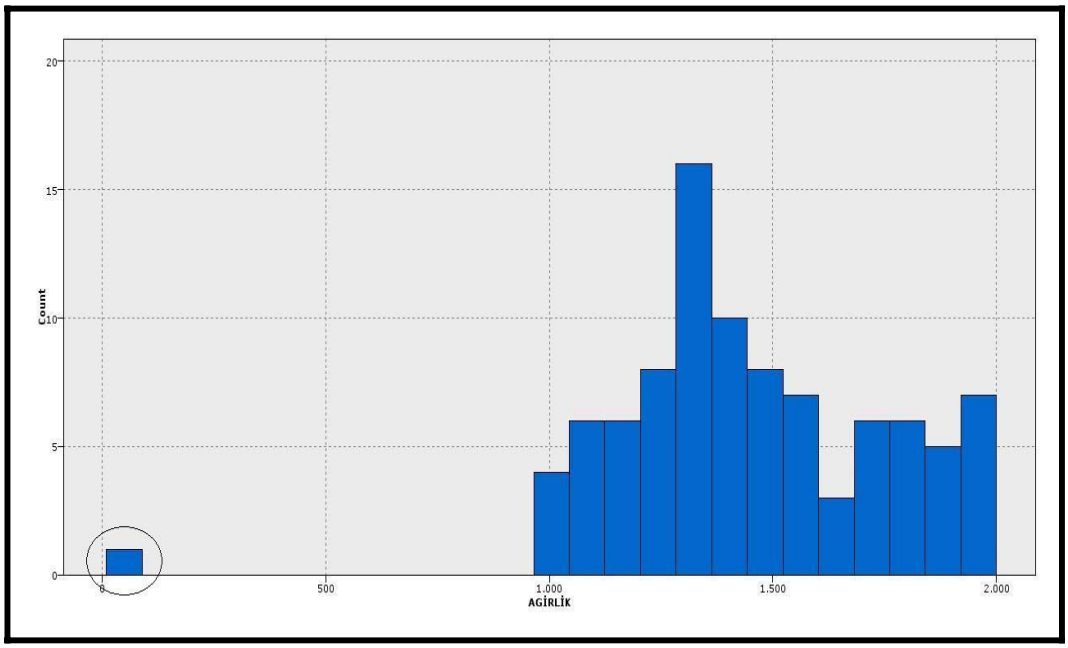
b. Herhangi bir yanlışlık yoksa sapan değer gözlemlerden çıkarılmalı,

c. Uygun dönüşümler yapılmalıdır.

Bu işlemlerden, sonuçları aşırı etkilemeyecek olan herhangi birisi seçilerek sürece devam edilebilir.

Sapan değerleri tespit etmek için bilgisayar programlarından yararlanılır. Bu programlar ile histogram, serpilme diyagramı ve kümeleme analizleri elde edilerek sapan değerlerin görselleştirilmesi sağlanır. Bu görselleştirmelere örnek, Şekil 4 ve Şekil 5 verilebilir.

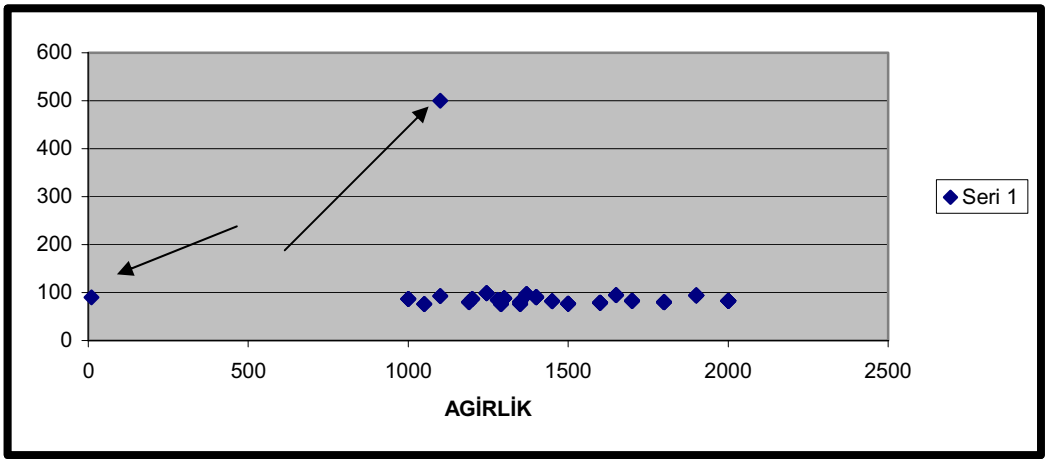
Şekil 4: Histogram İle Sapan Değer Tespiti



Kaynak: Larose, 2005, s.34

Bu şekil, arabaların gözlem değerlerine göre ağırlıklarının histogramıdır. Burada daire içine alınmış gözlemler diğer gözlemlerden aşırı sapma göstermiştir. Bu veriler incelenmeli ve önlemleri alınmalıdır. Veri madenciliği sürecinin ileri aşamalarında yanıltıcı sonuçlar doğurabilir.

Şekil 5: Serpilme Diyagramıyla Sapan Değer Tespiti



Kaynak: Larose, 2005, s.35

Bu serpilme diyagramı da, gözlenen arabaların ağırlıklarıyla, bir depo benzinle gidebildikleri yol (mil cinsinden) arasındaki ilişkiyi göstermiştir. Ok işareti ile gösterilen değerler verilerden oldukça saptığı açıkça görülmektedir.

1.8.4 Verilerin Yeniden Yapılandırılması

Veri madenciliği uygulamalarında, veriler her algoritma için aynı değildir. Bazı algoritmalar sadece sayısal değerlerle çalışırken bazıları kategorik değerlerle çalışır. Bazı algoritmalar ise 1 veya 0 ile kodlanmış değerlerle çalışır. Örneğin, bir süper market veri tabanından bir ay içinde satışı yapılan mallar veri setini oluştursun. Uygulamanın amacı ise bu mallar arasındaki birlikteliklerin var olup olmadığını araştırmak olsun. Bu durumda uygulama için, verilerin satılmaları durumu 1, satılmama durumu ise 0 olarak kodlanması gerekecektir. Bundan dolayı veri seti amaca uygun olarak yeniden yapılandırılacaktır. Diğer bir örnek olarak karar ağaçları verilebilir. Karar ağaçları, sürekli değerler yerine aralıklı değerler kullanırlar. Örneğin, ağırlık değişkeni 500 ile 10000 arasında değerler alıyorsa, bu değerler, 500-1000, 1000-1500 vb. gibi aralıklara bölünerek karar ağaçları uygulanacaktır (Silahtaroğlu, 2008:25).

Uygulamada karşılaşılan diğer bir durum ise değişken noktalarının birbirlerinden çok uzak genişliğe yayılmış olmasıdır. Bu durum bazı veri madenciliği algoritmalarında gereksiz ve yanıltıcı sonuçlara neden olabilir. Bu durumdan kurtulmak ve her değişkenin sonuca etkisini artırmak için normalleştirme uygulamaları kullanılır (Larose, 2005:35).

1.8.4.1 Min-max Normalleştirmesi

Bu normalleştirme, esas veriler üzerinde doğrusal dönüşüm yapmayı ifade eder (Han ve Kamber, 2006:96). Bu dönüşüm, aşağıdaki formülün bütün değişkenlere uygulanması ile elde edilir.

$$s' = \frac{s - \min}{\text{maks} - \min} \tag{1.1}$$

Burada *min*, verinin alabileceği en küçük değer, *maks* ise alabileceği en yüksek değeri gösterir. *s'* ile verinin dönüştürülmüş hali simgelenirken, *s* ile esas veri

simgelenmektedir (Silahtaroğlu, 2008:25). Bu dönüşüm bütün verilere uygulanarak dönüşüm gerçekleştirilir. Dönüştürülmüş veriler uygulamada kullanılır.

1.8.4.2 Sıfır-ortalama Normalleştirmesi (z-score normalization)

Bu normalleştirme, herhangi bir "A" değişkeninin ortalaması ve standart sapması kullanılarak yapılır. Normalleştirmeyle birlikte, değişkenlerin ortalama etrafında yayılması sağlanarak değişkenliklerden dolayı karşılaşılacak sorunlar en aza indirgenmiş olur (Han ve Kamber, 2006:96).

$$s' = \frac{s - ort}{\sigma} \tag{1.2}$$

Burada *s'* ile gösterilen değer, değişkenin normalleştirildikten sonra alacağı değeri, *s* ise esas değeri göstermektedir. Bunun yanında *ort*, verinin ortalamasını ve σ ise verinin standart sapmasını gösterir. Yukarıdaki formül bütün değişkenlere uygulanarak normalleştirme yapılmış olur.

1.8.4.3 Ondalık Derecesi ile Normalleştirme

"A" herhangi bir değişkeni simgelesin. Bu değişkenin mutlak değerce en büyük değerine bağlı olarak değişkenlerin ondalık sayılarının değişmesi ile elde edilen normalleştirmedir.

$$s' = \frac{s}{10^j} \tag{1.3}$$

Bu formülde *s'* normalleştirilmiş değerleri, *s* esas değerleri gösterir. Aynı zamanda *j* ise dönüştürülmüş değerlerin mutlak değerce en büyüğünü 1 den küçük yapan en küçük tam sayıdır (Han ve Kamber, 2006:96).

Örneğin "A" değişkeni -954 ile 934 arasında değişen değerlere sahip olsun. Burada mutlak değerce en büyük değer 954' dür. Bu değeri 1 den küçük yapan en küçük değer ise 1000 değeridir. Buna bağlı olarak *j=3* olarak belirlenmiş olur. -954 değeri formül uygulandıktan sonra -0.954' e dönüşmüş olur. Aynı şekilde diğer değerler için de formül kullanılarak normalleştirme yapılmış olur.

1.8.5 Veri İndirgeme

Veri indirgeme yöntemleri, esas verilerden, daha küçük veri kümeleri elde etmek için kullanılır. Bu indirgenmiş veri ile elde edilen sonuçlar, daha etkili olur. Böylelikle veri madenciliği çalışmasının güvenilirliği artarken zamandan da kazanılmış olur. Veri indirgeme yöntemleri aşağıda gösterilmiştir (Han ve Kamber, 2006:97).

a. Veri küpü birleştirme;

Veri küpü yapılarında birleştirme işlemlerinin uygulanmasını içeren tekniktir. Böylece çözümlemeler sadece belirlenen boyutlara göre yapılır (Özkan, 2008:41).

b. Boyut indirgeme;

Veri madenciliği amacıyla ilgili olmayan, az ilgili olan ya da gereksiz değişkenleri ve boyutları tespit edip uygulamadan çıkarmayı amaçlar. İndirgeme işlemi aynı zamanda, korelasyonu yüksek birden çok değişkeni birleştirerek tek bir değişkene dönüştürmeyi de amaçlamaktadır. Boyut indirgeme istatistiğe dayalı yöntemlerle yapılabilir. Temel bileşenler analizi, faktör analizi örnek olarak gösterilebilir.

c. Veri sıkıştırma;

Kodlama teknikleri ile veri setinin indirgendiği tekniklerdir. Eğer esas veri tekrar yapılandırıldığında bilgi kaybı olmuyorsa bu veri sıkıştırma tekniği *kayıpsız* 'dır denir. Sıkıştırma işlemi için iki önemli teknikten söz edilir. Bunlar, temel bileşenler analizi ve dalga dönüşümüdür.

d. Çokluk azaltımı;

Verilerin, küçük veri kümeleriyle tahminlemesini içeren yöntemdir. Bu tahminlemede parametrik modeller, parametrik olmayan modeller, kümeleme, örnekleme yöntemleri kullanılır. Böylelikle günlük verileri biriktirmek yerine model parametrelerini kaydetmek yeterli olur.

e. Ayrıklaştırma ve kavram hiyerarşisi üretme;

Ayırma teknikleri, sürekli özelliğe sahip değişken sayılarının indirgenmesi için kullanılır. Aralık etiketleri, günlük veri değerlerinin yerini alır. Bu ayırma özellikle karar ağaçları uygulamalarında kullanılır. Örneğin ücret değişkeninin 500 ile 10000

arasında değiştiğini varsayalım. Bu genişliği 500-1000, 1000-1500 gibi aralıklara ayırarak indirgemiş oluruz.

Kavram hiyerarşisi üreterek de veri indirgenir. Bu yöntem verileri derecelendirme işlemidir. Örneğin yaş değişkenini küçük-orta-büyük olarak derecelendirerek indirgemiş olur.

İKİNCİ BÖLÜM

VERİ MADENCİLİĞİ TEKNİKLERİ VE MODELLERİ

Veri madenciliğinde kullanılan modeller tahminleyici ve tanımlayıcı olmak üzere iki ana başlık altında toplanır. Tahminleyici modeller ileriye dönük tahminler geliştirmeyi hedeflerken tanımlayıcı modeller mevcut durumu değerlendirmeyi ve bundan sonuçlar çıkarmayı hedeflemektedir. Veri madenciliği modelleri gördükleri işleve göre üç ana başlık altında toplanır. Bunlar;

a. Sınıflama ve Regresyon
b. Kümeleme
c. Birliktelik kuralları (Akpınar, 2000:3)

Bu modeller arasında sınıflama ve regresyon tahminleyici, kümeleme ve birliktelik kuralları tanımlayıcı modellerdir.

Model kuruluş aşaması öğrenimin denetimli ve denetimsiz olmasına göre farklılık gösterir. Örnekten öğrenme olarak da bilinen denetimli öğrenme, bir kullanıcı tarafından hedef sınıflar önceden belirlenen bir ölçüte göre ayrılarak her sınıf için çeşitli örnekler verilir. Sistemin amacı verilen örneklerden hareket ederek her bir sınıfa ilişkin özelliklerin bulunması ve bu özelliklerin kural cümleleri ile ifade edilmesidir (Akpınar, 2000:6). Denetimsiz öğrenmede ise tanımlanmış herhangi bir hedef sınıf yoktur (Larose, 2005:91).

Denetimli öğrenmede öncelikle verilerin bir kısmı modelin öğrenimi için diğer bir kısmı ise modelin geçerliliğini test etmek için kullanılır. Basit geçerlilik yöntemi verinin bölünmesi için kullanılan basit bir yöntemdir. Bu yöntemde verilerin %5 ile %33 arasındaki bir kısmı test verisi olarak ayrılır. Geri kalan kısmı ise öğrenme verileridir. Bir diğer yöntem ise çapraz geçerlilik yöntemidir Bu yöntemde ise veri kümesi rastgele iki eşit parçaya bölünür. İlk başta ilk parça test diğer parça öğrenim sonra ikinci parça test ilk parça öğrenim verileri olarak kullanılır. Elde edilen hata oranlarının ortalaması modelin tahmini hata oranı olur.(Akpınar, 2000:6).

2.1 Sınıflandırma Teknikleri

Sınıflandırma teknikleri, veri madenciliğinin en çok kullanılan teknikleridir. Kullanıldığı yerlere örnek olarak;

a. Bankacılık: Kredi uygulamalarında müşterinin kötü ya da iyi kredi risk puanına sahip olup olmadığına karar vermede,

b. Eğitim: Yeni öğrencileri özelliklerine göre sınıflara yerleştirmede ve buna göre eğitim programları hazırlamada,

c. Sağlık: Özel bir hastalığın olup olmadığını teşhis etmede,

d. Hukuk: Bir vasiyetin kimin tarafından yazılmış olabileceğine karar vermede kullanıldığı verilir (Larose, 2005:95).

Sınıflandırmada, önceden sınıflara ayrılmış bir hedef kategorik değişken bulunur. Gelir düzeyi ele alınırsa, orta, düşük, yüksek gelir olarak sınıflandırılması örnek olarak gösterilebilir. Veri madenciliği modeli geniş veri kümelerini sorgular ve veri setindeki her kayıt hedef değişken üzerinde bilgi içerir (Larose, 2005:95).

Diğer bir tanım olarak sınıflandırma, çeşitli içeriklere sahip değişkenleri sınıflara ayırma işlemidir. Bu sınıflar, iş kuralları, sınıf sınırları ve bazı matematiksel fonksiyonlar tarafından tanımlanır. Sınıflandırma işlemi, bilinen bir sınıf ataması ve sınıflanacak değişkenin özelliği arasındaki ilişkiye dayanır. Bu tip sınıflandırmaya denetimli sınıflandırma denir (Nisbet ve diğerleri, 2009:235).

Sınıflandırma teknikleri genel olarak beş bölüme ayrılır.

- Karar ağaçları ile sınıflandırma
- Sınıflandırma ve regresyon ağaçları (CART)
- Bellek tabanlı sınıflandırma: En yakın k-komşu algoritması
- İstatistiğe dayalı sınıflandırma
- Yapay sinir ağları

2.1.1 Karar Ağaçları ile Sınıflandırma

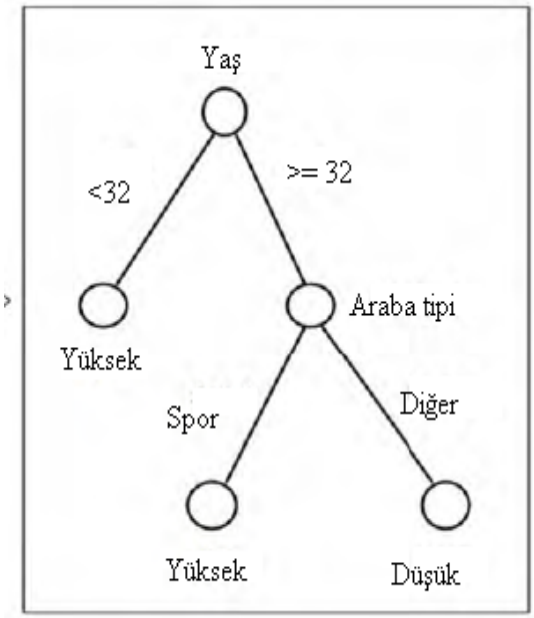
Karar ağaçları, sınıflandırmada en yaygın kullanılan yöntemlerden biridir. Bu yöntem, tahmin edici değişkenleri kullanarak model kurulması ve bu model sayesinde yeni değişkenlerin üyelik derecelerine göre farklı sınıflara ayrılması işlemidir.

Sınıflandırma görselliği açısından kullanışı caziptir. Karar ağaçları, istatistiksel bakış ile örüntü tanıma alanında çok yaygın olmamasına rağmen, hastalık tanısında, bilgisayar bilimlerinde (veri yapıları), psikolojide (davranışsal karar teorisi), yaygın olarak kullanılır. Kalp damarlarında tıkanma tanısında karar ağacı oluşturulması örnek olarak gösterilebilir. Kalp krizi geçiren hastalara birçok test uygulanır. Bunlar, kalp atış oranı, kan basıncı, EKG (Electrocardiogram). Bütün bu testler hasta hakkında bilgi veren verilerdir. Bunlardan yola çıkarak oluşturulacak karar ağacı ile başka hangi hastanın risk grubunda olabileceği tahminlenebilmektedir (Gorunescu, 2011:160).

Şekil 6' da örnek bir karar ağacı gösterilmektedir.

Şekil 6: Örnek Karar Ağacı

Yaş	Araba tipi	Kaza riski
20	Spor	Yüksek
18	Spor	Yüksek
40	Minibüs	Düşük
50	Lüks	Düşük
35	Aile tipi	Düşük
30	Spor	Yüksek
32	Spor	Yüksek
40	Otobüs	Düşük
33	Mini	Yüksek
39	Üstü açık	Düşük



Kaynak: Gorunescu, 2011, s.160

Şekil 6' da araba tiplerine ve yaşlarına göre kaza riskleri karar ağacı ile sınıflandırılmıştır. Sonuç olarak 32 yaşından küçük arabaların kaza risklerinin yüksek, 32 yaşından büyük veya eşit olan arabalardan spor tipi arabaların kaza riski yüksek diğerlerinin düşük olduğu görülmektedir.

Karar ağaçları, üç kısımdan oluşmaktadır. Bunlar; kök, dallar ve yapraklardır. Şekil 6 'da gösterilen karar ağacında, yaş ve araba tipi ile gösterilen düğümler birer kök düğüm, 32 yaşından büyük veya küçük olduğunu gösteren ve kök ile yaprakları

bağlayan yapı dal, kaza riskinin yüksek veya düşük olduğunu gösteren düğümler ise yaprak olarak adlandırılır.

Karar ağaçlarını oluşturmadan önce;

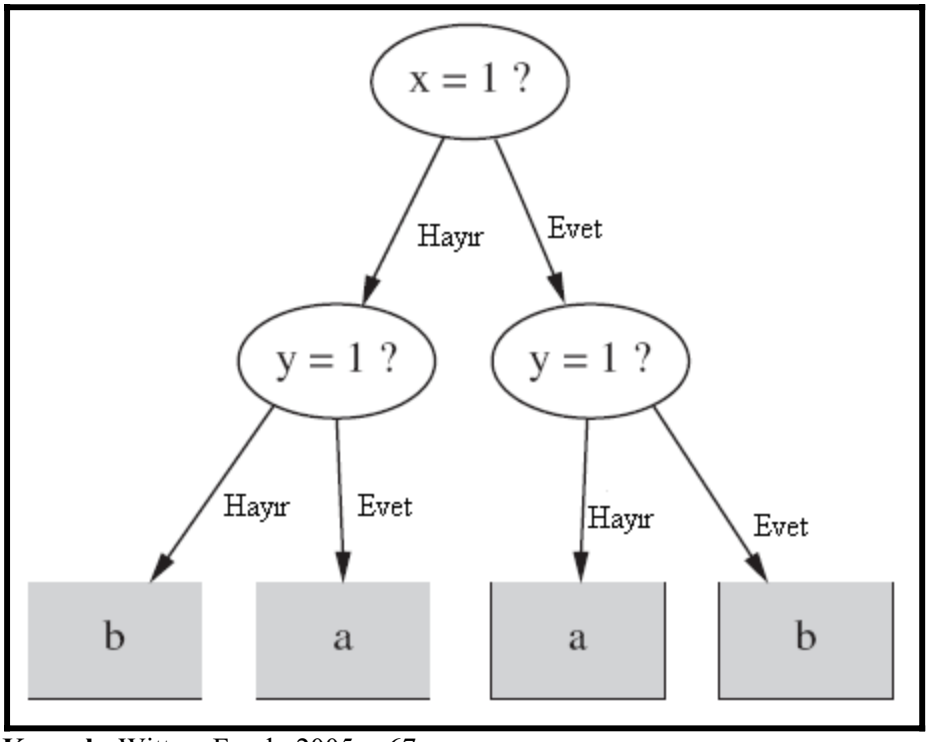
a. Karar ağacı algoritmaları, denetimli öğrenme ile uygulanır. Eğitim kümesi, hedef değişkeni destekler nitelikte olmalı,

b. Eğitim kümesi, hedef değişkene uygun zengin ve çeşitli değişkenler barındırmalı,

c. Hedef değişken sınıfı kesikli değişkenlerden oluşmalıdır (Larose, 2005:109).

