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

VERİ MADENCİLİĞİ MODELLERİ VE ÖRNEK BİR UYGULAMA

Zahide DENİZLİ

Danışman
Prof. Dr. Levent ŞENYAY

İZMİR-2019

YÜKSEK LİSANS
TEZ ONAY SAYFASI

Üniversite	: Dokuz Eylül Üniversitesi
Enstitü	: Sosyal Bilimler Enstitüsü
Adı ve Soyadı	: Zahide DENİZLİ
Öğrenci No	: 2007800342
Tez Başlığı	: Veri Madenciliği Modelleri ve Örnek Bir Uygulama
Savunma Tarihi	: 17/07/2019
Danışmanı	: Prof.Dr.Levent ŞENYAY

JÜRİ ÜYELERİ

Ünvanı, Adı, Soyadı	Üniversitesi	İmza
Prof.Dr.Levent ŞENYAY	-Dokuz Eylül Üniversitesi	.....
Prof.Dr.Cenk ÖZLER	-Dokuz Eylül Üniversitesi	.....
Prof.Dr.Şaban EREN	- Yaşar Üniversitesi	.....

Zahide DENİZLİ tarafından hazırlanmış ve sunulmuş olan bu tez savunmada başarılı bulunarak oy birliği(X) / oy çokluğu( ) ile kabul edilmiştir.

Prof. Dr. Metin ARIKAN
Müdür

YEMİN METNİ

Yüksek Lisans Tezi olarak sunduğum **"Veri Madenciliği Modelleri Ve Örnek Bir Uygulama**" adlı çalışmanın, tarafımdan, akademik kurallara ve etik değerlere uygun olarak yazıldığını ve yararlandığım eserlerin kaynakçada gösterilenlerden oluştuğunu, bunlara atıf yapılarak yararlanılmış olduğunu belirtir ve bunu onurumla doğrularım.

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

Zahide DENİZLİ

İmza

ÖZET

Yüksek Lisans Tezi

VERİ MADENCİLİĞİ MODELLERİ VE ÖRNEK BİR UYGULAMA
Zahide DENİZLİ

Dokuz Eylül Üniversitesi
Sosyal Bilimler Enstitüsü
Ekonometri Ana Bilim Dalı
Ekonometri Programı

Gelişen teknoloji ile beraber bilgiye ulaşmak her geçen gün daha kolay hal almaktadır. Bu bilgileri yerinde kullanmak ve kullanışlı hale getirmekte gittikçe önem taşımaktadır. Veri tabanlarında bulunan büyük boyutlardaki ham verilerden kullanılabilir bilgiye ulaşmak veri madenciliği sayesinde mümkündür. Veri madenciliği ile çok büyük miktardaki veriler arasındaki ilişkilerden anlamlı bilgiler ortaya çıkarılmaktadır.

Bu çalışma içeriğinde, ham verilerden faydalı bilgi elde etme aşamaları olan veri madenciliği kavramı anlatılmaya çalışılmıştır. Bu ham veriyi işlemede yaygın olarak kullanılan CRISP-DM aşamaları açıklanmıştır. Veri madenciliği algoritmalarının tanımlayıcı ve tahmin edici olarak ikiye ayrılmasından bahsedilmiştir. Ayrıca sınıflandırma, kümeleme ve birliktelik kuralları algoritmalarıyla kurulan modellerin aşmaları ayrıntılı olarak anlatılmıştır.

Bu anlatımlar doğrultusunda uygulamada Türkiye'de kitap satışı yapan internet sitelerinden alınan en çok satan kitap listelerindeki kitapları türlerine göre sınıflandırma yaparken hangi faktörlerin etkili olduğu belirlenmeye çalışılmıştır. Çalışmada, veri madenciliği sınıflandırma algoritmalarından biri olan K- En Yakın Komşuluk (k-nearest neighbor, k-nn) algoritması en başarılı sonucu verdiği için seçilmiştir.

Anahtar Kelimeler: Veri Madenciliği, Sınıflandırma, k-nn Algoritması, Kitap, En Çok Satan, CRISP-DM, ROC Eğrisi.

ABSTRACT

Master's Thesis

DATA MINING MODELS AND A SAMPLE APPLICATION

Zahide DENİZLİ

Dokuz Eylül University

Graduate School of Social Sciences

Department of Econometrics

Econometrics Program

To reach the information is getting easier than past through the improving technologhy day by day. To use decently and make functional those information is also getting importance. It is possible to reach usable information from the vast amount of raw data, which is found in the data sets, through data mining. The meaningful information is revealed from the relations of the vast amount of data sets through data mining.

In this study, data mining concept, which means extracting useful information from raw data, has been elucidated. Besides, CRISP-DM phases, which is commonly used in raw data processing has been explained. Moreover, it has been mentioned that two types of data mining algorithms as descriptive and predictive. In addition, the phases of models built-up by the rules of classification, clustering and association algorithms has been explained in detail.

In line with those expressions, it has been determined that which factors affects to classification of the book sale web sites' bestsellers of Turkey according to their types. In this study, one of the data mining classification algorithm, k-nearest neighbor – k-nn algorithm has been selected due to its best result.