Eğitim verileri ile oluşturulan karar ağacına, test kümesinin her bir kaydı uygulanarak modelin geçerliliği test edilir. Oluşturulan ağaç aslında birçok "*eğer (if then)"* den oluşur (Silahtaroğlu, 2008:48). Bu *eğer* 'lere verilen *evet/hayır* cevaplarıyla sınıflandırma yapılmış olur. Şekil 7' de örnek bir karar ağacı ile gösterilmiştir.

Şekil 7: Karar Ağaçlarında *Eğer (if then)* Örneği



Kaynak: Witten, Frank, 2005, s.67

Şekil 7 ‘ de gösterilen karar ağacında sınıflandırma;

- *if* x=1 ve y=0 *then* sınıf=a, (Eğer x=1 ve y=0 iken a sınıfında)
- *if* x= 0 ve y=1 *then* sınıf=a, (Eğer x=0 ve y=1 iken a sınıfında)
- *if* x=0 ve y=0 *then* sınıf=b, (Eğer x=0 ve y=0 iken b sınıfında)
- *if* x=1 ve y=1 *then* sınıf=b (Eğer x=1 ve y=1 iken b sınıfında)

sorgularına dayanarak elde edilir. (Witten ve Frank, 2005:67)

Karar ağaçlarına dayalı olarak birçok algoritma geliştirilmiştir. Bunlar arasında en yaygın olarak kullanılanları ID3 ve C4.5, CART, SPRINT ve SLIQ algoritmalarıdır. Bu algoritmalar Tablo 5’ de gösterilen kaba kod çerçevesinde çalışır.

Tablo 5: Karar Ağacı Algoritmalarının Akış Şeması

D: *Öğrenme kümesi*
T: *Kurulacak ağaç*
T=0 // *Başlangıçta ağaç boş küme*
Dallara ayırma ölçütlerini belirle
T= Kök düğümü belirle
T= Dallara ayırma kurallarına göre kök düğümü dallara ayır;
Her bir dal için
do
Bu düğüme gelecek değişkeni belirle
if (durma koşuluna ulaşıldı)
yaprak ekle ve dur
else
loop

Kaynak: Silahtaroğlu, 2008, s.50

2.1.1.1 ID3 Algoritması

ID3 algoritması 1986 yılında Quinlan tarafından geliştirilen basit bir karar ağacı algoritmasıdır. Bölünme ölçütü olarak bilgi kazanımını kullanır. ID3 algoritması, bütün durumlar hedef değişkenin tek bir değerine ait olduğunda ya da en iyi bilgi kazanç ölçütü sıfırdan büyük olmadığı durumda karar ağacı gelişimini durdurur (Rokach ve Maimon, 2008:71). ID3 algoritması kategorik değişkenler için kullanılır.

ID3 algoritması entropiye dayalı bir algoritmadır. Karar ağacında dallanmanın hangi niteliğe göre yapılacağı kazanç ölçütüne göre belirlenir. Kazanç ölçütünün tespitinde entropi kavramı kullanılır.

Entropi, bilgi kazanımını en çoklamaya dayalı bölümleme için en uygun değerin seçimidir. Bundan dolayı bu yönteme dayanarak seçilen bölümleme noktası, sınıflandırma için gerekli bilginin maksimum olduğu noktadır. Bu yüzden eğer bütün değerler aynı sınıfa ait olursa entropi değeri sıfıra eşit olur (Gorunescu, 2011:169).

Başka bir tanım olarak entropi, belirsizliğin ölçüsüdür. Örnek olarak S' nin bir kaynak olduğu varsayılısın. Bu kaynağın $\{m_1, m_2..m_n\}$ gibi n mesaj ürettiği düşünülsün. Tüm mesajlar birbirinden bağımsızdır ve m_i mesajlarının üretilme olasılıkları p_i 'dır. $P = \{p_1, p_{2,} p_3...p_n\}$ olasılık dağılımına sahip mesajlar üreten S kaynağının entropisi H(S);

$$H(S) = -\sum_{i=1}^{n} p_i \log_2(p_i) \qquad (2.1)$$

formülüyle bulunur (Özkan, 2008:55).

Entropi formülü yardımı ile kazanç ölçütü hesaplanarak dallanma için nitelikler seçilir. Nitelik seçimi için P sınıfına ait p değerden ve N kümesine ait n değerden oluşan bir C veri seti ele alınsın. Eğer karar ağacının kökü için $\{A_1, A_2..A_V\}$ değerlerine sahip A değişkeni kullanılacak ise bu, C veri setini $\{C_1, C_2...C_V\}$ şeklinde parçalara bölecektir. Bu C_i değerleri P sınıfına ait p_i değerden ve N sınıfına ait n_i değerden oluşur. C_i alt ağacı için beklenen bilgi ihtiyacı $H(p_i, n_i)$ ' ye eşittir. Kök olarak seçilecek A değişkeni için beklenen bilgi ihtiyacı;

$$E(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p + n} H(p_i, n_i) \quad (2.2)$$

formülüyle hesaplanır (Quinlan, 1986:90). Bunun yardımı ile kazanç ölçütü;

$$\text{kazanç}(A) = H(p, n) - E(A) \quad (2.3)$$

formülü ile hesaplanır (Quinlan, 1986:90).

Kazanç ölçütleri her bir değişken için tespit edildikten sonra en yüksek kazanç ölçütüne sahip değişken karar ağacının ilk kökünü oluşturur. Belirlenen kök değişkene ait sınıf değerlerde bu kök değişkenin dallarını oluşturur. Bu işlemler her bir sınıf, hedef değişkene ait olana kadar sürdürülür. Bu şekilde karar ağacı sonlandırılmış olur.

2.1.1.2 C4.5 Algoritması

C4.5 algoritması, karar ağacı oluşturmak için yaygın kullanılan bir algoritmadır. Bu algoritmanın işleyişi temel olarak ID3 algoritması gibidir. ID3 algoritmasından farklı olarak kesikli değişkenlerin yanında sürekli değişkenleri de kullanır. Yani C4.5 algoritması yardımıyla sayısal değişenlerde algoritmada kullanılabilir. Sayısal değişkenler gruplandırılarak işleme sokulur (Berry ve Browne, 2006:82).

C4.5 algoritması, diğer verilerden öngörerek kayıp değerleri de kullanır. Böylelikle daha anlamlı ve daha duyarlı kurallar elde edilebilen bir ağaç üretilir (Silahtaroğlu, 2008:56). Bu değerleri kullanabilmek için düzeltilmiş bir kazanç ölçütüne ihtiyaç vardır. Düzeltilmiş kazanç ölçütü;

$$\text{kazanç}(A) = F(H(p, n) - E(A)) \quad (2.4)$$

formülüyle hesaplanır. Burada "F", düzeltme faktörüdür (Özkan, 2008:81).

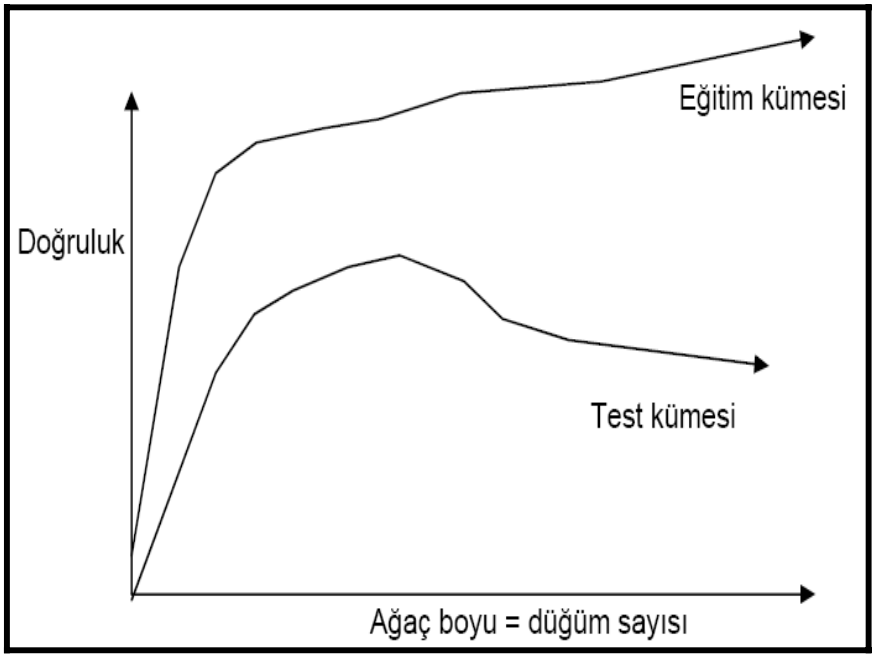
$$F = \frac{\text{Veri tabanında değeri bilinen örneklerin sayısı}}{\text{Veri tabanındaki tüm örneklerin sayısı}} \quad (2.5)$$

Bir veya birden çok alt ağacı ortadan kaldırarak onların yerine uygun yapraklar koyma işlemiyle karar ağaçlarının basitleştirilmesine karar ağacını budama denir. C4.5 algoritması budama işlemine olanak sağlamaktadır. Bir alt ağacın yerine yaprak koyma işleminde algoritma, tahmini hata oranını azaltmak ve sınıflandırma modelinin kalitesini arttırmayı hedefler. Ancak hata oranını hesaplamak kolay

değildir. Yalnızca eğitim verilerine dayanan hata oranı uygun bir tahmin sağlamaz. Tahmini hata oranını öngörmek için, ilave test örneklerinden yeni bir küme oluşturulur. Bu teknik, ilk olarak önceden var olan örnekleri eşit aralıklı bloklara ayırır. Her blok için bu bloklardan beklenen bütün örneklemlerle karar ağacı oluşturulur ve verilen örneklem blokları ile test edilir. Uygun test ve eğitim örneklemleri ile karar ağacı budama işleminin temel fikri ortaya çıkar. Bu temel fikir, gizli test örneklemlerinin sınıflandırma doğruluğunda katkısı olmayanları ağacın parçalarından (alt ağaçlardan) çıkarmaktır. Böylelikle daha az karmaşık ve daha anlaşılır bir ağaç elde edilmiş olur (Kantardzic, 2011:184).

Basit bir veri yığınından oluşturulan karar ağacının çok büyük çıkmasına şişme (overfitting) denir. Ağaç oluşturma algoritmaları her zaman şişme oluştururlar. Oluşan ağacın çok büyük olması bu etkiyi artırır. Ağacın dengeli olabilmesi için belli bir büyüklüğün üstünde olmalıdır. Bu büyüklü arttıkça test verisinin hata oranı yükselmekte ve ağacın doğruluğu azalmaktadır. Bu durumda ağaç budama işlemi kullanılır. Eğitim kümesi ile test kümesinin ağaç boyuna göre doğruluk performansı Şekil 8' de gösterilmiştir (Yıldırım, 2003:31).

Şekil 8: Karar Ağacı Boyuna Göre Doğruluk Performansı



Kaynak: Yıldırım, 2003, s.31

İki çeşit budama yöntemi vardır bunlar;

- Ön budama
- Sonradan budama

Bazı durumlarda örneklem kümesini daha fazla bölmemek kararı alınır. Bölme işlemine son verme ölçütü olarak ki-kare gibi istatistiksel testler uygulanır. Bölünme öncesine ve sonrasında önemli bir fark yoksa o zaman söz konusu düğüm bir yaprak olarak gösterilir. Bu ön budama çeşididir (Özkan, 2008:83).

Seçilen bir doğruluk ölçütü kullanarak bazı ağaçlar budanabilir. Bu yöntem ağaç oluşturulduktan sonra uygulanır. Bundan dolayı bu yönteme sonradan budama denir. C4.5 algoritmasında bu budama yöntemi kullanılır (Özkan, 2008:83). Bu yöntem, kötümser budama olarak da adlandırılır. Örneğin, "T" , "S" eğitim kümesinden üretilmiş yapraksız bir karar ağacı olsun. T_i^* ise karar ağacının budanmışlığını simgelesin. Ek olarak , T_f^* "B" düğümünün en sık gözlenen alt ağacını ve "L" ise "S" kümesinin en sık gözlenenleri ile sınıflanmış bir yaprağını ifade etsin. Sırasıyla E_T, $E_{T_f^*}$ ve E_L ise "S" sınıfı içinde, T, T_f^* ve L tarafından sınıflandırılmamış durumların sayılarını göstersin. Bunlara göre üç çeşit hata oranı tahminlenebilir (Kohavi ve Quinlan, 1999:8). Bunlar;

- $U_{CF}(E_T, |S|)$
- $U_{CF}(E_L, |S|)$
- $U_{CF}(E_{T_f^*}, |S|)$

Burada "U_{CF}" ile istatistiksel tablolar kullanılarak hesaplanan binomial dağılımı göstermektedir. "CF" ise güven düzeyini gösterir. C4.5 algoritması genelde % 25' lik güven düzeyini kullanır (Kantardzic, 2011:184). Bu hata oranlarına göre alt ağaç, kök düğüm haline getirilerek budama işlemi tamamlanmış olur.

2.1.1.3 CART Algoritması

CART, sınıflama ve regresyon ağaçları 1984 yılında Breiman tarafından ortaya atılmıştır (Larose, 2005:109). Bu algoritma C4.5 algoritması ile aynı temelde karar ağacı kurarak karar üretir. CART algoritması da C4.5 gibi en uygun değeri

seçme prosedürünü kullanır. C4.5 'in tersine CART algoritması, sadece ikili ağaç yapımına olanak sağlar (Berry ve Browne, 2006:85). İkili ağaç, her düğümün iki dala ayrıldığı ağaç çeşididir. Bölünme işlemi, twoing ve gini algoritmalarıyla yapılır. CART algoritmasının en önemli özelliği, regresyon ağacı oluşturmasıdır. Regresyon ağacında, ağacın yaprakları bir sınıfı tahminlemez, gerçek sayıları tahminler (Maimon, Rokach, 2005:181). CART algoritmasında her düğüm, mümkün bütün bölünmelerle karşılaştırılır ve homojenlik derecesi en yüksek olan özellik seçilir. Budama işleminde, C4.5 algoritması binomial güven sınırlarını kullanırken CART, en az maliyetli karmaşıklık budama yöntemini kullanır. Bu yaklaşım, tekrar yerine koyma hatası (re-substitution error) yanlılığının, karar ağacı yaprak sayısını doğrusal olarak arttırdığını varsayar. Bir alt ağaca yüklenen maliyet iki terimden oluşur; yerine koyma hatası (re-substitution error) ve karmaşıklığın ölçüsünü gösteren α parametresinin yapraklardaki sayısı. (Kantardzic, 2011:190).

CART algoritmasında iki tip bölünme algoritması vardır; Twoing ve gini. Bölünme için twoing algoritması seçilirse algoritma şu şekilde çalışır (Özkan, 2008:89);

Adım 1

a. Niteliklerin içerdiği değerler göz önüne alınarak eğitim kümesi iki dala ayrılır. Bunlara aday bölünme denir. Bir "t" düğümünde "sağ" ve "sol" olmak üzere iki ayrı dal bulunur. Bu bölümlenen kümeler $t_{sağ}$ ve t_{sol} biçimindedir.

b. Aday bölünmelerin her biri için P_{sol} ve $P(j/t_{sol})$ olasılıkları hesaplanır. Burada $P(j/t_{sol})$ ifadesi bir j sınıf değerinin sol tarafta olma olasılığını verir. Söz konusu olasılıklar şu şekildedir;

$$P_{sol} = \frac{t_{sol}\text{'daki herbir niteliğin i lg ili nitelik sütunundaki tekrar sayısı}}{\text{Eğitim küme sin deki kayıtların sayısı}} \qquad (2.6)$$

$$P(j/t_{sol}) = \frac{t_{sol}\text{'daki kayıtların j sınıfları sayısı}}{t_{sol}\text{'daki herbir niteliğin i lg ili nitelisütunundaki tekrar sayısı}} \qquad (2.7)$$

c. Aday bölünmelerin her biri için formül 2.6 ve 2.7, sağ dal için de kullanılarak $P_{sağ}$ olasılıkları hesaplanır. Formüllerde, "sağ" dalın olasılıkları hesaplanmak istenildiğinden t_{sol} yerine $t_{sağ}$ değeri kullanılır.

d. $\phi(s/t)$, "t" düğümündeki "s" aday bölünmelerinin uygunluk ölçüsü olarak kabul edilir. Uygunluk ölçüsü;

$$\phi(s/t) = 2P_{sol}P_{sağ}\sum_{j=1}^{n}\left|P(j/t_{sol}) - P(j/t_{sağ})\right| \tag{2.8}$$

formülüyle hesaplanır.

e. $\phi(s/t)$ değerleri hesaplandıktan sonra içlerinde en büyük olanı seçilir. Bu değerin ilgili olduğu aday bölünme satırı, dallanmanın yapılacağı satırı bildirecektir.

f. Dallanma bu şekilde yapıldıktan sonra bu adıma ilişkin karar ağacı çizilir.

Adım 2

Birinci adıma dönülerek alt ağaçlara da aynı işlemler uygulanır.

Bölünme için Gini algoritması kullanılırsa algoritma aşağıdaki gibi çalışır (Özkan, 2008:106);

Adım 1

Her nitelik değeri ikili olacak şekilde gruplanır. Bu şekilde elde edilen sağ ve sol bölünmelere karşılık gelen sınıf değerleri gruplandırılır.

Adım 2

Her bir nitelikle ilgili sol ve sağ taraftaki bölünmeler için $Gini_{sağ}$ ve $Gini_{sol}$ değerleri hesaplanır. Bu hesaplamalar;

$$Gini_{sol} = 1 - \sum_{i=1}^{k}\left(\frac{L_i}{|T_{sol}|}\right)^2 \tag{2.9}$$

$$Gini_{sağ} = 1 - \sum_{i=1}^{k}\left(\frac{R_i}{|T_{sağ}|}\right)^2 \tag{2.10}$$

Formülleriyle hesaplanır. Burada;

k: Sınıfların sayısı

T: Bir düğümdeki örnekler

T_{sol} : Sol taraftaki örneklerin sayısı

$T_{sağ}$: Sağ taraftaki örneklerin sayısı

L_i : Sol tarafta *i* kategorisindeki örneklerin sayısı

R_i : Sağ tarafta *i* kategorisindeki örneklerin sayısı

Adım 3

Her *j* niteliği için *n* eğitim kümesindeki satır sayısı olmak üzere formül 2.11' deki bağıntı hesaplanır.

$$Gini_j = \frac{1}{n}(|T_{sol}|Gini_{sol} + |T_{sağ}|Gini_{sağ}) \quad (2.11)$$

Adım 4

Her *j* niteliği için hesaplanan $Gini_j$ değerleri arasından en küçük olanı seçilir ve bölünme bu nitelik üzerinden gerçekleşir.

Adım 5

İlk adıma dönülerek işlemler sürdürülür.

2.1.1.4 SPRINT Algoritması

Karar ağacı tabanlı ilk paralel sınıflama algoritmalarından birisidir. Bu ölçeklenebilir algoritma çok sayıda işlemcinin paralel olarak bir karar ağacını beraber üretmesine dayanır. Algoritma ayrıca disk tabanlı sınıflama için tasarlanmıştır (Özdoğan ve diğerleri , 2006:3).

SPRINT algoritmasında girdilerin boyutu için herhangi bir sınırlama bulunmaz. Sonuca hızlı ulaşabilen bir algoritmadır. SLIQ algoritması ile benzerlik gösterse de farklı veri yapıları kullanmaları bakımından ayrılırlar. SPRINT, ilk olarak Tablo 6' da gösterilen veriler gibi bir veri tablosu oluşturur. Bütün bu veri kayıtları ile veri değerleri, bir sınıf etiketi ve bu değerlerle elde edilen kayıt endeksi oluşur. Bu kayıt endeksi "*rid"* olarak gösterilir. İlk tablolar sürekli değişkenlere göre sıralanır. Eğitim verilerinden elde edilen ilk tablolar, karar ağacının kök düğümüyle ilişkilendirilir. Ağaç bu şekilde gelişir ve bölünmelerle yeni ürünler oluşur ve her düğüme ait değişken tabloları bölünür ve bu ürünlerle ilişkilendirilir. Bir tablo bölündüğünde kayıtların sırası korunur. Bundan dolayı, tekrar sıralama yapmaya gerek kalmaz (Shafer ve diğerleri, 1996:546).

Tablo 6: SPRINT İçin Veri Listesi

Yaş	Sınıf	Kayıt endeksi
17	Yüksek	1
20	Yüksek	5
23	Yüksek	0
32	Düşük	4
43	Yüksek	2
68	Düşük	3

Araba çeşidi	Sınıf	Kayıt endeksi
Aile	Yüksek	0
Spor	Yüksek	1
Spor	Yüksek	2
Aile	Düşük	3
Kamyon	Düşük	4
Aile	Yüksek	5

Kaynak: Shafer, Agrawal, Mehta, 1996, s.546

Ayırma şartları altında sürekli değişkenler için her karar ağacı düğümü iki histogramla ilişkilendirilir. Bu histogramlar, C_{ALT} ve $C_{ÜST}$ gösterilir ve bu histogramlar kayıtların sınıf dağılımlarını belirler. Kategorik değişkenler, bir düğümle ilişkili olarak histogramı çıkarılır. Bu tek histogram, her değişkenin sınıf dağılımını gösterir. Bu histograma sayı matrisi adı verilir (Shafer ve diğerleri, 1996:546). Algoritma, bölünme ölçütü olarak gini ölçütünü kullanır.

2.1.1.5 SLIQ Algoritması

SLIQ algoritması kategorik ve sürekli değerler için kullanılabilir. SPRINT algoritmasına benzer şekilde, belleğe uygunluk açısından çok geniş disk yerleşkeli verilere önceden sıralama tekniğini kullanarak işlem yapar. SLIQ algoritması, disk yerleşkeli değişken tablosunu ve tek bellek yerleşkeli sınıf tablosunu kullanır. Sınıf tablosu, ağaç oluşturulana ve budanana kadar hafızada aynen kalır. Sınıf tablosu belleğe uygun olmadığı zaman SLIQ algoritmasının performansı azalır (Han ve Kamber, 2006:232).

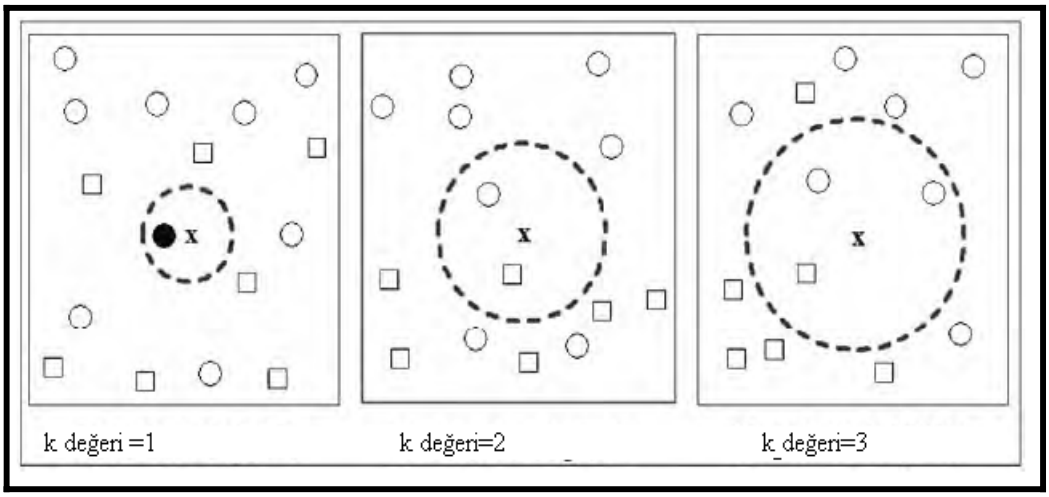
ID3 ve C4.5 gibi algoritmalar "önce derinlik" ilkesine göre çalışırken SLIQ algoritması "önce genişlik " ilkesine göre çalışır (Silahtaroğlu, 2008:58).

SLIQ algoritması, örnekleme veya bölünme yoluyla oluşturulmuş küçük bellek genişliğindeki veri kümelerini kullanmaz. Ancak bütün eğitim kümesini kullanarak karar ağacı oluşturur. Bu yüzden bellek yerleşkesindeki her kayıtlı veriye her zaman ihtiyaç duyar. Bellekteki veri yapısının genişliği giriş kayıtlarıyla orantılı olarak gelişirken, bu, SLIQ tarafından sınıflandırılan veri miktarını sınırlar. Bölünme ölçütü olarak gini ölçütünü kullanır (Shafer ve diğerleri, 1996:545).

2.1.2 Bellek Tabanlı Sınıflandırma: En Yakın k-komşu Algoritması

En yakın k-komşu algoritması sınıflandırmada çok sık kullanılan bir tekniktir. Öncelikle, eğitim veri kümesi oluşturulur. Sınıflanmamış yeni bir kayıtın sınıfı, eğitim kümesindeki en çok benzeyen kayıtlarla karşılaştırılır. Böylelikle yeni kayıt sınıflandırılmış olur (Larose, 2005:96). Bu benzerliklerin bulunması, en küçük uzaklığa sahip *k* adet gözlemin seçilmesi esasına dayanmaktadır (Özkan, 2008:117). Bu algoritmada *k* değeri uygulayıcı tarafından belirlenir. Bu değerin çok büyük olması, sınıflandırılmak istenen yeni değişkeni, benzerlikleri az olan bir sınıfa atarken, bu değerin çok küçük olması ise aynı sınıfın noktaları oldukları halde bazı noktaların ayrı sınıflara konmasına ya da bu tür noktalar için ayrı sınıf açılmasına neden olmaktadır (Silahtaroğlu, 2008:65). Şekil 9 'da bazı *k* değerleri için sınıflandırma örnekleri gösterilmiştir.

Şekil 9: Bazı *k* Değerleri İçin Örnekler



Kaynak: Gorunescu, 2011, s.257

Şekil 9' da, *k* değeri büyüdükçe sınıf dairesinin de büyüdüğü görülmektedir. Benzerliklerin hesaplanması için uzaklık formüllerinden faydalanılır. En çok kullanılan uzaklık, Öklid uzaklık formülüdür. Bu formül;

$$d(i,j) = \sqrt{\sum_{i=1}^{p} (x_i - y_i)^2} \qquad (2.12)$$

şeklindedir. Bu formülde $x = x_1, x_2 ... x_m$ ve $y = y_1, y_2 ... y_m$ şeklinde m değerden oluşan iki kayıt kümesidir (Larose, 2005:99).

Bunlara göre algoritmanın işleyişi şu şekilde olur;

1. *k* parametresi belirlenir.
2. Sınıflandırılacak olan yeni kayıtın mevcut kayıtlardaki tüm noktalarla olan uzaklıkları hesaplanır. En çok kullanılan uzaklık formülü Öklid uzaklık formülüdür.
3. Bu uzaklılara göre en küçük *k* tanesi seçilir.
4. Bu seçilen satırların hangi kategorilere ait oldukları belirlenir. En çok tekrarlanan kategori seçilir.
5. Belirlenen bu kategori yeni değişkenin sınıfı olarak atanır.

2.1.3 Bayesyen Sınıflandırma

Bayesyen sınıflandırma, istatistiksel sınıflandırma yöntemlerindendir. Bu yöntem, üyelik olasılıklarının sınıfını tahminlemek için kullanılır. Bayesyen sınıflandırma yöntemi Bayes teoremine dayanır (Han ve Kamber, 2006:310). Bayesyen sınıflandırma yöntemi, sağlık, genetik ve sanayi gibi birçok farklı alanda kullanılır. Sınıflandırma sonuçlarının basit yorumlanabilmesinden dolayı kullanım alanı geniştir. Ancak veri tiplerinin çok basit ve saf olmasından dolayı birçok dezavantajı da vardır. Bayesyen sınıflandırma, belirgin bir kural bulma yerine olasılıkları tahmin eden bir öğrenme sürecidir. Bu yaklaşım avantajı, belli olasılıkları tahminleme yöntemi tutarlı olduğunda ve veri kümesi büyük olduğunda bayes sınıflandırıcısı en küçük hataya ulaşacaktır (Berry ve Browne, 2006:17). Bayes teorisi bir koşullu olasılıkları hesaplayan istatistiksel bir teoridir. Bu koşullu olasılık hesabı;

$$P(A/B) = \frac{P(A \cap B)}{P(B)} \qquad (2.13)$$

şeklinde hesaplanır. Burada A ve B ayrık olmayan iki olaydır. $P(A/B)$, B olayı olma koşulu altında, A olayının gerçekleşmesi olasılığıdır. $P(A \cap B)$, A ve B olaylarının kesişimlerinin olasılığını ve $P(B)$, B olayının gerçekleşme olasılığını ifade etmektedir.

Bayes teorisi, bir olayın ilk gerçekleşme kanısını derecelendirmesiyle başlar ve yeni bilgi ile bu kanının derecelendirmesi güncellenir. Bu iki derecelendirme, sırasıyla önsel olasılık $P(A/B)$ ve sonrasal olasılık $P(B/A)$ (Berry ve Browne, 2006:20). Buna göre Bayes teorisini ;

$$P(A/B) = \frac{P(A)P(B/A)}{P(B)} \tag{2.14}$$

şeklinde göstermek mümkündür.

Bayesyen sınıflandırma yönteminde öncelikle eğitim verileri düzenlenip uygun hale getirilir. Düzenleme işlemine, sayısal verileri aralıklandırma örnek olarak verilebilir. Bu işlemden sonra 2.14' deki formüle dayanarak, P(A); her sınıfın verilen öğrenme kümesi içinde bulunma sıklığı, P(B); sınıflanmamış verilerin, eğitim kümesinde bulunma sıklığı, P(B/A); Sınıflanmamış verilerin A' da bulunma sıklığı hesaplanarak sınıflanmamış veriler sınıflandırılmış olur. Burada $A = x_1, x_2 ... x_i$ -i boyutlu bir vektörü ve $B = C_1, C_2 ... C_J$ j tane sınıfı göstermektedir (Silahtaroğlu, 2008:61).

- Sade bayesyen sınıflandırma;

Sade bayesyen sınıflandırma, büyük boyuttaki veri kümelerinde sınıflandırma yaparken kolaylık sağlamaktadır. Bayes teorisine dayanır. $S = \{S_1, S_2 ... S_m\}$, m örnekten oluşan bir eğitim kümesi olsun. Her S_i örneklemi, $\{x_1, x_2 ... x_n\}$ 'den oluşan n boyutlu bir vektör olduğu kabul edilsin. Bu x_i' ler sırasıyla $A_1, A_2 ... A_n$ özellikleriyle ilişkilendirilsin. Her örneklem, $C_1, C_2 ... C_k$ gibi k tane sınıfa ait olsun. Buna göre bilinmeyen bir X veri örneği verildiğinde, bu verinin sınıfı, en yüksek $P(C_i / X)$ olasılığı kullanılarak tahminlenebilir. Bu, sade bayesyen sınıflandırmanın

temelidir (Kantardzic, 2011:147). Bu olasılıklar bayes teorisi kullanılarak hesaplanır. Verilenlere göre bayes teorisi yazılacak olursa;

$$P(C_i / X) = \frac{[P(X / C_i).P(C_i)]}{P(X)} \tag{2.15}$$

şeklindedir. Burada P(X) her sınıf için aynı olduğundan yalnızca $P(X / C_i).P(C_i)$ olasılıklar çarpımının maksimum yapılmaya ihtiyacı vardır. Önsel olasılık formül 2.16' daki gibi hesaplanır.

$$P(C_i) = C_i \text{ sınıfı eğitim örneklemlerinin sayısı} / m \tag{2.16}$$

$P(X / C_i)$ olasılığını hesaplamak özellikle geniş veri kümelerinde oldukça karmaşıktır. Bunun için naive bayes yaklaşımı kullanılır. Bu yaklaşıma dayanarak $P(X / C_i)$ olasılığı;

$$P(X / C_i) = \prod_{t=1}^{n} P(x_t / C_i) \tag{2.17}$$

şeklinde hesaplanır (Kantardzic, 2011:147). Formül 2.17 yardımıyla bulunan olasılıklar arasından en büyük olanı seçilerek yeni değişkenin sınıfı bulunmuş olur.

2.1.4 Yapay Sinir Ağları

Yapay sinir ağları, biyolojik sinir ağlarının genel özelliklerine sahip bir bilgi işleme sistemidir. Bu ağlar, insan algısının veya sinir sisteminin matematik modelinin genelleştirilmesi ile geliştirilir (Fausett, 1994:3). Yapay sinir ağları dört yaklaşıma dayanır;

1. Bilgi işleme, nöron olarak adlandırılan basit parçalarda oluşur.
2. Sinyaller, iletim bağlantıları üzerinden nöronlar arasında gider gelir.
3. Her iletim bağlantısı, birleşmiş bir ağırlığa sahiptir.
4. Her nöron, ağa gelen bir girdinin çıktısına karar verebilmek için bir aktivasyon fonksiyonu uygular ve bu fonksiyon genelde doğrusal değildir.

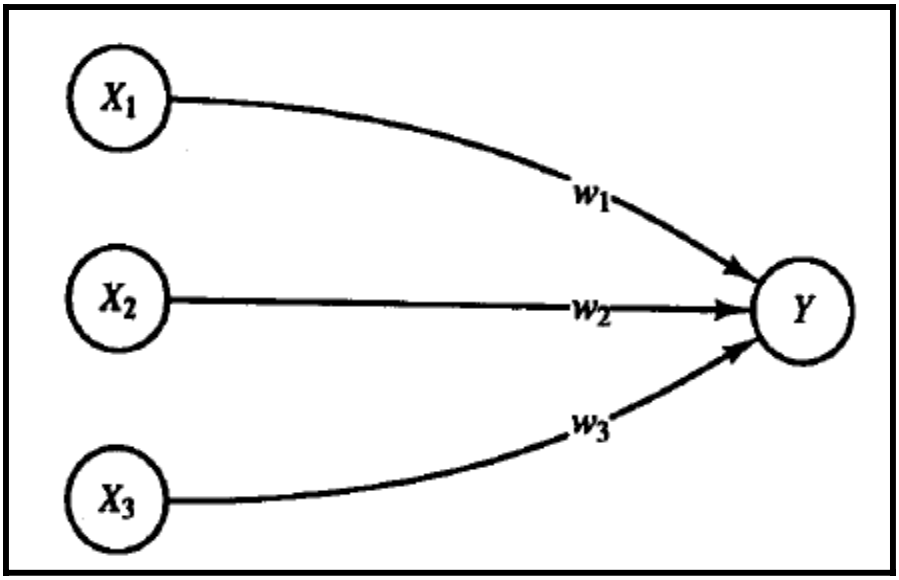
Bir yapay sinir ağı, çok sayıda basit işleme parçacığından oluşur. Bunlar; nöronlar, katmanlar, hücreler ve düğümlerdir. Her nöron, yönlendirilmiş iletişim bağlantıları ile diğer nöronlarla iletişim sağlar. Her nöronun bir dahili durumu vardır. Buna aktivasyon ve ya aktivite düzeyi denir. Bu girdilerin bir fonksiyonudur. Bir nöron, diğer nöronlara sinyal olarak, kendi aktivasyonunu gönderir. Belirli bir

zamanda bir nöron, diğer nöronlara sinyal yayılsa da, sadece tek bir sinyal gönderebilir. Örneğin; Şekil 10'da basit bir ağ gösterilmiştir. Burada Y bir nörondur ve bu nöron girdilerini X_1, X_2 ve X_3 nöronlarından alır. Bu nöronların aktivasyonları sırasıyla x_1, x_2 ve x_3 olarak gösterilmektedir. X_1, X_2 ve X_3 nöronlarından Y nöronuna iletim ağırlıkları sırasıyla w_1, w_2 ve w_3 olarak gösterilmiştir. Y nöronun girdisi - y_{girdi} -, X_1, X_2 ve X_3 nöronlarından gelen ağırlıklandırılmış sinyallerin toplamıdır (Fausett, 1994:3). Bu toplam;

$$y_{girdi} = w_1x_1 + w_2x_2 + w_3x_3 \tag{2.18}$$

şeklinde hesaplanır.

Şekil 10: Yapay Sinir Ağı Örneği



Kaynak: Fausett, 1994, s.4

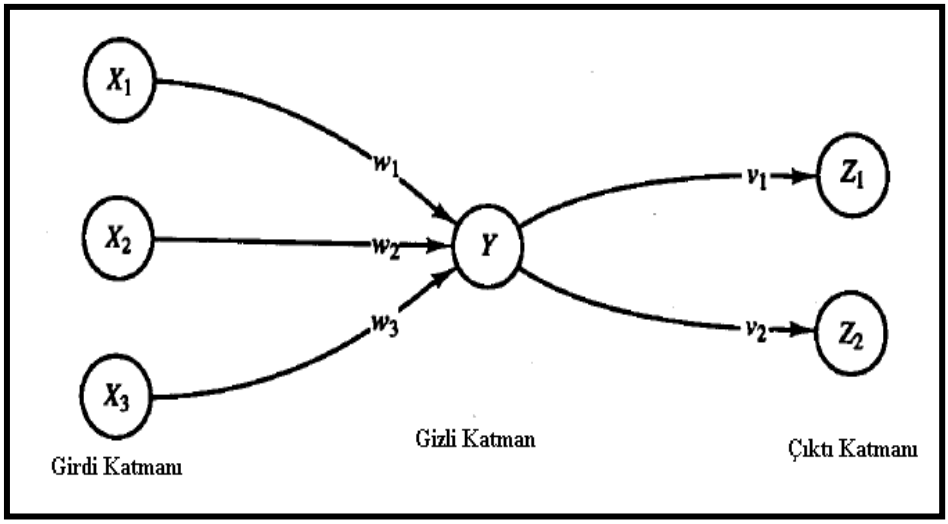
Y nöronunun aktivasyonu y, girdilerin fonksiyonu olarak verilir. Örnek olarak $y = f(y_{girdi})$ fonksiyonu bir lojistik sigmoid fonksiyon olabilir. Lojistik sigmoid fonksiyon;

$$f(x) = \frac{1}{1+\exp(-x)} \tag{2.19}$$

şeklindedir. Bu fonksiyonun dışında, katı sınırlama, simetrik katı sınırlama, doğrusal, doyurulmuş doğrusal, simetrik doyurulmuş doğrusal, hiperbolik tanjant sigmoid fonksiyonları aktivasyon fonksiyonu olarak kullanılabilir (Kantardzic, 2011:203).

Şekil 11, bir yapay sinir ağındaki katmanları göstermektedir. Burada Y nöronu, sırasıyla v_1 ve v_2 ağırlıklarıyla Z_1 ve Z_2 nöronlarına bağlantılıdır. Y nöronu bu her bir katmana y sinyalini gönderir. Ancak bu değerler Z_1 ve Z_2 tarafından farklı alınacaktır. Çünkü her sinyal v_1 ve v_2 ağırlıklar ile ölçeklendirilir. Z_1 ve Z_2 nöronlarının aktivasyonları z_1 ve z_2, birçok nörondan gelen girdilere dayanır (Fausett, 1994:4).

Şekil 11: Yapay Sinir Ağı Katmanları



Kaynak: Fausett, 1994, s.4

Sınıflandırma için kullanılan yapay sinir ağları, bir öğrenme sürecini sürdürür. Bu öğrenme sürecinde çıktı katmanına ulaşabilmek için w ağırlıkları hesaplanır. Öğrenme kümesinde bu ağırlıklar hesaplandıktan sonra eldeki verilerin diğer kısmı kullanılarak, öğrenmenin ne kadar gerçekleştiği test edilir. Eğer test sonucunda bulunan ağırlıkların etkinliği doğrulanırsa öğrenme tamamlanmış olur. Aksi halde, w ağırlıkları üzerinde düzeltme ya da yeniden değer hesaplama işlemleri yapılır. Öğrenme işlemi tamamlandıktan sonra herhangi bir yeni verinin eldeki ağırlıklarla bağlı olduğu sınıf hesaplanabilir (Silahtaroğlu, 2008:70).

2.2 Kümeleme Teknikleri

Kümeleme, birliktelik ölçülerini kullanarak, örneklemleri gruplara otomatik sınıflandırmak için kullanılan yöntemdir. Böylece, bir kümeye ait olan örnekler bu kümenin elemanları ile benzerdir. Kümeleme yöntemi sisteminin girdileri, örneklem kümesi ve benzerliğin ölçüsüdür. Kümeleme yönteminin çıktıları, örneklem kümelerinden elde edilen kümeler grubudur (Kantardzic, 2011:250).

Kümeleme yöntemi, sınıflandırmadan farklıdır. Kümelemede, sınıflandırmadaki gibi bir hedef değişken yoktur. Kümeleme yöntemi, veri kümelerini sınıflandırmaya, tahminlemeye çalışmaz. Bunların yerine, bütün veri kümelerini, göreceli olarak benzerliğin en fazla olduğu kümelere ayırmayı amaçlar. Bu yüzden kümeleme yöntemleri denetimsiz öğrenmeye en iyi örnektir (Larose, 2008:147).

Kümeleme yöntemlerinin kullanıldığı yerler;

- Büyük bütçeye sahip olmayan şirketler için uygun ürünün hedef pazarlaması,
- Finansal davranışları iyi niyetli veya şüpheli olarak kümeleyerek hesap denetimi,
- Yüzlerce değişkene sahip veri kümelerinde boyut indirgeme,
- Çok fazla miktardaki genlerin benzer davranışlar gösterenlerini kümeleme

örnek olarak gösterilebilir.

Kümeleme algoritmaları benzerliklerin veya uzaklıkların ölçülmesini temel alarak işlerler. Benzerlik ölçüsü olarak genelde aşağıdaki ölçüler kullanılmaktadır (Silahtaroğlu, 2008:101).

a. Cosine korelasyon ölçüsü

$$\text{ben}(X_m, X_j) = \frac{\sum_{i=1}^{n} x_{mi}.x_{ji}}{\sqrt{\sum_{i=1}^{n} x_{mi}^{2} . \sum_{i=1}^{n} x_{ji}^{2}}} \qquad (2.20)$$

b. Dice katsayı ölçüsü

$$ben(X_m, X_j) = \frac{2\sum_{i=1}^{n} x_{mi}.x_{ji}}{\sum_{i=1}^{n} x_{mi}^{\,2} + \sum_{i=1}^{n} x_{ji}^{\,2}} \tag{2.21}$$

c. Jaccard ölçüsü

$$ben(X_m, X_j) = \frac{\sum_{i=1}^{n} x_{mi}.x_{ji}}{\sum_{i=1}^{n} x_{mi}^{\,2} + \sum_{i=1}^{n} x_{ji}^{\,2} - \sum_{i=1}^{n} x_{mi}x_{ji}} \tag{2.22}$$

d. Overlap ölçüsü

$$ben(X_m, X_j) = \frac{\sum_{i=1}^{n} x_{mi}.x_{ji}}{\min\left(\sum_{i=1}^{n} x_{mi}^{\,2} + \sum_{i=1}^{n} x_{ji}^{\,2}\right)} \tag{2.23}$$

Kümeleme algoritmalarında uzaklık ölçüleri de kullanılmaktadır. Bu uzaklılar aşağıda verilmiştir.

a. Öklid uzaklığı

$$d(i, j) = \sqrt{\sum_{k=1}^{n} (x_{ik} - x_{jk})^2} \tag{2.24}$$

b. Manhattan uzaklığı

$$d(i, j) = \sum_{k=1}^{n} \left(\left| x_{ik} - x_{jk} \right| \right) \qquad i,j=1,2\ldots n;\ k=1,2\ldots p \tag{2.25}$$

c. Minkowski uzaklığı

$$d(i, j) = \left(\sum_{k=1}^{n} \left(\left| x_{ik} - x_{jk} \right| \right)^m \right)^{\frac{1}{m}} \qquad i,j=1,2\ldots n;\ k=1,2\ldots p \tag{2.26}$$

Kümeleme algoritmalarını genel olarak üçe ayırmak mümkündür. Bunlar;

- Hiyerarşik kümeleme yöntemleri
- Bölümlemeli kümeleme yöntemleri
- Yoğunluk tabanlı kümeleme yöntemleri

2.2.1 Hiyerarşik Kümeleme Yöntemleri

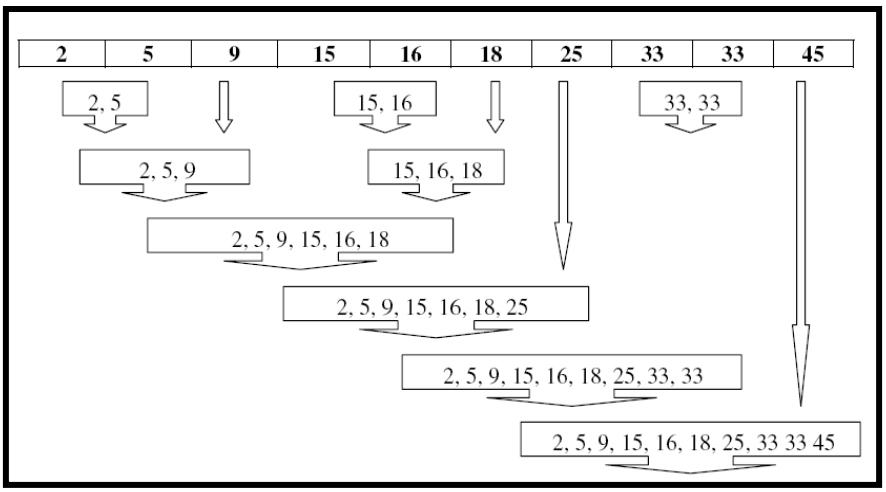
Hiyerarşik kümeleme yöntemlerinde girdilerin bir parçası olarak kümelerin sayısı belirtilmez. Şöyle ki, sistemin girdileri (X,s) olduğu kabul edilsin. Burada X örneklem kümeleri ve s benzerliğin ölçüsüdür. Sistemin çıktıları, aşamalandırılmış kümelerdir. Burada amaç, yakınsaklıklar gerçekleşene kadar, bölünmelerin geliştirilmesi için iterasyonlar kullanarak en uygun çözüme ulaşmaktır (Kantardzic, 2011:260). Hiyerarşik kümeleme yöntemleri, dendogram grafikleri ile görselleştirilebilir. En çok kullanılan hiyerarşik kümeleme yöntemleri en yakın komşu, en uzak komşu ve BIRCH algoritmalarıdır.

2.2.1.1 En Yakın Komşu Algoritması

En yakın komşu algoritması, tek bağlantı kümeleme yöntemi olarak da adlandırılır. Bu algoritma, bir kümenin elemanlarından başka bir kümenin elemanlarına olan en kısa uzaklıkları denkleştirmek için iki küme arasındaki uzaklıları göz önüne alır. Eğer veriler benzer özelliklerden oluşuyorsa, kümeler arasındaki benzerliği, bir kümenin elemanlarından diğer kümenin elemanlarına en çok benzeyene denkleştirmeyi amaçlar (Maimon ve Rokach, 2005:331).

Bu algoritmada öncelikle her bir kayıt kendi bir kümeyi oluşturur. Sonra algoritma, iki kümeye ait kayıtlar arasındaki en küçük uzaklıları araştırır. Şekil 12, en yakın uzaklık algoritması işleyişine bir örnektir (Larose, 2008:150).

Şekil 12: En Yakın Uzaklık Algoritması Örneği



Kaynak: Larose, 2008, s.151

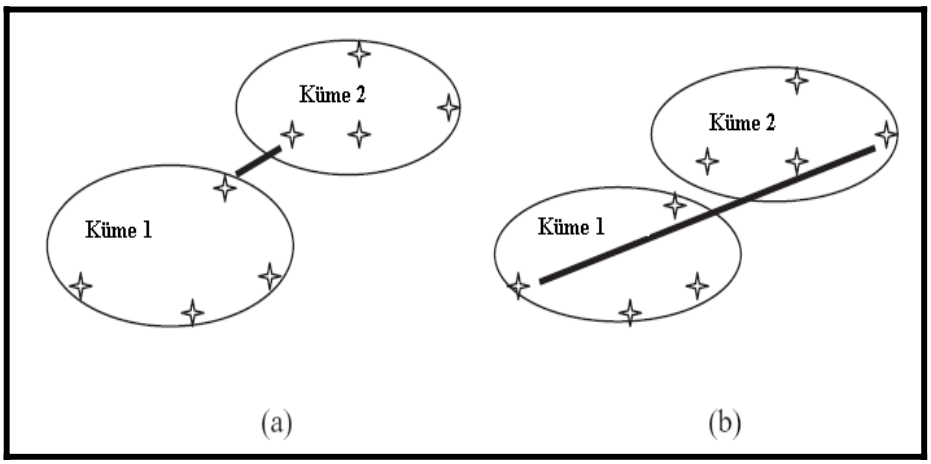
Şekil 12' de {2.5,9,15,16,18,25,33,33,45} sayıları öncelikle hepsi birer kümeyi temsil eder. 33 sayısı iki kez kullanıldığından ikisi bir küme olarak gösterilir. En yakın komşu algoritması ilk olarak iki küme arasındaki uzaklık en yakın olan kümeyi araştırır. Burada aralarındaki uzaklık en yakın olan iki küme 15 ve 16 'dır. Aralarındaki uzaklık bir birimdir. Bunlar bir kümede birleştirilir. Geriye kalan kümeler arasındaki minimum uzaklığa sahip uygun küme {15,16} kümesine en yakın olan {18} kümesidir. Bunlar arasındaki uzaklık üç birimdir. Bu kümeler bir kümede birleştirilir. Oluşan yeni küme {15, 16, 18} kümesidir. Geriye kalan kümeler arasından iki küme arasındaki en yakın uzaklığa sahip küme {2} ve {5} kümeleridir. Aralarındaki uzaklık üç birimdir. Bu kümeler bir küme oluşturur. Oluşan küme {2,5} kümesidir. Aynı şekilde geriye kalan kümeler arasındaki en yakın uzaklığa sahip kümeler {2,5} ile {9} kümeleridir. Aralarındaki uzaklık dört birimdir. Bu kümeler bir küme oluşturur. Yeni küme {2, 5, 9} olur. Geriye kalan kümeler arasındaki en yakın uzaklığa sahip kümeler {2,5, 9} ve {15, 16, 18} kümeleridir. {2,5,9} kümesinin elemanı 9 ile {15, 16, 18} kümesinin elemanı 15 arasındaki uzaklık altı birimdir. Bunlar bir küme oluşturur (Larose, 2008:151). Bu şekilde kümeleme işlemi hiyerarşik olarak sürdürülür. Basit olarak kümeleme işlemi örneklendiğinden uzaklıklar basitleştirilmiştir. Bu algoritmada daha çok Öklid uzaklık yöntemi kullanılır. Değerler arasındaki Öklid uzaklıkları bulunarak en yakın olan iki tanesi

seçilerek işleme başlanır. Bunlar bir kümede birleştirilir. Daha sonra işlem hiyerarşik olarak sürdürülür.

2.2.1.2. En Uzak Komşu Algoritması

Bu algoritma tam bağlantı kümeleme algoritması olarak da adlandırılır. Algoritmanın işleyişi en yakın komşu algoritmasına benzer. En yakın komşu algoritması iki küme arasındaki en yakın uzaklıkları araştırırken en uzak komşu algoritması, en uzak komşuların arasındaki uzaklığı kümeler arsındaki uzaklık olarak tayin eder. Algoritmanın işleyişi, en yakın uzaklık algoritmasındaki gibi en yakın uzaklığa sahip iki kümeyi seçer ve bunları bir kümede birleştirir. Daha sonra bu kümeyi diğer kümelerle işleme sokar ve iki küme arasındaki en küçük uzaklığa sahip iki kümeyi arar. Bulunan bu iki kümeyi de bir küme olarak birleştirir. İlk oluşturulan kümenin elemanları ile diğer kümeler arasındaki uzaklıkları hesaplarken birbirine en uzak olan değerleri iki küme arasındaki uzaklık olarak tayin eder. İşlemler bu şekilde devam ederek hiyerarşik olarak kümeleme işlemini tamamlanır. Şekil 12, en yakın ve en uzak komşu algoritmalarının işleyiş prensiplerini gösterir.

Şekil 13: En Yakın Ve En Uzak Komşu Algoritmaları



Kaynak: Kantardzic, 2011, s.260

Şekil 13’ de (a) ile gösterilen iki kümenin elemanları arasındaki en yakın uzaklığa sahip iki elemanın uzaklığı kümelerin uzaklığı olara tayin edilirken (b) ile gösterilen iki kümenin elemanları arasındaki en fazla uzaklığa sahip iki elemanın

arasındaki uzaklık iki kümenin uzaklığı olarak atanır. Şekil 13, en uzak ve en yakın uzaklık algoritmalarının işleyiş prensibini ve aralarındaki farkı çok iyi şekilde göstermektedir.

2.2.1.3 BIRCH Algoritması

Hiyerarşik kümeleme algoritmalarından olan BIRCH, 1996 yılında Zhang, Ramarkishnan ve Livny tarafından geliştirilmiş bir algoritmadır. BIRCH algoritması iki kavramı ortaya koyar. Bunlar, küme niteleyici (Clustering feature) ve küme niteleyici ağacı (clustering feature tree) kavramlarıdır. Küme niteleyici CF olarak gösterilir (Han, Kamber, 2006:412). Küme niteleyicisi, bir küme hakkında üçlü özetleme bilgisini kullanır. Örneğin, N d-boyutlu veri noktalarını gösteren bir küme olsun. Bu küme, $\{\vec{X}_i\}$ i=1,2,…N ile gösterilsin. Küme niteleyicisi, $CF = \{N, \vec{LS}, SS\}$ olarak tanımlanır. Burada N; veri noktalarının sayısını, $\vec{LS}$; veri noktalarının doğrusal toplamını, SS; veri noktalarının kareler toplamını ifade eder. CF vektörü veri noktalarının oluşturduğu küme hakkında bir fikir verir. Tek başına etkili değildir (Zhang ve diğerleri, 1996:105).

Küme niteleyici ağacı (CF ağacı) iki parametre ile oluşan yüksek dengeli bir ağaçtır. Bu parametreler, dallanma faktörü B ve eşik değeri T olarak gösterilir. Her yapraksız düğüm, en fazla B adet giriş içerir. Bu giriş formu, $[CF_i, çocuk_i]$ i=1,2,…B olarak ifade edilir. "$çocuk_i$" , i. çocuk düğümün (child node) işaretçisidir ve CF_i, bu çocuk düğümleri (child node) gösteren alt kümelerin küme niteleyicisini göstermektedir. Yapraksız düğüm, kendi girdilerinden oluşan bir kümeyi oluşturur. Yaprak düğümü, en fazla L girdiyi içerir. Her yaprak düğümü iki işaretçiye sahiptir. Bunlar "önceki" ve "sonraki" olarak adlandırılan işaretçilerdir. Bütün yaprak düğümler kendi girdileriyle bir kümeyi ifade eder. Ancak bir yaprak düğüm içindeki tüm girdiler bir eşik değerini karşılamak zorundadır. Bu eşik değeri T olarak gösterilir. Ağaç genişliği T 'nin bir fonksiyonudur. Geniş bir eşik değeri, küçük bir ağacı ifade eder (Zhang ve diğerleri, 1996:106). Ağaca yeni noktalar eklendikçe CF ağacı yaratılmış olur. Her bir nokta kendisine en yakın olan yaprağa bağlanır. Noktalar eklene eklene büyüyen yaprak T eşik değerini aşarsa ağaçta bölme işlemi

yapılır (Silahtaroğlu, 2008:113). CF ağacı kurulmadan önce eşik değeri önceden belirlenir. Veri noktaları arasındaki uzaklıklar yardımıyla ağacın kökleri ve yaprakları oluşturulur.

BIRCH algoritması, öncelikle CF ağacını oluşturmak için veritabanını tarar ve CF ağacı oluştuktan sonra eğer kümeler gerçeği yansıtmıyorsa CF ağacına herhangi bir kümeleme yöntemi kullanılarak veri noktaları kümelenmiş olur. Eğer kümeler doğal ise yaprak düğümlerin her biri bir kümeyi oluşturur (Han ve Kamber, 2006:413).

2.2.2 Bölümlemeli Kümeleme Yöntemleri

Hiyerarşik olmayan bir kümeleme yöntemidir. Her bölümlemeli kümeleme yöntemi, hiyerarşik tekniklerin oluşturduğu kümeleme yapısının yerine, tek veri bölmesinden oluşur. Bölümlemeli kümeleme yöntemleri, geniş veri kümelerini içeren uygulamalarda avantaj sağlar. Bu yöntemler, hem bölgesel (bir örneklem alt kümesinde) hem de global (tanımlanmış bütün örneklemler üzerinde) olarak tanımlanan bir ölçüt fonksiyonunu en iyileştirme yoluyla kümeler oluşturur. Global ölçüt, her kümeyi asıl örnek ya da merkez olarak gösterir ve örneklemleri, en çok benzeyen prototiplere göre kümelere atar (Kantardzic, 2011:263).

Verilen n nesneye sahip bir veritabanı olduğu varsayılsın. Burada k ‘ da küme sayısını ifade etsin. Bir bölümleme algoritması, her parça bir kümeyi gösterecek şekilde, veri tabanındaki nesneleri k tane parçaya ayırır. Kümeler benzerlik fonksiyonları kullanılarak en iyileştirilir (Han ve Kamber, 2006:402). En çok kullanılan bölümleme algoritması, k-ortalamalar algoritmasıdır.

- k-ortalamalar algoritması;

k-ortalamalar algoritması, kümelemede etkili bir şekilde kullanılmaktadır. Bu algoritma sadece sayısal verilerde kullanılabilir. Kategorik verilerde kullanılmaz. Bunun yanında, ortalamalardan faydalanılarak işlem yaptığından uç değerlerden çok etkilenir. Bu algoritmanın işleyişi aşağıdaki adımlarda gösterilir (Larose, 2008:153; Silahtaroğlu, 2008:115).

Adım 1

Veri kümesinin kaç parçaya bölüneceğini gösteren k sayısı belirlenir. Bu uygulamayı yapacak olan kişi tarafından belirlenir.

Adım 2

İlk küme merkezleri için rasgele k tane kayıt atanır.

Adım 3

Her kayıt için en yakın küme merkezi bulunur. En yakın uzaklıklar genelde Öklid uzaklığıyla bulunur. Bu kayıtlar yakın oldukları küme merkezlerine göre kümelendirilir.

Adım 4

Bir önceki adımda oluşan kümelerin yeni merkezleri bulunur. Bu yeni merkezlere göre veri kümesindeki en yakın değerler tekrar kümelenir.

Adım 5

Oluşan kümelerde bir değişme olmayana kadar üçüncü adımdan beşinci adıma kadar döngü devam eder.

Adımlardaki merkez kavramı, kümelerin ortalamalarını ifade eder. Bölümleme için geliştirilen algoritmaların çoğu k-ortalamalar algoritmasından türetilmiştir. Bu algoritmalara örnek olarak k-medoids ve CLARANS algoritmaları verilebilir.

2.2.3 Yoğunluğa Dayalı Kümeleme Yöntemleri

Keyfi olarak yayılmış veri noktalarını kümelemek için kullanılan yöntemlerdir. Yüksek yoğunluktaki veri noktalarının oluşturduğu bölgelerin küme olarak gösterilmesi yöntemleridir (Han ve Kamber, 2006:418). Bu yöntemlerin en temel yaklaşımları, en yakın komşu noktaların yerel dağılımları bakımından ölçülebilen yoğunluları ve bağlanabilirlikleridir (Kantardzic, 2011:270). Yoğunluğa dayalı kümeleme yöntemlerinde en çok kullanılan algoritma DBSCAN algoritmasıdır.

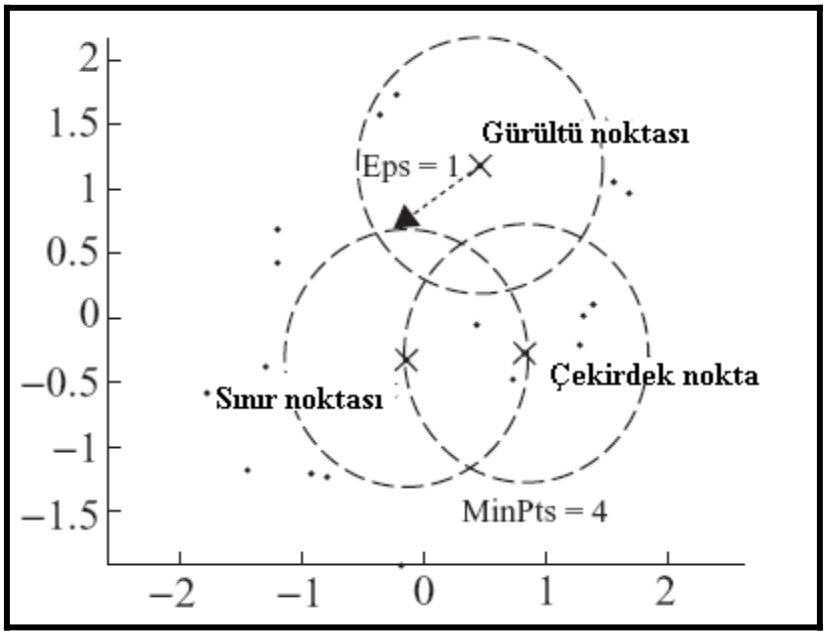
- DBSCAN algoritması

DBSCAN algoritması 1996 yılında Ester, Kriegel, Sander ve Xu tarafından geliştirilmiş bir algoritmadır. Bu algoritmada bir küme, yoğunluk bağlantı noktaları en yüksek veri setleri olarak tanımlanır (Han ve Kamber, 2006:418).

DBSCAN, iki temel kavrama dayanır. Bunlar; yoğunluk ulaşılabilirliği ve yoğunluk bağlanabilirliği. Bu iki kavram iki girdi parametresine bağlıdır. Bunlar; epsilon komşuluk boyutu (ε) ve bir kümede bulunması gereken minimum nokta sayısı (MinPts). Bunlara bağlı olarak DBSCAN algoritmasının anahtar fikri, bir kümenin her noktası için verilen Eps (ε) boyutundaki komşuluk en az bir minimum sayıda nokta (m) içerir (Kantardzic, 2011:270).

Kümenin içinde kalan noktalara çekirdek (core) nokta adı verilir. Eğer iki çekirdek nokta birbirinin komşusu sayılıyorsa aynı kümeye dâhil edilir. Çekirdek nokta olmayan her nokta bir sınır (border) noktadır. Bir sınır noktanın çevresinde yeteri kadar nokta yoktur. Fakat bir çekirdek noktanın komşusu sayılır. Çekirdek veya sınır nokta olmayan her nokta gürültü (noise) noktadır (Özdamar, 2002:31). Bu noktalar Şekil 14'de gösterilmiştir.

Şekil 14: DBSCAN Algoritmasında Çekirdek, Sınır Ve Gürültü Noktaları



Kaynak: Kantardzic, 2011, s.271

DBSCAN algoritması için gerekli tanımlamalar aşağıda gösterilmiştir.

1. Yoğunluğa doğrudan erişilebilirlik (Directly density-reachable): Bir kümede iki nokta p ve q ile gösterilsin. Eğer p noktası Eps (q) genişliğinin bir elemanı ise ve Eps(q) genişliğinin mutlak değeri, kümede bulunması gereken minimum nokta sayısından büyük ise bu p noktası, q noktası sayesinde yoğunluğa doğrudan erişilebilir denilir (Silahtaroğlu, 2008:123).

2. Yoğunluğa erişilebilirlik (Denstiy-reachable): $p_1,...,p_n$ gibi noktalar zinciri olduğu kabul edilsin. $p_1 = q$ ve $p_n = p$ olduğu durumda p_{i+1}, p_i sebebiyle doğrudan erişilebilirdir (Han ve Kamber, 2006:418).

3. Yoğunluk bağlantısallığı (Density-connected): Bir p noktası, bir q noktasına, her ikisi içinde yoğunluğa erişebilirlik sağlayan başka bir o noktası ile bağlanabilir (Silahtaroğlu, 2008:123).

Bunlara bağlı olarak yoğunluk tabanlı kümeleme, yoğunluk bağlantısallığı olan noktalar kümesinin, yoğunluk erişilebilirliğine bağlı olarak ençoklanmasıdır. DBSCAN algoritması, veritabanındaki her noktanın Eps komşuluklarını kontrol eder. Eğer bir p noktasının Eps komşuluğu, küme içinde bulunması gereken nokta sayısından (MinPts) fazla ise çekirdek nokta olarak p ile yeni bir küme yaratılmış olur. Tekrarlı olarak bu çekirdek noktadan, yoğunluğa doğrudan erişilebilecek noktalar toplanır. Bu işlem herhangi bir küme oluşturacak nokta kalmayana kadar devam eder. (Han ve Kamber, 2006:418)

2.3 Birliktelik Kuralları

Birliktelik kuralları keşfi, en önemli veri madenciliği yöntemlerinden biridir. Yerel örüntü keşfi formunda yaygın olarak kullanılır. Birliktelik kuralları denetimsiz öğrenme yöntemlerindendir (Kantardzic, 2011:281).

Barkod sistemlerinin gelişmesi, perakende sektöründe çok sayıda satış verisinin toplanmasını ve saklanmasını mümkün hale getirmiştir. Bu satış verileri sepet verileri olarak da adlandırılır. Bu veriler veri tabanlarında, işlem tarihleri ve bu işlemde alınan nesneler olarak saklanır. Başarılı organizasyonlar bu veri tabanlarından çıkarsamalar yaparak strateji belirlerler. (Agrawal ve Srikant, 1994:487) Bu stratejilerin belirlenmesinde birliktelik kuralları büyük önem taşır.

Perakende sektöründeki bu kurallar pazar sepeti analizi (market basket analysis) olarak adlandırılır.

Birliktelik kuralları, bir veri kümesinde bir veya birden fazla değişkenin diğer değişkenlerle olan birlikteliğini gösterir. Böylelikle eğer-sonra (if-then) durumları üretilerek değişkenler arasında gizli kalmış önemli ilişkiler ortaya çıkarılır (Oladipupo ve Oyelade, 2009:200)

Birliktelik kurallarının kullanım alanlarına örnek olarak;

- Telekomünikasyon ağlarındaki düşüşleri tahminlemek,
- Bir süpermarkette hangi ürünlerin birlikte alındıklarını ortaya çıkarmak,
- Yeni bir ilacın tehlikeli yan etkilerinin olduğu durumlara karar vermek,
- Borsada işlem gören hisse senetlerini arasındaki birliktelikleri ortaya çıkarmak gösterilebilir (Larose, 2008:180).

Bir birliktelik kuralı, iki çeşit kümeden oluşur. Bunlar; önceki (antecedent) ve izleyen (consequent) olarak tanımlanır. İzleyen, sıklıkla tek parça içermeyle sınırlanır. Kurallar, öncekinden sonrakini işaret eden bir ok ile gösterilir. Buna örnek olarak $\{\text{domates}\} \rightarrow \{\text{marul}\}$ şeklindeki bir birliktelik gösterilebilir. Burada $\{\text{domates}\}$ önce alınan ürünü ve $\{\text{marul}\}$ öncekinden sonra alınan ürünü yani izleyen ürünü gösterir (Webb, 2003:27). Kural, "domates alanların %75' i marul almıştır" şeklinde olabilir.

$I = \{i_1, i_2, ..., i_m\}$ ürünlerden oluşan gerçek bir küme olsun. Herhangi bir işlemler kümesi de D ile gösterilsin. Burada her T işlemi bir ürün kümesini oluşturur ve $T \subseteq I$ olarak tanımlıdır. Her işlemle ilgili tek bir belirteç vardır ve buna kısaca TID (transaction identifier) adı verilir. Eğer X, I kümesinde birkaç ürünün kümesini oluşturuyorsa işlem T, X kümesini içerir. Bunun yanında bir birliktelik kuralı, I kümesi X kümesini kapsıyor ise $X \subset I$, I kümesi Y kümesini kapsıyor ise $Y \subset I$ ve $X \cap Y = \varnothing$ koşulunu sağlıyorsa oluşur. Bu $X \rightarrow Y$ kuralı şeklinde gösterilir. Eğer D işlemler kümesinin %c kısmı X ve Y kümelerini içeriyorsa, bu kural c güven düzeyinde sınırlanmış olur. Eğer D işlemler kümesinin %s kısmı X ve Y kümelerinin birleşimlerini destekliyorsa kuralın s destek düzeyine sahip olduğu söylenir (Agrawal ve Srikant, 1994:487). Güven düzeyi, kuralın gücünü, destek seviyesi ise kuralda oluşan örüntülerin sıklıklarını ifade eder (Kantardzic, 2011:282) Bir veritabanında birliktelik kurallarının ortaya çıkarılması, kullanıcının vereceği en küçük destek

(minsup) seviyesi ve en küçük güven (minconf) seviyesinden daha büyük destek ve güven düzeyine sahip kuralların tespit edilmesiyle olur (Silahtaroğlu, 2008:84). Güven ve destek seviyelerinin bulunması formül 2.27 ve 2.28' de gösterilmiştir (Larose, 2008:184).

$$\text{destek seviyesi} = \frac{\text{X ve Y 'yi içeren işlemlerin sayısı}}{\text{bütün işlemlerin sayısı}} \tag{2.27}$$

$$\text{güven seviyesi} = \frac{P(X \cap Y)}{P(X)} = \frac{\text{X ve Y 'yi içeren işlemlerin sayısı}}{\text{X 'i içeren işlemlerin sayısı}} \tag{2.28}$$

Birliktelik kurallarını belirlemek için AIS, SETM, Apriori, AprioriTid algoritmaları geliştirilmiştir. Bunlar içinde geniş veri kümelerinde en çok uygulanan ve kullanılan algoritma apriori algoritmasıdır.

- Apriori algoritması

Apriori algoritması, sıklıkla gözlenen ürün kümelerini içinde gizli örüntüleri ortaya çıkarmak için kullanılan bir birliktelik kuralı algoritmasıdır. Bu algoritmada, veriler, var ya da yok şeklinde kodlanarak işleme sokulur. Boolean olarak bilinen bu sistem var olarak kaydedilenlerin 1 ile yok olanların ise 0 ile kodlanması şeklindedir. Örneğin bir marketten alınan bir ürün 1 değeri ile ifade edilir.

Bu algoritmanın ismi sıklıkla gözlenen ürün kümelerinin önceki bilgilerini kullanmasına dayanır. Apriori, seviye yöntemli (level-wise) bir araştırma olup algoritmanın belirli koşulları sağlayana kadar tekrarlanmasıyla uygulanır. Öncelikle, sıklığı 1 olarak gözlenen ürün kümelerinden bir küme oluşturulur. Bu küme L_1 olarak gösterilir. Bu L_1 kümesi L_2 kümesini bulmak için kullanılır. L_2 kümesi, sıklığı iki olan ürün kümelerinden oluşan kümedir. Bu işlemler sıklığı k olan ürün kümesi L_k, bulunmayana kadar devam eder. Her L_k kümesinin bulunması için ilk olarak veritabanının tümüyle taranması gerekir. Apriori algoritmasının özelliği, sıklıkla gözlenen ürün kümelerinin bütün alt kümelerinin de sıklıkla tekrarlanmasına dayanır (Han ve Kamber, 2006:234). Bu özellik, oluşturulacak aday kümelerinden belirli gözlem sıklığına ulaşmamış alt kümelerin aday kümelerden çıkarılarak aday kümenin tekrar yapılandırılmasına olanak verir. Bu özelliğin nasıl kullanıldığını anlamak için

L_k kümesini bulma için L_{k-1} kümesinden nasıl faydalanıldığını anlamak gerekir. Bu süreç iki basamaklı bir süreçten oluşur. Bunlar birleşme (join) ve budama adımlarıdır.

a. Birleşme adımı: L_k kümesini bulmak için, k ürün kümesinden oluşan aday kümenin, L_{k-1} kümesinin kendi kendiyle birleşmesi sayesinde üretilmesi ile sağlanır. Bu aday küme C_k ile gösterilir (Agrawal ve Ramakrishnan, 1994:490)

b. Budama adımı: C_k aday kümesi, L_k kümesinin, üyelerinin sıklıkla gözlenip gözlenmediğin gösteren bir süperkümesidir (superset) ancak k gözlemden oluşan ürün kümeleri C_k aday kümesi tarafından içerilir. L_k kümesini sonuçlandırmak için C_k aday kümesi içindeki her adayın sayısı, veri tabanının taranmasıyla saptanır. C_k çok büyük boyutta olabilir. Bu durumda hesaplamalar zorlaşır. C_k kümesinin boyutunu indirgemek için apriori özelliğinden faydalanılır yani sıklıkla gözlenmeyen (k-1) ürün kümesi, sıklıkla gözlenen k ürün kümesinin bir alt kümesi olamaz ve bu ürün kümesi aday kümeden çıkarılarak boyut indirgenmiş olur (Han ve Kamber, 2006:235). Bu adımlar aynı zamanda apriori_gen() fonksiyonunu oluşturur (Döşlü, 2008:35).

Apriori algoritmasının işleyişini göstermek için Tablo 7' de verilen veri kümesinden faydalanılsın.

Tablo 7: Apriori Algoritması İçin Örnek Veri Kümesi

TID	Ürün listesi
1	Bilgisayar, LCD, Bilgisayar oyunu
2	LCD, USB kulaklık, Cep Telefonu
3	USB kulaklık, Cep telefonu
4	Bilgisayar, LCD, USB kulaklık, Cep telefonu

Tablo 7' de gösterilen veriler bir elektronik mağazasına farklı zamanlarda gelen müşterilerin satın aldıkları ürünleri gösterir. Burada TID, yapılan işlemlerin belirteçleridir yani müşterileri tanımlamaktadır. Bu örnekte, istenilen minimum destek ölçüsünün %50 olduğu kabul edilsin. 4 kayıt olduğundan istenilen destek sayısının 4 x 0,5=2 olduğuna ulaşılır.

Tablo 8‘ de verilerin apriori algoritmasına uygun şekilde hazırlanışı gösterilmiştir. Burada, örnek veri kümesinden faydalanılarak, 1 ile kodlanmış olan veriler müşterilerin aldıkları ürünleri, 0 ile kodlanılanlar ise almadıkları ürünleri göstermektedir.

Tablo 8: Örnek Veri Kümesinin Kodlanması

TID	BİLGİSAYAR	LCD	USB KULAKLIK	CEP TELEFONU	BİLGİSAYAR OYUNU
1	1	1	0	0	1
2	0	1	1	1	0
3	0	0	1	1	0
4	1	1	1	1	0

Örneğin, ‘1’ işlem numarasına sahip müşteri, bilgisayar, LCD ve bilgisayar oyunu aldığı için 1 olarak kodlanmış ve almadığı ürünler ise 0 olarak kodlanmıştır.

Adım 1

Öncelikle eldeki bütün veriler taranarak C_1 aday kümesi oluşturulur. Bu küme, müşteriler tarafından alınan ürünlerin sıklıklarını gösterir. Tablo 9‘ da bu sıklıklar gösterilmiştir.

Tablo 9: Apriori Örneğine Bağlı C_1 Aday Kümesi

Ürün	Destek değeri
Bilgisayar	2
LCD	3
USB kulaklık	3
Cep telefonu	3
Bilgisayar oyunu	1

Adım 2

Tablo 9' da gösterilen aday küme bulunduktan sonra istenilen minimum destek (2) sayısından az olan ürün algoritmadan çıkarılarak L_1 kümesi oluşturulmuş olur. C_1 aday kümesinde son ürün olan bilgisayar oyunu 1 destek sayısına sahip olduğundan L_1 kümesinde yer almaz. L_1 kümesi Tablo 10' da gösterilmiştir.

Tablo 10: Apriori Örneğine Bağlı L_1 Kümesi

Ürün	Destek değeri
Bilgisayar	2
LCD	3
USB kulaklık	3
Cep telefonu	3

Adım 3

Bulunan L_1 kümesi tekrar taranır ve L_1 kümesindeki ürünlerin ikili kombinasyonları bulunarak C_2 aday kümesi oluşturulur. Bu, ürünlerin ikili olarak alınma sıklıklarını gösterir. Tablo 10' dan faydalanılarak oluşturulur. Tablo 11' de bu aday kümesi gösterilmiştir.

Tablo 11: Apriori Örneğine Bağlı C_2 Aday Kümesi

Ürün grubu	Destek değeri
Bilgisayar - LCD	2
Bilgisayar - USB kulaklık	1
Bilgisayar - Cep telefonu	1
LCD - USB kulaklık	2
LCD – Cep telefonu	2
USB kulaklık – Cep telefonu	3

Adım 4

Tablo 11' de gösterilen aday küme bulunduktan sonra istenilen minimum destek sayısından (2) az olan ürün grubu algoritmadan çıkarılarak L_2 kümesi oluşturulmuş olur. C_2 aday kümesinde bilgisayarı ve USB kulaklığı birlikte alanlar ve bilgisayarı ve cep telefonunu birlikte alanlar 1 destek sayısına sahip olduklarından L_2 kümesinde yer almazlar. L_2 kümesi Tablo 12 ' de gösterilmiştir.

Tablo 12: Apriori Örneğine Bağlı L_2 Kümesi

Ürün grubu	Destek değeri
Bilgisayar - LCD	2
LCD - USB kulaklık	2
LCD – Cep telefonu	2
USB kulaklık – Cep telefonu	3

Adım 5

L_2 kümesi elde edildikten sonra bu küme tekrar taranır ve L_2' deki ürün gruplarının üçlü kombinasyonları bulunarak C_3 aday kümesi oluşturulur. Bu aday küme, {Bilgisayar – LCD - USB kulaklık}, {Bilgisayar – LCD – Cep telefonu} ve {LCD – USB kulaklık – Cep telefonu} şeklinde oluşacaktır. Ancak apriori özelliğine göre sıklıkla gözlenen aday kümelerinin alt kümeleri de sıklıkla tekrarlanan olmalıdır. Bu durumda, C_2 aday kümesinden de görüleceği üzere {Bilgisayar – USB kulaklık} ve {Bilgisayar – Cep telefonu} ürün kombinasyonları minimum destek sayısını sağlamadığından sık tekrarlanan olamaz. Bu iki kümenin, {Bilgisayar – LCD - USB kulaklık} ve {Bilgisayar – LCD – Cep telefonu} kümelerinin bir alt kümesi olması nedeniyle üçlü ürün kombinasyonlarından sadece {LCD – USB kulaklık – Cep telefonu} kümesi, C_3 aday kümesini oluşturur. Bu üçlü ürün grubunun birlikte satın alınma sayısı (destek sayısı) iki olarak tespit edilir. Minimum destek sayısını sağladığından L_3 kümesinde bulunurlar.

Adım 6

Sık tekrarlanan başka ürün grubu bulunmadığından algoritma sona erer.

Sonuç

L_3 kümesinin bütün alt kümeleri değerlendirilerek bulunan güven düzeyleri ile baştan belirlenen minimum güven düzeyi karşılaştırılıp birliktelik kuralları ortaya çıkarılır. L_3 kümesini oluşturan {LCD – USB kulaklık – Cep telefonu} kümesi {LCD – USB kulaklık}, {LCD – Cep telefonu}, {USB kulaklık – Cep telefonu}, {LCD}, {USB kulaklık}, {Cep telefonu} olarak parçalanır. Bu parçalı kümelere göre karar kuralları aşağıdaki şekilde tespit edilir.

Kural 1

$\text{LCD}, \text{USB kulaklık} \rightarrow \text{Cep telefonu}$: LCD ve USB kulaklık alanlar cep telefonu da alır.

$$\text{Kuralın güven düzeyi} = \frac{\text{Üç ürünü birlikte alanların sayısı}}{\text{LCD ve USB kulaklık alanların sayısı}} \tag{2.29}$$

Tablo 8' den faydalanarak üç ürünü birlikte alanların sayısının 2 olduğu görülebilir. Yine aynı tablodan faydalanarak LCD' yi ve USB kulaklığı birlikte alanların sayısının da 2 olduğu bulunur. Formül 2.29' da istenilenleri yerine koyarak 2/2=1 eşitliğine ulaşılır. Bu kuralın güven düzeyi %100 olara tespit edilir.

Kural 2

$\text{LCD}, \text{Cep telefonu} \rightarrow \text{USB kulaklık}$: LCD ve cep telefonu alanlar USB kulaklık da alır.

$$\text{Kuralın güven düzeyi} = \frac{\text{Üç ürünü birlikte alanların sayısı}}{\text{LCD ve Cep telefonu alanların sayısı}} \tag{2.30}$$

Tablo 8' den faydalanarak üç ürünü birlikte alanların sayısının 2 olduğu görülebilir. Yine aynı tablodan faydalanarak LCD ve cep telefonunu birlikte alanların sayısının da 2 olduğu görülür. Formül 2.30 yardımıyla 2/2=1 bulunur ve bu kural içinde güven düzeyi %100 olarak tespit edilmiş olur.

Kural 3

$\text{USB kulaklık}, \text{Cep telefonu} \rightarrow \text{LCD}$ **:** USB kulaklık ve Cep telefonu alanlar LCD alır.

$$\text{Kuralın güven düzeyi} = \frac{\text{Üç ürünü birlikte alanların sayısı}}{\text{USB kulalık ve cep telefonu alanların sayısı}} \tag{2.31}$$

Tablo 8' den faydalanarak formül 2.31' de istenilenler yerine konulursa, 2/3=0.67 bulunur. Kuralın güven düzeyi %67 olarak tespit edilir.

Kural 4

$\text{LCD} \rightarrow \text{USB kulaklık}, \text{Cep telefonu}:$ LCD alanlar USB kulaklık ve cep telefonu da alır.

$$\text{Kuralın güven düzeyi} = \frac{\text{Üç ürünü birlikte alanların sayısı}}{\text{LCD alanların sayısı}} \tag{2.32}$$

Tablo 8' den faydalanarak istenilenler formül 2.32' de yerine konulursa 2/3=0.67 olarak bulunur. Burada güven düzeyi %67 olur.

Kural 5

$\text{USB kulaklık} \rightarrow \text{LCD}, \text{Cep telefonu}:$ USB kulaklık alanlar LCD ve cep telefonu da alır**.**

$$\text{Kuralın güven düzeyi} = \frac{\text{Üç ürünü birlikte alanların sayısı}}{\text{USB kulaklık alanların sayısı}} \tag{2.33}$$

Tablo 8' den faydalanarak istenilenler formül 2.33' de yerine konulursa 2/3=0.67 olara bulunur. Burada güven düzeyi %67 olur.

Kural 6

$\text{Cep telefonu} \rightarrow \text{LCD}, \text{USB kulaklık}:$ Cep telefonu alanlar LCD ve USB kulaklık da alır.

$$\text{Kuralın güven düzeyi} = \frac{\text{Üç ürünü birlikte alanların sayısı}}{\text{Cep telefonu alanların sayısı}} \tag{2.34}$$

Tablo 8' e göre güven düzeyi %67 olarak bulunur.

Sonuç

Algoritma başlamadan önce belirlenen güven düzeyine bağlı olarak bu kurallar değerlendirilir. Örneğin, başlangıçta güven düzeyi %70 olarak belirlendiği varsayılsın. Bu durumda yalnızca %100 güven düzeyine sahip kural 1 geçerli olur. Yani "*LCD ve USB kulaklık alanlar cep telefonu da alır*" kuralı geçerlidir.

ÜÇÜNCÜ BÖLÜM

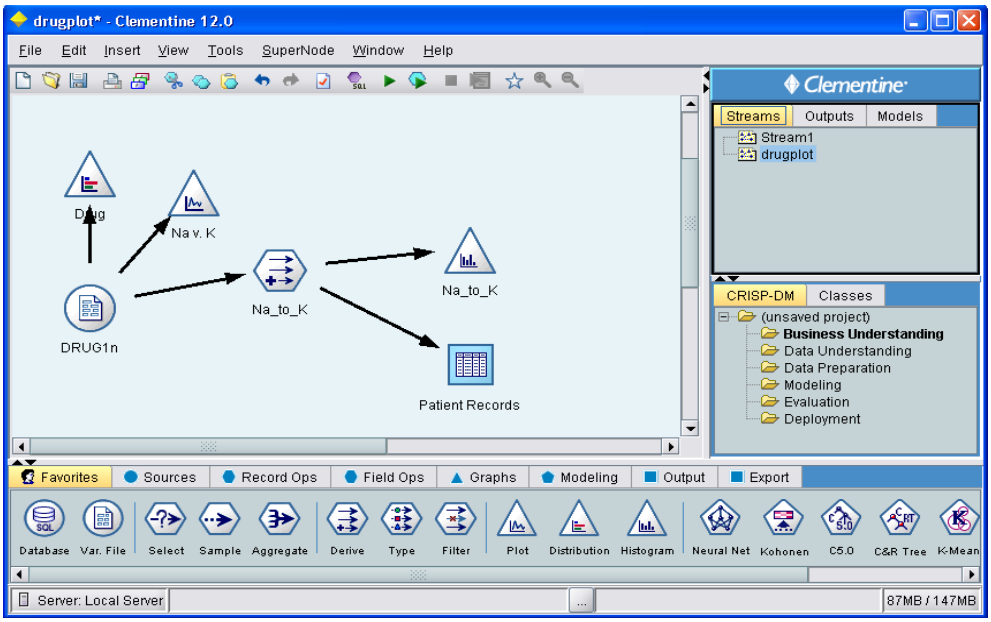
UYGULAMA

Bu uygulama, veri madenciliği tekniklerinden uygun olanın seçilip İMKB Ulusal Pazar' da işlem gören 10 hisse senedi üzerinde uygulanmasını içermektedir. Uygulama, veri madenciliğinde kabul edilmiş bir standart olan CRISP-DM basamakları takip edilerek sürdürülmüştür. Bu basamaklar;

- Yapılacak işi ve ya araştırmayı anlama,
- Kullanılacak veriyi anlama,
- Veriyi hazırlama,
- Modelleme,
- Değerlendirme,
- Sonuçları yayma şeklindedir.

Verilerin araştırılmasında ve model elde edilmesinde, SPSS Clementine 12.0 programı kullanılmıştır. Bu program, veri madenciliğinde yüksek performansta çözüm sunan lisanslı bir yazılımdır. Veri yönetimi bakımından kullanıcıya sağladığı kolaylıklardan, kullanıcı arayüzünün kullanışlı olmasından ve işlem hızının yüksek olmasından dolayı uygulama için bu program seçilmiştir. Bu yazılımın 2009 yılında SPSS Inc. tarafından PASW Modeler olarak ismi değiştirilmiş ve daha sonra IBM şirketinin ortaklığı ile adı IBM SPSS Modeler olmuştur. SPSS Clementine 12.0 yazılımının kullanıcı arayüzünden bir görünüm Şekil 15' de gösterilmiştir.

Şekil 15: SPSS Clementine 12.0 Yazılımında Örnek Arayüz



Şekil 15' deki arayüzde source (kaynak), record ops (kayıt operasyonları), field ops (alan operasyonları), graphs (grafikler), modeling (modelleme araçları), output (çıktı araçları) seçenekleri bulunmaktadır. Bu seçeneklerden hangi kaynaktan veri alınacağını seçmek için *source*, kayıtlar üzerinde işlemler yapmak için *record ops*, kayıtların tiplerini belirlemek, uygulamaya ilişkin filtreler oluşturmak gibi operasyonlar için *field ops*, kayıtlar ile grafik oluşturmak için *graphs,* kayıtlar üzerinde uygulanacak modelin seçilmesi için *modeling* ve çıktının çeşidini belirlemek için *output* seçenekleri kullanılır. Ortadaki alan ise modelin akışını gösterir. Kullanıcı, yukarıda bahsedilen alt alandan kullanacağı seçenekleri ortaya taşır ve bunları oklarla birbirine bağlayarak akış şemasını oluşturur. Şekil 15' de basit bir akış şeması verilmiştir. Akış şemasının çalıştırılması ile kullanıcının elde etmek istediği sonuçlar ortaya çıkar.

3.1 Yapılacak İşi veya Araştırmayı Anlama

Yapılacak araştırmayla, İMKB Ulusal pazarda işlem gören 10 şirketin hisse senedi endekslerinin artışları ve azalışları arasındaki birlikteliklerin ortaya çıkarılması amaçlanmıştır. Birliktelik kurallarının açık bir şekilde görülebilmesi için şirket sayısı 10 olarak belirlenmiştir. Uygulamada sağlık sektöründen, enerji

sektöründen, spor sektöründen, yatırım sektöründen, teknoloji sektöründen, perakende sektöründen, bankacılık sektöründen, gıda sektöründen, ulaştırma sektöründen ve orman ürünleri sektöründen birer tane şirket rasgele olarak ve bu şirketlerin birbirinden bağımsız olması göz önünde bulundurularak seçilmiştir. Örneğin inşaat sektöründe faaliyet gösteren ve hisselerinin artış ve azalışları birbirini doğrudan etkileyen iki firma uygulamaya katılmamıştır. Uygulamaya katılan şirketler ve bu şirketlere ait hisse senedi kısaltmaları Tablo 13' de gösterilmiştir.

Tablo 13: Uygulamaya Katılan Şirketler Ve Hisse Senedi Kısaltmaları

Şirketler	Kısaltmaları
ACIBADEM SAĞLIK	ACIBD
AKSA ENERJİ	AKSEN
BEŞİKTAŞ FUTBOL YAT.	BJKAS
GLOBAL YAT. HOLDİNG	GLYHO
LOGO YAZILIM	LOGO
T. HALK BANKASI	HALKB
TESCO KİPA	KIPA
T. TUBORG	TBORG
VİKİNG KAĞIT	VKING
TÜRK HAVA YOLLARI	THYAO

3.2 Kullanılacak Veriyi Anlama

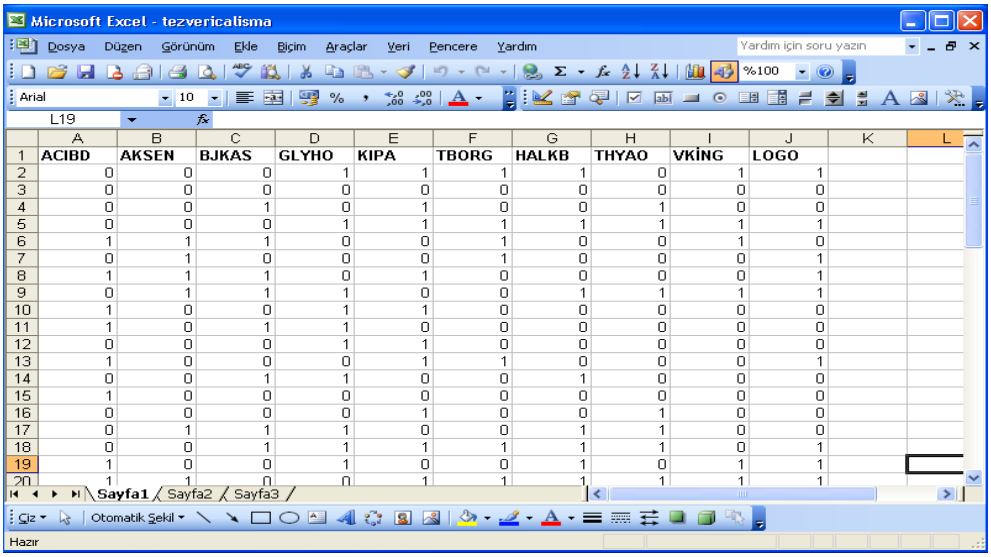
Uygulamada kullanılan veriler, Tablo 13' de verilen 10 şirkete ait hisse senedi endekslerinden oluşmaktadır. Bu veriler, İMKB resmi sitesinde 01.07.2011 ve 03.10.2011 tarihleri arasında yayımlanan, altmış iki işlem gününün ikinci seans kapanış bültenlerinden elde edilmiştir. Verileri anlamak için keşfedici veri analizi uygulanmıştır. Bunun için öncelikle verinin hazırlanması gerekmektedir.

3.3 Veriyi Hazırlama

Uygulamada kullanılan veriler hazır olarak elde edildiğinden herhangi bir kayıp veri yoktur. Uygulamayla, 10 şirketin hisse senetleri endekslerindeki değişmeler arasındaki birlikteliklerin ortaya çıkarılması amaçlandığından, kullanılacak algoritma olarak, birliktelik kuralı algoritmalarından Apriori algoritması belirlenmiştir. Verilerin bu algoritmaya uygun olarak hazırlanması gerekmektedir.

Bunun için uygulamada kullanılacak şirketlere ait hisse senetleri, altmış üç günlük işlem günü içerisinde, bir gün öncesinin endeks değerine göre değişimleri göz önüne alınarak tekrar kodlanmıştır. Örneğin 02.07.2011 tarihinde Acıbadem Sağlık hissesi 2.5 puan ile kapatmış ve 03.07.2011 tarihinde yine aynı hisse 2.4 puan ile kapatmış olsun. Bu hisse puanı azaldığından dolayı algoritmadaki kodu 0 olarak girilmiştir. Diğer durumlar ise 1 olarak kodlanmıştır. Bu kodlama MS Office Excel 2003 ortamında girilmiştir. Kodlamaya ilişkin örnek bir ara yüz Tablo 14' de verilmiştir.

Tablo 14: Uygulama Verileri



	A	B	C	D	E	F	G	H	I	J	K	L
1	ACIBD	AKSEN	BJKAS	GLYHO	KIPA	TBORG	HALKB	THYAO	VKİNG	LOGO		
2	0	0	0	1	1	1	1	0	1	1		
3	0	0	0	0	0	0	0	0	0	0		
4	0	0	1	0	1	0	0	1	0	0		
5	0	0	0	1	1	1	1	1	1	1		
6	1	1	1	0	0	1	0	0	1	0		
7	0	1	0	0	0	1	0	0	0	1		
8	1	1	1	0	1	0	0	0	0	1		
9	0	1	1	1	0	0	1	1	1	1		
10	1	0	0	1	1	0	0	0	0	0		
11	1	0	1	1	0	0	0	0	0	0		
12	0	0	0	1	1	0	0	0	0	0		
13	1	0	0	0	1	1	0	0	0	1		
14	0	0	1	1	0	0	1	0	0	0		
15	1	0	0	0	0	0	0	0	0	0		
16	0	0	0	0	1	0	0	1	0	0		
17	0	1	1	1	0	0	1	1	0	0		
18	0	0	1	1	1	1	1	1	0	1		
19	1	0	0	1	0	0	1	0	1	1		
20	1	1	0	0	1	1	1	1	1	1		

Spss Clementine 12.0 programı kullanılarak veriler arasında güçlü korelasyon olup olmadığı kontrol edilmiştir. Bu program Pearson Korelasyon katsayısını kullanarak değişkenler arası korelasyonları güçlü, zayıf, orta şeklinde sınıflandırmaktadır. Pearson korelasyon katsayısı şu formülle hesaplanır;

$$\rho_{X,Y} = \frac{Cov(X,Y)}{\sigma_X \sigma_Y} = \frac{E((X-\mu_X)(Y-\mu_Y)}{\sigma_X \sigma_Y} \qquad (3.1)$$

Bu formül düzenlendiğinde;

$$\rho_{X,Y} = \frac{E(XY)-E(X)E(Y)}{\sqrt{E(X^2)-(E(X))^2}\ \sqrt{E(Y^2)-(E(Y))^2}} \qquad (3.2)$$

formül 2.36' da gösterilen şekliyle daha kullanışlı hale getirilebilir.

Ayrıca bu analiz ile verilerin ortalamaları (mean), en küçük (min) ve en büyük (max) değerleri, gözlem sayıları (count), varyans (variance), standart sapma

(standard deviation) istatistik değerleri görülebilir. Bu analiz ile elde edilen korelasyon değerleri Tablo 15'de verilmiştir.

Tablo 15: Değişkenler Arasındaki Korelasyon Değerleri

	ACIBD	AKSEN	BJKAS	GLYHO	LOGO	HALKB	KIPA	TBORG	VKING	THYAO
ACIBD	1	0,347	0,006	0,127	0,12	-0,046	0,136	0,131	0,356	0,231
AKSEN		1	0,343	0,173	0,206	0,316	0,077	0,158	0,397	0,25
BJKAS			1	0,295	0,049	0,283	0,088	0,029	0,13	0,059
GLYHO				1	0,289	0,388	0,165	0,098	0,323	0,131
LOGO					1	0,501	0,114	0,3	0,327	0,338
HALKB						1	0,151	0,204	0,427	0,309
KIPA							1	0,029	0,065	0,189
TBORG								1	0,258	0,257
VKING									1	0,356
THYAO										1

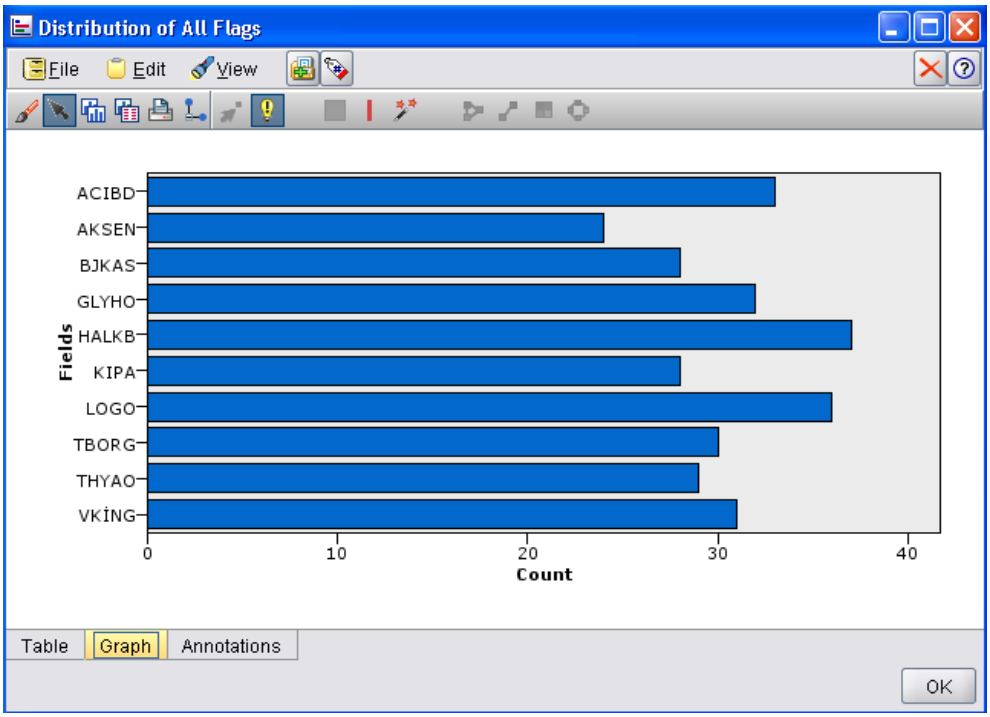
Tablo 15' de görüldüğü gibi birbirleri arasında güçlü korelasyona sahip herhangi iki değişken bulunmamaktadır. Eğer değişkenler arasında 0,70 ve üzeri korelasyon değerleri bulunsaydı güçlü korelasyondan bahsedilirdi. Uygulamanın amacı, değişkenler arasındaki birliktelik kurallarına ulaşmak olduğundan birbirleri ile güçlü korelasyonlara sahip değişkenlerin bulunmaması yanıltıcı sonuçların en aza indirilmiş olması demektir. Değişkenlere ait tanımlayıcı istatistikler Tablo 16' da gösterilmiştir.

Tablo 16: Değişkenlere Ait Tanımlayıcı İstatistikler

DEĞİŞKENLER	GÖZLEM SAYILARI	ORTALAMA	STANDART SAPMA	VARYANS
ACIBD	62	0,532	0,503	0,253
AKSEN	62	0,387	0,491	0,241
BJKAS	62	0,452	0,502	0,247
GLYHO	62	0,516	0,504	0,254
KIPA	62	0,452	0,502	0,252
TBORG	62	0,484	0,504	0,254
HALKB	62	0,597	0,495	0,245
THYAO	62	0,468	0,503	0,253
VKING	62	0,500	0,504	0,254
LOGO	62	0,581	0,497	0,247

Bunun yanında değişkenler arasında sapan değerlerin veya aşırı uç değerlerin bulunması uygulamanın sonucunu olumsuz etkileyeceğinden sapan değerler olup olmadığı kontrol edilmiştir. Tablo 17' de değişkenlerin çubuk grafiği gösterilmiştir.

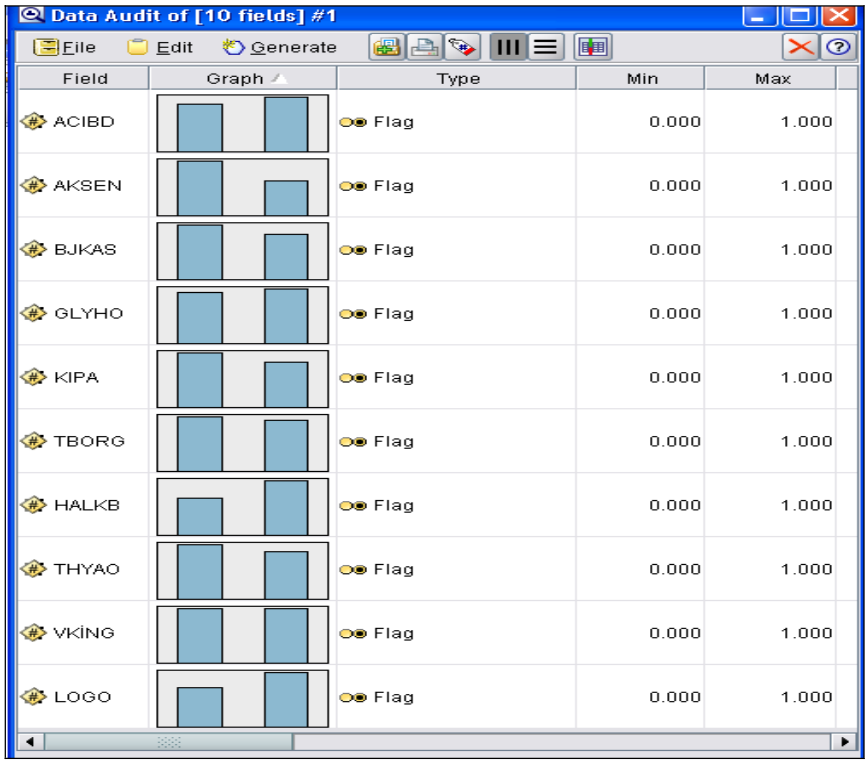
Tablo 17: Değişkenlerin Çubuk Grafiği



Tablo 17' de değişkenlerin çubuk grafiğinden anlaşılacağı gibi uç değerler rastlanmamıştır. Bu dağılım grafiğinde görüldüğü gibi ACIBD değişkeni %53.23, AKSEN değişkeni %38.71, BJKAS değişkeni %45.16, GLYHO değişkeni %51.61, HALKB değişkeni %59.68, KIPA değişkeni %45.16, LOGO değişkeni %58.06, TBORG değişkeni %48.39, THYAO değişkeni %46.77, VKING değişkeni %50 oranları ile artış göstermiştir. Bu oranlarda görüldüğü gibi altmış iki işlem gününde en çok artışı HALKB hisseleri göstermiştir. En az artışı ise AKSEN hisseleri göstermiştir. Değişkenlerin dağılım grafiği ile sapan değer tespiti birinci bölümde Şekil 4 ile gösterilmiştir.

Hisselerin endeks değişimleri Tablo' 18 de açıkça görülebilir.

Tablo 18: Hisse Değişim Grafiği



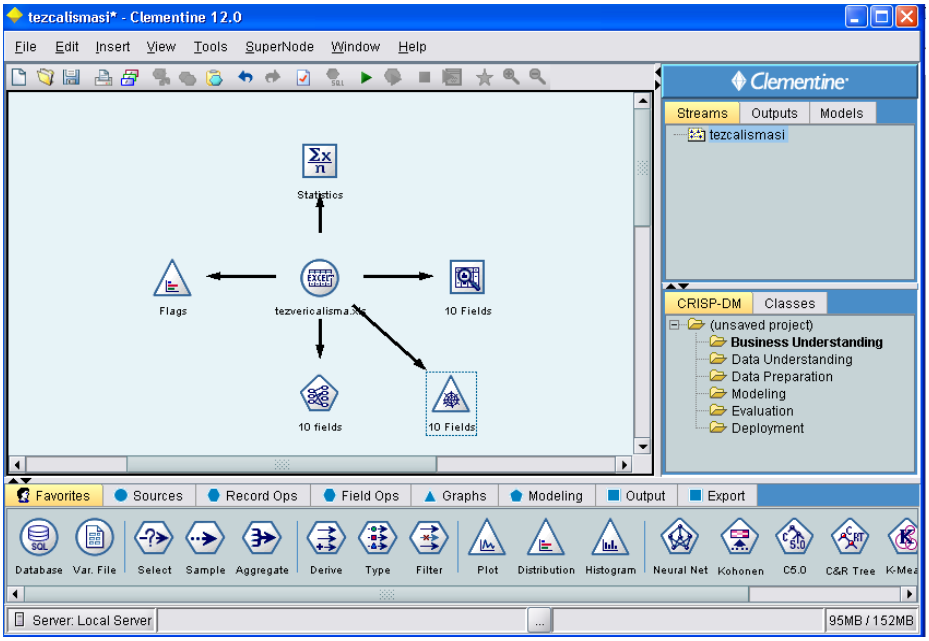
Tablo 18'de verilen ikili grafiklerde sağ tarafta olan grafikler artışları, sol tarafta kalanlar ise azalışları göstermektedir. Apriori algoritmasında değişkenler 1 ve 0 olarak kodlandığı için SPSS Clementine 12.0 programı apriori algoritmasını çalıştırabilmek için kodlanan değişkenleri flag tipi değişken olarak atamaktadır.

Değişkenlerde sapan değerler ve yüksek korelasyona sahip değişkenler bulunmadığından verilerin hazırlığı tamamlanmıştır.

3.4 Modelleme ve Sonuç

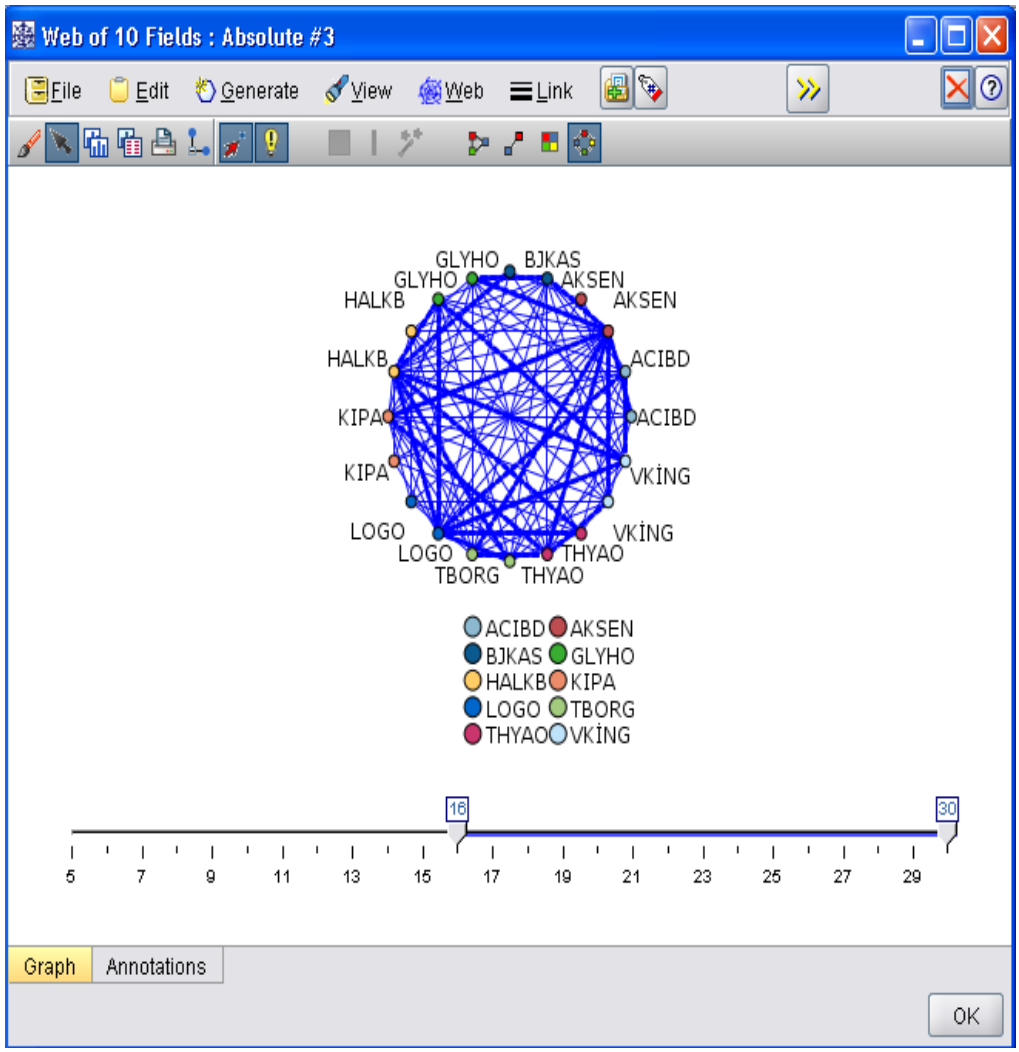
Birliktelik kurallarının ortaya çıkarılması için kullanılacak algoritma Apriori algoritması olarak belirlenmiştir. Bu algoritma kullanılarak bir model elde edilmiştir. Apriori algoritmasında minimum destek ve minimum güven düzeyleri, kullanıcı tarafından analiz öncesi girilmelidir. Fazla kural elde etmek için minimum güven düzeyi %80, minimum destek düzeyi %35 olarak belirlenmiştir. Bu düzeylerin altında güvensiz kurallar ortaya çıkarken, bu düzeylerin çok üstünde ise kural elde edilememektedir. SPSS Clementine 12.0 yazılımında, verilerin hazırlanması ve modelleme aşamalarında kullanılan akış şeması Şekil 16' da gösterilmiştir.

Şekil 16: Yazılımda Kullanılan Akış Şeması



Şekil 16' da gösterilen akış şeması çalıştırıldığında birliktelik kuralları ve değişkenlerin birlikteliklerini gösteren web grafiği ortaya çıkmaktadır. Bu grafiğe ait sonuçlar Şekil 17' de gösterilmiştir.

Şekil 17: Değişkenlere Ait Web Grafiği



Web grafiği, değişkenler arasındaki bağlantıların gücünü gösteren bir grafiktir. Şekil 17‘ de verilen grafikte, değişkenlerin kendi arasındaki güçlü bağlantılar kalın çizgi ile gösterilmiştir. Kalın çizgilere nazaran daha ince olarak çizilmiş çizgiler ise orta güçlükteki bağlantıları ortaya koymaktadır. Web grafiği ortaya çıkarılırken değişkenler arasında oluşan 20 bağlantı çizgisinin üstündekiler güçlü, 5 bağlantı çizgisinin altında kalanlar ise zayıf bağlantı olarak sınıflandırılmıştır. Arada kalan durumlar ise orta güçlükteki bağlantıları göstermektedir. Grafik karmaşık bir yapıya sahip olduğundan bağlantıların hangileri olduğu ve ne kadar bağlantıya sahip olduklarına dair sonuç çıktısı Tablo 19 ve Tablo 20’ de gösterilmiştir. Bu tablolar değişkenler arasında ki güçlü ve zayıf bağlantıları göstermektedir. Orta güçlükteki bağlantıları gösteren tablo EK’ te verilmiştir. Tablo 19’ da HALKB=”1” ve

LOGO=”1” arasında 29 bağlantı (links) olduğu görülmektedir. Bu, HALKB hissesindeki artış ile LOGO hissesindeki artışın 29 kere birlikte gözlenmesi ile yorumlanabilir. Diğer değişkenlerde buna benzer olarak yorumlanır.

Tablo 19: Güçlü Bağlantıya Sahip Değişkenler

Summary | Controls

Strong Links

Links	Field 1	Field 2
29	HALKB = "1.000000"	LOGO = "1.000000"
26	AKSEN = "0.000000"	BJKAS = "0.000000"
25	GLYHO = "1.000000"	HALKB = "1.000000"
25	HALKB = "1.000000"	VKİNG = "1.000000"
25	AKSEN = "0.000000"	VKİNG = "0.000000"
24	AKSEN = "0.000000"	THYAO = "0.000000"
23	ACIBD = "0.000000"	AKSEN = "0.000000"
23	GLYHO = "1.000000"	LOGO = "1.000000"
23	VKİNG = "1.000000"	LOGO = "1.000000"
22	TBORG = "1.000000"	LOGO = "1.000000"
22	AKSEN = "0.000000"	KIPA = "0.000000"
22	AKSEN = "0.000000"	TBORG = "0.000000"
22	THYAO = "0.000000"	VKİNG = "0.000000"
22	HALKB = "1.000000"	THYAO = "1.000000"
22	THYAO = "1.000000"	LOGO = "1.000000"
22	ACIBD = "1.000000"	VKİNG = "1.000000"
21	GLYHO = "1.000000"	VKİNG = "1.000000"
21	TBORG = "1.000000"	HALKB = "1.000000"
21	AKSEN = "0.000000"	GLYHO = "0.000000"
21	BJKAS = "0.000000"	GLYHO = "0.000000"
21	KIPA = "0.000000"	THYAO = "0.000000"
21	TBORG = "0.000000"	THYAO = "0.000000"
21	ACIBD = "1.000000"	LOGO = "1.000000"
21	BJKAS = "1.000000"	HALKB = "1.000000"

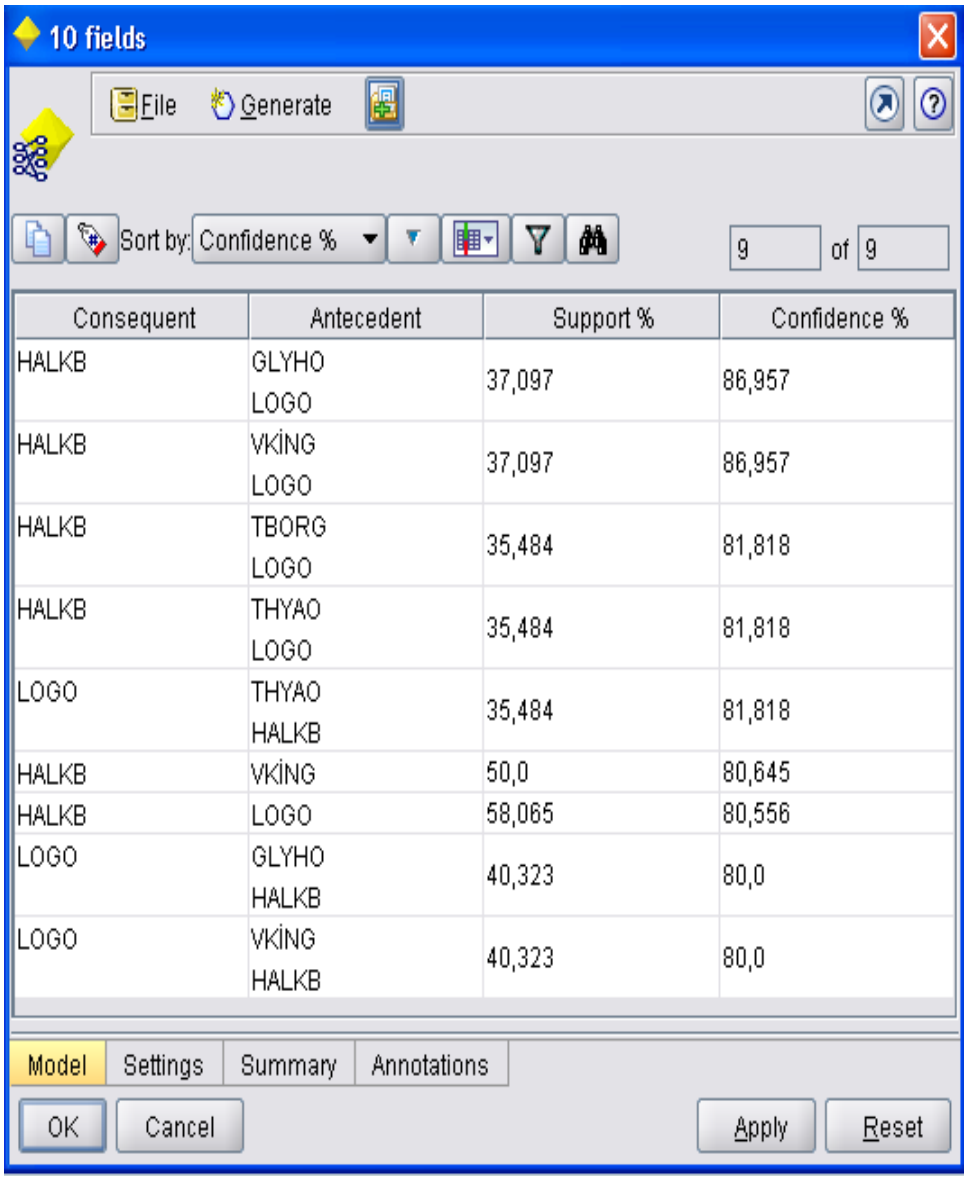
Tablo 20: Zayıf Bağlantıya Sahip Değişkenler

Weak Links

Links	Field 1	Field 2
9	ACIBD = "0.000000"	VKİNG = "1.000000"
9	BJKAS = "1.000000"	GLYHO = "0.000000"
9	KIPA = "1.000000"	HALKB = "0.000000"
9	THYAO = "1.000000"	VKİNG = "0.000000"
9	AKSEN = "1.000000"	GLYHO = "0.000000"
9	AKSEN = "1.000000"	THYAO = "0.000000"
9	TBORG = "1.000000"	HALKB = "0.000000"
9	GLYHO = "1.000000"	LOGO = "0.000000"
8	TBORG = "1.000000"	LOGO = "0.000000"
8	VKİNG = "1.000000"	LOGO = "0.000000"
8	AKSEN = "1.000000"	BJKAS = "0.000000"
8	HALKB = "1.000000"	LOGO = "0.000000"
7	BJKAS = "1.000000"	HALKB = "0.000000"
7	HALKB = "0.000000"	THYAO = "1.000000"
7	THYAO = "1.000000"	LOGO = "0.000000"
7	AKSEN = "1.000000"	LOGO = "0.000000"
7	HALKB = "0.000000"	LOGO = "1.000000"
7	GLYHO = "1.000000"	HALKB = "0.000000"
6	HALKB = "0.000000"	VKİNG = "1.000000"
6	ACIBD = "0.000000"	AKSEN = "1.000000"
6	AKSEN = "1.000000"	VKİNG = "0.000000"
5	AKSEN = "1.000000"	HALKB = "0.000000"

Şekil 16' da gösterilen akış şemasının çalıştırılmasıyla modelleme öncesi belirlenen minimum destek ve güven düzeylerine göre toplam 9 kural ortaya çıkmıştır. Bu kurallar Tablo 21' de gösterilmiştir. Kuralların elde edilmesi ile modelleme adımı sonlanmış olur. Elde edilen bu kurallar ile değerlendirme yapılmalıdır. Bu model ile ortaya konan tahminler hisse senetlerinin altmış iki günlük değişmelerindeki birlikteliklerden oluştuğu için bu hisselerin geleceğe dönük değerlerini tahminlemez. Model ile amaçlanan yalnızca hangi hisse senetlerinin birlikte hareket ettiğini ortaya koymaktır.

Tablo 21: Elde Edilen Birliktelik Kuralları



Tablo 21' de gösterilen kurallar ile ilgili değerlendirmeler aşağıda verilmiştir.

Kural 1

$$\text{HALKB} \rightarrow \text{GLYHO}, \text{LOGO}$$

HALKB hissesi artarken, %86.957 güven ve %37.097 destek düzeyleri ile GLYHO ve LOGO hisseleri de artış göstermektedir.

Kural 2

$$HALKB \rightarrow VKING, LOGO$$

HALKB hissesi artarken, %86.957 güven ve %37.097 destek düzeyleri ile VKING ve LOGO hisseleri de artış göstermektedir.

Kural 3

$$HALKB \rightarrow TBORG, LOGO$$

HALKB hissesi artarken, %81.818 güven ve %35.484 destek düzeyleri ile TBORG ve LOGO hisseleri de artış göstermektedir.

Kural 4

$$HALKB \rightarrow THYAO, LOGO$$

HALKB hissesi artarken, %81.818 güven ve %35.484 destek düzeyleri ile THYAO ve LOGO hisseleri de artış göstermektedir.

Kural 5

$$LOGO \rightarrow THYAO, HALKB$$

LOGO hissesi artarken, %81.818 güven ve %35.484 destek düzeyleri ile THYAO ve HALKB hisseleri de artış göstermektedir.

Kural 6

$$HALKB \rightarrow VKING$$

HALKB hissesi artarken, %80.645 güven ve %50 destek düzeyleri ile VKING hissesi de artış göstermektedir.

Kural 7

$$HALKB \rightarrow LOGO$$

HALKB hissesi artarken, %80.556 güven ve %58.065 destek düzeyleri ile LOGO hissesi de artış göstermektedir.

Kural 8

$$LOGO \rightarrow GLYHO, HALKB$$

LOGO hissesi artarken, %80 güven ve %40.323 destek düzeyleri ile GLYHO ve HALKB hisseleri de artış göstermektedir.

Kural 9

$$LOGO \rightarrow VKING, HALKB$$

LOGO hissesi artarken, %80 güven ve %40.323 destek düzeyleri ile VKING ve HALKB hisseleri de artış göstermektedir.

Uygulama öncesi belirlenen güven ve destek düzeylerine göre kurallar ortaya çıkmıştır. Bütün bu kurallar belirlenen güven ve destek düzeylerine göre nitelikli kurallardır. Ancak bu kurallar arasında, %86.957 güven düzeyi ile en çok güvene sahip kurallar kural 1 ve kural 2‘ dir. 10 şirkete ait hisse senetleri değerleri arasındaki değişmelerin birlikteliği değerlendirilirken bu iki kuralın diğerlerine göre önemi daha büyüktür. Bu hisse senetlerine yatırım kararı alanlar, normal ekonomi şartlarında bu kuralları göz önüne alabilirler.

SONUÇ

Apriori algoritmasına göre Tablo 21' de gösterilen kurallar ortaya çıkmıştır. Bu kuralları değerlendirmeden önce birliktelik kuralları ve apriori algoritması bölümünde örneklenen apriori algoritmasının işleyişi aşağıda özetlenmiştir.

- Birliktelik kurallarının ortaya çıkarılabilmesi için uygulamacı tarafından güven ve destek düzeyleri belirlenir. Oluşturulan birliktelik kümelerindeki değerlerin bu güven ve destek düzeylerine eşit ya da bunlardan büyük olması beklenir.
- Veritabanı taranarak uygulamaya dahil edilecek her bir değişken için destek düzeyleri hesaplanır ve bu destek düzeyi ile önceden girilmiş destek düzeyi karşılaştırılır. Destek düzeyi, önceden girilmiş destek düzeyinden küçük olan ürünler çözümlemeden çıkarılır.
- Kalan değişkenler ikili gruplandırılır ve destek düzeyleri hesaplanır. Bu düzey, önceden girilen düzey ile karşılaştırılır ve eğer önceden girilen destek düzeyinden küçük ise bu ikili kümeler çözümlemeden çıkarılır.
- Kalan değişkenler üçerli olarak gruplanır ve destek düzeyleri hesaplanır. Bu düzey, önceden girilen düzey ile karşılaştırılır ve eğer önceden girilen destek düzeyinden küçük ise bu üçerli kümeler çözümlemeden çıkarılır.
- Sık tekrarlanan başka değişken grubu bulunmayana kadar yukarıdaki adımlar dörderli, beşerli gruplamalar şeklinde devam eder.
- Uygun değişken grubu ortaya çıktıktan sonra birliktelik kuralları üretilir ve her bir kurala ait olan güven düzeyleri hesaplanır.

Veri madenciliğinin amacı, değişkenler arasındaki gizli kalmış örüntüleri ortaya çıkarmaktır. Uygulamada kullanılan değişkenler arasındaki korelasyonlar zayıf olduğundan ortaya çıkan model ile veri madenciliği uygulamasının amacına ulaştığı söylenebilir.

Sermaye piyasaları verileri kullanılarak yapılacak veri madenciliği çalışmalarında yapılacak analizi ve bu analiz için kullanılacak veriyi anlamak, verileri amaca uygun bir şekilde hazırlamak büyük önem taşımaktadır. Bu aşamalarda yapılacak hatalar modelde anlamsız sonuçlara neden olmakta ve süreci uzatmakta ve

sürecin uzamasıyla maliyetler artmaktadır. Sermaye piyasalarına ait veriler anlık değişim gösterdiğinden yapılacak çalışmalarda uygulayıcıların yapılacak işi ve veriyi anlama, veriyi hazırlama adımlarında ileride karşılaşılacak sorunları en aza indirmek için dikkatle çalışmaları büyük önem taşır. Yapılacak analizin amacının ne olduğu, bu amaç için ne tür verilere ihtiyaç olduğu tespit edilmelidir. Kullanılacak veriler tespit edildikten sonra amaca uygun model belirlenmelidir. Veri madenciliği uygulamalarında her veri madenciliği tekniği farklı tip verilerle çalıştığından verilerin modelleme için kullanılacak algoritmaya uygun olarak girilmesi gerekmektedir. Bu tez çalışmasındaki uygulamadan örnekle hisse senetlerinin endeks değerleri doğrudan olarak çalışmaya sokulmamış değerlerdeki azalışlar 0, diğer durumlar ise 1 olarak kodlanmıştır. Kodlama yapılırken kayıp verilerin veya yanlış olarak girilmiş verilerin varlığı dikkate alınmalı ve gerekli işlemler yapılmalıdır. Kayıp verileri yok etmek için aşağıda özetlenen çalışmalardan bir tanesi uygulanabilir.

- Kayıp verilerin yerine bu verilerin ait olduğu değişkenin ortalamasının yazılması
- Kayıp verilerin ait olduğu değişkendeki diğer veriler kullanılarak regresyon yöntemi ile kayıp verilerin tahminlenmesi
- Çok fazla vakit kaybedilme riski olduğundan uygun olmasa da kayıp verilerin çalışmadan çıkarılması

Veriler uygun olarak kodlandıktan ve kayıp veriler düzenlendikten sonra keşfedici veri analizi tekniklerine başvurulur. Keşfedici veri analizi teknikleri değişkenler arasında var olan korelasyonları ve modelleme sonuçlarını doğrudan etkileyen sapan değerlerin tespiti için gereklidir. Örneğin bu tez çalışmasının uygulama kısmında bir çimento şirketine ait hisse senedi ve bir inşaat şirketine ait hisse senedi birlikte çalışmaya sokulsaydı aralarında güçlü korelasyonlar seçilebilirdi. Bu durumda modelleme kısmında bu iki şirkete ait hisse senedi değerlerinin birlikte azalıp arttığı gözlemlenebilirdi. Oysaki veri madenciliğinin amacı değişkenler arasında önceden tahmin edilemeyen ilişkilerin ve örüntülerin ortaya çıkarılmasıdır. Yüksek korelasyona sahip değişenlerin varlığı durumunda bu değişkenlerden biri uygulamadan çıkarılır ve ya iki değişken tek değişkene dönüştürülerek modelleme yapılır. İki veya daha çok değişkenin birleştirilmesi işlemi temel bileşenler analizi ile

gerçekleştirilebilir. Sapan değerlerin varlığı da çalışmayı istenmeyen sonuçlara götürmektedir. Bu değerlerin tespiti için histogram, serpilme diyagramı ve kümeleme tekniklerine başvurulur. Sapan değerlerin varlığı söz konusu ise bu değerler çalışmadan çıkarılmalıdır.

Veriyi hazırlama kısmında karşılaşılacak bir diğer sorun ise değişkenlerin ortalama ve standart sapmaları arasındaki büyük farklılıklardır. Bu durumlarda ortalaması ve standart sapması büyük olan değişkenler diğerleri üzerinde baskın olacak ve modelleme sonucunu olumsuz etkileyecektir. Bunun için veriler üzerinde aşağıda gösterilen dönüşümler yapılır.

- Min- max normalleştirmesi
- Sıfır ortalamalar normalleştirmesi
- Ondalık derecesi ile normalleştirme

Yukarıda anlatılanlara bu tezin veri önişleme kısmında ayrıntılı olarak yer verilmiştir. Veriler hazırlandıktan sonra modelleme adımına geçilir. Modelleme adımında, veri madenciliği çalışmasının amacı büyük önem taşır. Bu amaç doğrultusunda modelleme tekniği seçilir. Bu teknikler, kümeleme, sınıflandırma ve birliktelik kuralları olabilir. Uygun teknik seçildikten sonra bu tekniğe ait algoritma belirlenir ve bir makine öğrenimi programı seçilerek modelleme yapılmış olur. Bu tezin uygulama kısmında 10 şirkete ait hisse senedi endeksi değerleri arasındaki birlikteliklerin ortaya çıkarılması amaçlandığından birliktelik kuralları seçilmiş ve bu kuralları ortaya çıkarabilmek için apriori algoritması uygulanmıştır. Sonuç olarak model ortaya konmuş ve kurallar elde edilmiştir.

Modelleme adımında ne kadar çok veri ile çalışılırsa o kadar iyi sonuçlar elde edilir. Ancak sermaye piyasaları verileri üzerinde çalışma yapılırken olağanüstü ekonomik koşullarda ortaya çıkarılan birliktelik kuralları doğru sonuçlar vermeyebilir. Bu çalışmalar yapılırken mevsimsel değişimler dikkate alınmalıdır. Örneğin kış mevsiminde enerji şirketlerinin karlarında artış olacağı beklentisi hisse senedi değerlerinde artışa neden olabilmekte, bu durum da yapılacak veri madenciliği çalışmasında yanıltıcı sonuçlar vermektedir.

Bu tez çalışması ile amaçlanan, şirketlere ait hisse senedi değerlerinin apriori algoritması kullanılarak birliktelik kurallarının ortaya çıkarılması ve bu modelleme adımına gelene kadar verinin ne tür hazırlıklardan geçmesi ve bu sürecin nasıl yönetilmesi gerektiğini göstermektir. Bu adımlar tamamlandıktan sonra anlık verileri kullanarak birliktelikleri ortaya çıkarmak mümkün hale gelebilir. Bu anlık birliktelik kuralları yatırımcıların kazançlarını artırabilir. Anlık verilerde düzenlemeler yapılması ve modellerin ortaya çıkarılması zaman kaybına neden olacağından buna uygun programlar geliştirilebilir. Bunun yanında, faiz oranları, yurt dışı borsa değerlerindeki değişmeler, petrol fiyatları, enflasyon oranları, işgücü göstergeleri gibi diğer ekonomik göstergeler veri madenciliği sürecine katılarak uygun birliktelikler ortaya konulabilir.

KAYNAKÇA

Adriaans, P., Zantinge, D. (1996). *Data Mining.* India: Pearson Education Ltd.

Agrawal, R., Ramakrishnan, S. (1994). Algorithms For Mining Association Rules. *Proceedings of the 20th International Conference on Very Large Data Bases.* (ss. 487-499), Chile.

Akgöbek, Ö., Çakır, F. (2009). Veri Madenciliğinde Bir Uzman Sistem Tasarımı. *6. Akademik Bilişim Konferansı Bildirileri.* (ss. 801-806), Düzenleyen Harran Üniversitesi. Şanlıurfa. 11-13 Şubat 2009.

Akpınar, H. (2000). Veri Tabanlarında Bilgi Keşfi Ve Veri Madenciliği. *İ.Ü. İşletme Fakültesi Dergisi.* 29(1): 1-22.

Alpaydın, E. (2010). *Introduction To Machine Learning.* Cambridge: MIT Press.

Argüden, Y., Erşahin, B. (2008). *Veri Madenciliği: Veriden Bilgiye, Masraftan Değere.* İstanbul: Arge Danışmanlık Yayınları.

Berry, M.W., Browne, M. (2006). *Lecture Notes In Data Mining.* Singapore: World Scientific Publishing Co. Pte. Ltd.

Berthold, R.M., Borgelt, C., Höppner, F. ve Klawonn, F. (2010). *Guide To Intelligent Data Analysis: How To Intelligently Make Sense Of Real Data.* London: Springer – Verlag London Limited.

Cabena, P., Hadjinian, P., Stadler, R., Verhees, J. ve Zanasi, A. (1998). *Discovering Data Mining: From Concept To Implementation.* Upper Saddle River: Prentice Hall.

Döşlü, A. (2008). *Veri Madenciliğinde Market Sepet Analizi Ve Birliktelik Kurallarının Belirlenmesi.* Yüksek Lisans Tezi. İstanbul: Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü.

Fausett, L. (1994). *Fundementals Of Neural Networks.* USA: Prentice Hall.

Fayyad, U., Piatetsky-Shapiro, G., Smyth, P. (1996). From Data Mining To Knowledge Discovery In Databases. *AI Magazine.* 17(1):37-54.

Gorunescu, F. (2011). *Data Mining Concepts, Models And Techniques.* Berlin: Springer.

Han, J., Kamber, M. (2006). *Data Mining: Concepts And Techniques.* San Francisco: Elsevier Inc.

Hand, D., Mannila, H., Smyth, P. (2001). *Principles Of Data Mining.* Cambridge: MIT Press.

Jackson, J. (2002). Data Mining: A Conceptual Overview. *Communication Of The Association For Information System Magazine.* 8(1): 267-296

Jensen, B.S. (2006). *Exploratory Data Mining In Music.* Master Thesis. Denmark: Technical University Of Denmark Department Of Informatics And Mathematical Modelling.

Kohavi, R., Quinlan, R. (2002). Decision Tree Discovery. *Handbook Of Data Mining And Knowledge Discovery* (ss. 267-276). Oxford: Oxford University Press.

Koyuncugil, A.S. (2007). *Borsa Şirketlerinin Sektörel Risk Profillerinin Veri Madenciliği ile Belirlenmesi.* Sermaye Piyasası Kurulu Araştırma Raporu.

Küçüksille, E. (2009). *Veri Madenciliği Süreci Kullanılarak Portföy Performansının Değerlendirilmesi Ve İMKB Hisse Senetleri Piyasasında Bir Uygulama.* Doktora Tezi. Isparta: Süleyman Demirel Üniversitesi Sosyal Bilimler Enstitüsü İşletme Anabilim Dalı.

Larose, T.D. (2005). *Discovering Knowledge In Data: An Introduction To Data Mining.* New Jersey: A. John Willey&Sons, Inc.

Maimon, O., Rokach, L. (2005). *Data Mining And Knowledge Discovery Handbook.* USA: Springer.

Maimon, O., Rokach, L. (2008). *Data Mining With Decision Trees: Theory And Applications.* Singapore: World Scientific Publishing Co. Pte. Ltd.

Nisbet, R., Elder, J., Miner, G. (2009). *Handbook Of Statistical Analysis And Data Mining Applications.* Canada: Elsevier Inc.

Oladipupo, O.O., Oyelade, O.J. (2009). Knowledge Discovery From Students' Result Repository: Association Rule Mining Approach. *Intenational Journal Of Computer Science And Security.* 4(1): 199-207.

Özçakır, C.F. (2006). *Müşteri İşlemlerindeki Birlikteliklerin Belirlenmesinde Veri Madenciliği Uygulaması.* Yüksek Lisans Tezi. İstanbul: Marmara Üniversitesi Fen Bilimleri Enstitüsü.

Özdamar, E.Ö. (2002). *Veri Madenciliğinde Kullanılan Teknikler Ve Bir Uygulama.* Yüksek Lisans Tezi. İstanbul: Mimar Sinan Üniversitesi Fen Bilimleri Enstitüsü İstatistik Anabilim Dalı.

Özdoğan, Ö.G., Abul, O., Yazıcı, A. (2009). Paralel Veri Madenciliği Algoritmaları. *1. Ulusal Yüksek Başarım Ve Grid Konferansı.* (ss. 60-71), Düzenleyen Orta Doğu Teknik Üniversitesi. Ankara. 15-18 Nisan 2009.

Özkan, Y. (2008). *Veri Madenciliği Yöntemleri.* İstanbul: Papatya Yayıncılık.

Quinlan, J.R. (1986). Induction Of Decision Trees. *Journal Of Machine Learning.* (1): 81-106.

Shafer, J., Agrawal, R., Mehta, M. (1996). SPRINT: A Scalable Paralel Classifier For Data Mining. *22. International Conference On Very Large Database.* (ss. 544-555), Mumbai.

Silahtaroğlu, G. (2008). *Kavram Ve Algoritmaları İle Temel Veri Madenciliği.* İstanbul: Papatya Yayıncılık.

Solieman, O. (2006). *Data Mining In Sports: A Research Overview.* Master Project. California: University Of California Department Of Managment And Informatic System.

Sumathi, S., Sivanandam, S.N. (2006). *İntroduction To Data Mining And Its Applications. B*erlin: Springer.

Webb, G.I. (2003). Association Rules. *The Handbook Of Data Mining* (ss. 25-38). New Jersey: Lawrence Erlbaum Associates Publishers.

Witten, H.I, Frank, E. (2005). *Data Mining Practical Machine Learning And Techniques.* San Francisco: Morgan Kaufmann Publisher.

Yıldırım, S. (2003). *Tümevarım Öğrenme Tekniklerinden C4.5' in İncelenmesi.* Yüksek Lisans Tezi. İstanbul: İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü.

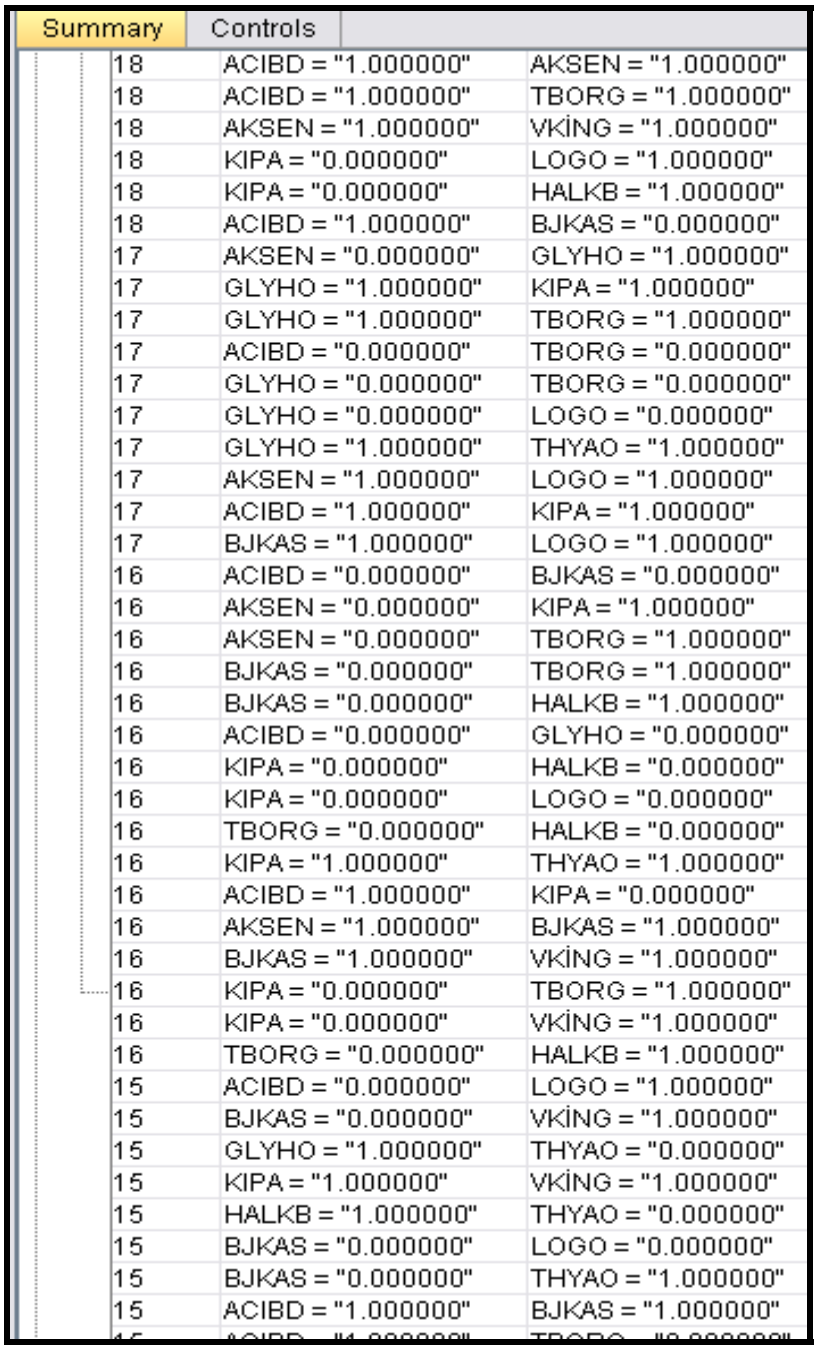
Zhang, T., Ramakrishnan, R., Livny, M. (1996). BIRCH: : An Efficient Data Clustering Method for Very Large Databases. *ACM International Conference On Management Of Data.* (ss. 103-114), USA.

EK

EK1 Orta Güçlükte Bağlantıya Sahip Değişkenler

Medium Links

Links	Field 1	Field 2
20	ACIBD = "0.000000"	VKİNG = "0.000000"
20	AKSEN = "0.000000"	HALKB = "0.000000"
20	BJKAS = "0.000000"	KIPA = "0.000000"
20	GLYHO = "0.000000"	VKİNG = "0.000000"
20	TBORG = "0.000000"	VKİNG = "0.000000"
20	THYAO = "1.000000"	VKİNG = "1.000000"
19	ACIBD = "0.000000"	THYAO = "0.000000"
19	AKSEN = "0.000000"	LOGO = "1.000000"
19	BJKAS = "0.000000"	THYAO = "0.000000"
19	BJKAS = "0.000000"	LOGO = "1.000000"
19	KIPA = "1.000000"	HALKB = "1.000000"
19	TBORG = "1.000000"	VKİNG = "1.000000"
19	AKSEN = "0.000000"	LOGO = "0.000000"
19	BJKAS = "0.000000"	VKİNG = "0.000000"
19	GLYHO = "0.000000"	KIPA = "0.000000"
19	HALKB = "0.000000"	VKİNG = "0.000000"
19	THYAO = "0.000000"	LOGO = "0.000000"
19	AKSEN = "1.000000"	HALKB = "1.000000"
19	BJKAS = "1.000000"	GLYHO = "1.000000"
19	ACIBD = "1.000000"	GLYHO = "1.000000"
19	ACIBD = "1.000000"	HALKB = "1.000000"
19	ACIBD = "1.000000"	THYAO = "1.000000"
18	ACIBD = "0.000000"	HALKB = "1.000000"
18	AKSEN = "0.000000"	HALKB = "1.000000"
18	KIPA = "1.000000"	LOGO = "1.000000"
18	ACIBD = "0.000000"	KIPA = "0.000000"
18	BJKAS = "0.000000"	TBORG = "0.000000"
18	BJKAS = "0.000000"	HALKB = "0.000000"
18	GLYHO = "0.000000"	HALKB = "0.000000"
18	GLYHO = "0.000000"	THYAO = "0.000000"
18	KIPA = "0.000000"	TBORG = "0.000000"
18	KIPA = "0.000000"	VKİNG = "0.000000"
18	TBORG = "0.000000"	LOGO = "0.000000"
18	HALKB = "0.000000"	THYAO = "0.000000"
18	HALKB = "0.000000"	LOGO = "0.000000"
18	VKİNG = "0.000000"	LOGO = "0.000000"
18	TBORG = "1.000000"	THYAO = "1.000000"



Summary | Controls

18	ACIBD = "1.000000"	AKSEN = "1.000000"
18	ACIBD = "1.000000"	TBORG = "1.000000"
18	AKSEN = "1.000000"	VKİNG = "1.000000"
18	KIPA = "0.000000"	LOGO = "1.000000"
18	KIPA = "0.000000"	HALKB = "1.000000"
18	ACIBD = "1.000000"	BJKAS = "0.000000"
17	AKSEN = "0.000000"	GLYHO = "1.000000"
17	GLYHO = "1.000000"	KIPA = "1.000000"
17	GLYHO = "1.000000"	TBORG = "1.000000"
17	ACIBD = "0.000000"	TBORG = "0.000000"
17	GLYHO = "0.000000"	TBORG = "0.000000"
17	GLYHO = "0.000000"	LOGO = "0.000000"
17	GLYHO = "1.000000"	THYAO = "1.000000"
17	AKSEN = "1.000000"	LOGO = "1.000000"
17	ACIBD = "1.000000"	KIPA = "1.000000"
17	BJKAS = "1.000000"	LOGO = "1.000000"
16	ACIBD = "0.000000"	BJKAS = "0.000000"
16	AKSEN = "0.000000"	KIPA = "1.000000"
16	AKSEN = "0.000000"	TBORG = "1.000000"
16	BJKAS = "0.000000"	TBORG = "1.000000"
16	BJKAS = "0.000000"	HALKB = "1.000000"
16	ACIBD = "0.000000"	GLYHO = "0.000000"
16	KIPA = "0.000000"	HALKB = "0.000000"
16	KIPA = "0.000000"	LOGO = "0.000000"
16	TBORG = "0.000000"	HALKB = "0.000000"
16	KIPA = "1.000000"	THYAO = "1.000000"
16	ACIBD = "1.000000"	KIPA = "0.000000"
16	AKSEN = "1.000000"	BJKAS = "1.000000"
16	BJKAS = "1.000000"	VKİNG = "1.000000"
16	KIPA = "0.000000"	TBORG = "1.000000"
16	KIPA = "0.000000"	VKİNG = "1.000000"
16	TBORG = "0.000000"	HALKB = "1.000000"
15	ACIBD = "0.000000"	LOGO = "1.000000"
15	BJKAS = "0.000000"	VKİNG = "1.000000"
15	GLYHO = "1.000000"	THYAO = "0.000000"
15	KIPA = "1.000000"	VKİNG = "1.000000"
15	HALKB = "1.000000"	THYAO = "0.000000"
15	BJKAS = "0.000000"	LOGO = "0.000000"
15	BJKAS = "0.000000"	THYAO = "1.000000"
15	ACIBD = "1.000000"	BJKAS = "1.000000"

15	ACIBD = "1.000000"	TBORG = "0.000000"
15	AKSEN = "1.000000"	GLYHO = "1.000000"
15	AKSEN = "1.000000"	THYAO = "1.000000"
15	GLYHO = "1.000000"	KIPA = "0.000000"
15	GLYHO = "1.000000"	TBORG = "0.000000"
15	ACIBD = "1.000000"	AKSEN = "0.000000"
14	BJKAS = "0.000000"	KIPA = "1.000000"
14	KIPA = "1.000000"	TBORG = "1.000000"
14	THYAO = "0.000000"	LOGO = "1.000000"
14	ACIBD = "0.000000"	LOGO = "0.000000"
14	AKSEN = "0.000000"	THYAO = "1.000000"
14	BJKAS = "1.000000"	KIPA = "1.000000"
14	BJKAS = "1.000000"	TBORG = "0.000000"
14	BJKAS = "1.000000"	THYAO = "1.000000"
14	KIPA = "1.000000"	TBORG = "0.000000"
14	ACIBD = "1.000000"	GLYHO = "0.000000"
14	ACIBD = "1.000000"	HALKB = "0.000000"
14	ACIBD = "1.000000"	THYAO = "0.000000"
14	AKSEN = "1.000000"	TBORG = "1.000000"
14	BJKAS = "1.000000"	KIPA = "0.000000"
14	BJKAS = "1.000000"	TBORG = "1.000000"
14	BJKAS = "1.000000"	THYAO = "0.000000"
14	TBORG = "0.000000"	LOGO = "1.000000"
13	ACIBD = "0.000000"	GLYHO = "1.000000"
13	AKSEN = "0.000000"	VKİNG = "1.000000"
13	BJKAS = "0.000000"	GLYHO = "1.000000"
13	ACIBD = "0.000000"	BJKAS = "1.000000"
13	KIPA = "1.000000"	VKİNG = "0.000000"
13	GLYHO = "0.000000"	TBORG = "1.000000"
13	GLYHO = "0.000000"	LOGO = "1.000000"
13	VKİNG = "0.000000"	LOGO = "1.000000"
13	KIPA = "0.000000"	THYAO = "1.000000"
12	ACIBD = "0.000000"	TBORG = "1.000000"
12	KIPA = "1.000000"	THYAO = "0.000000"
12	TBORG = "1.000000"	THYAO = "0.000000"
12	AKSEN = "0.000000"	BJKAS = "1.000000"
12	BJKAS = "1.000000"	VKİNG = "0.000000"
12	GLYHO = "0.000000"	THYAO = "1.000000"
12	ACIBD = "1.000000"	LOGO = "0.000000"

12	AKSEN = "1.000000"	KIPA = "0.000000"
12	AKSEN = "1.000000"	KIPA = "1.000000"
12	TBORG = "0.000000"	VKİNG = "1.000000"
12	HALKB = "1.000000"	VKİNG = "0.000000"
12	GLYHO = "0.000000"	HALKB = "1.000000"
11	ACIBD = "0.000000"	KIPA = "1.000000"
11	THYAO = "0.000000"	VKİNG = "1.000000"
11	ACIBD = "0.000000"	HALKB = "0.000000"
11	BJKAS = "1.000000"	LOGO = "0.000000"
11	GLYHO = "0.000000"	KIPA = "1.000000"
11	TBORG = "0.000000"	THYAO = "1.000000"
11	TBORG = "1.000000"	VKİNG = "0.000000"
11	ACIBD = "1.000000"	VKİNG = "0.000000"
11	GLYHO = "1.000000"	VKİNG = "0.000000"
10	ACIBD = "0.000000"	THYAO = "1.000000"
10	KIPA = "1.000000"	LOGO = "0.000000"
10	GLYHO = "0.000000"	VKİNG = "1.000000"
10	AKSEN = "1.000000"	TBORG = "0.000000"