Keywords: Data Mining, Classification, the k-nearest neighbor (k-nn) Algorithm, Book, Best-Selling, CRISP-DM, ROC

VERİ MADENCİLİĞİ MODELLERİ VE ÖRNEK BİR UYGULAMA

İÇİNDEKİLER

KISALTMALAR

AUC	Area Under this Curve
BIRCH	Balance Iterative Reducing and Clustering Using Hierarcihies
CART	Classification And Regression Trees
CHAID	Chi-square Automatic Interaction Detector
CRISP-DM	Cross - Industry St andard Process
CRM	Costumer Relationship Management
CURE	Clustering Using Representattives
DBSCAN	Density-Based Spatial Clustering Method Based on Connected
k-nn	K-nearest neighbor
OLTP	Online Transaction Processing
OLAP	Online Analytical Processing
PAM	Partitioning Around Medoids
ROC	Receiver Operating Characteristic
STING	Statistical Information Grid
VM	Veri Madenciliği
VTBK	Veri Tabanlarında Bilgi Keşfi
YSA	Yapay Sinir Ağları

TABLOLAR LİSTESİ

ŞEKİLLER LİSTESİ

GİRİŞ

Veri madenciliği günümüzde birçok organizasyonun stratejik kararlar alırken kullandığı verilerden anlamlı bilgiler çıkarma yöntemlerinin uygulandığı alandır. Çağımız koşullarında bilgisayar teknolojileri ile birlikte verileri depolamak ve verilere ulaşmak çok kolay bir hal almıştır. Bu kolayca depolanıp ulaşılabilinen verilerden anlamlı bilgiler çıkarmak veri madenciliği sayesinde olmaktadır. Bu bilgileri açığa çıkarmak ve bu bilgilerden faydalanmak için çok sayıda veri madenciliği algoritması oluşturulmuştur.

Teknolojinin hayatımıza yer etmesi ile birlikte gittikçe dijital dünyaya dönüşen sistemin kitap okuma oranlarını çok fazla etkilediği görülmektedir. Ayrıca gelişen teknoloji ile birlikte bilgiye ulaşmak kolay olsa da doğru ve güvenilir bilgiye ulaşmak gittikçe zorlaşmaktadır. Birçok bilgi kaynağı olmasına rağmen günümüzde halen en etkili ve doğru bilgi kaynağı kitaptır. Bu sebeple kitaplardan yol çıkarak en çok satan kitapların türlerinin sınıflama algoritmaları ile değerlendirilmesi yapılmaya çalışılmıştır.

Bu araştırmanın ilk bölümünde, veri madenciliği kavramı anlatılmış, nerelerde kullanıldığı ve tarihçesine değinilmiştir. Ayrıca veri tabanında bulunan verilerin anlamlı bilgilere dönüştürülmeden önce verenin temizlenmesi, düzenlenmesi, indirgenmesi vb. uygulamalara değinilmiştir. Ek olarak CRISP-DM aşamaları açıklanmıştır

Araştırmanın ikinci bölümünde, veri madenciliği algoritmaları anlatılmıştır. Tahmin edici ve tanımlayıcı olarak ikiye ayrılan algoritmalara ayrıntılı olarak yer verilmiştir. Ayrıca sınıflandırma, kümeleme ve birliktelik analizlerinin modelleri ayrı ayrı anlatılmıştır.

Araştırmamanın üçüncü bölümünde, birinci ve ikinci bölümde verilen bilgiler doğrultusunda örnek bir uygulama yapılmıştır. Bu çalışma için Türkiye'deki 21 internet sitesinin en çok satan kitap listelerinde bulunan kitap türlerinin sınıflandırma algoritmaları kullanılarak model kurulmaya çalışılmıştır. Kurulan model doğrultusundaki sonuçlar yorumlanmıştır.

BİRİNCİ BÖLÜM
VERİ MADENCİLİĞİ SÜRECİ

1.1 VERİ MADENCİLİĞİ

Gelişen bilgisayar teknolojileri ile birlikte her geçen gün kullanılmakta olan veri miktarları da hızla artmakta; hızla artan veri miktarları da çoğalan bu verilerin analiz edilmesini güçleştirmektedir. Veri madenciliğinin ortaya çıkışı veri yığınlarının geniş yer kaplamasına ve büyük miktardaki verilerin yararlı bilgilere dönüştürülmesi ihtiyacına dayanmaktadır (Han ve Kamber, 2006:1). Bu sebeple büyük veri tabanlarından istenilen anlamlı, kullanılabilir ve faydalı bilgiye erişmek, yeni bir veri işleme sürecinin doğmasına sebep olmuştur. Veriler çeşitli istatiksel metotlarla analiz edilerek kurumların karar verme sürecine ve yeni stratejiler geliştirmesine katkı sağlamaktır.

Ham haldeki işlenmemiş kayıtlara veri (Data) denilmektedir. Ham veri halindeki kayıtlar düzenlenmemiş, gerekli ilişkilendirme ve anlamlandırma işlemi yapılmamış haldedir. Veri tabanı yönetim sistemlerinin keşfi ve gelişen teknolojiyle birlikte elde edilen verilerin artması işletmelerde büyük miktarda veriler toplanmasını ve depolanmasını sağlamıştır.

Bireyler bilgiye ulaşmak için Veri Madenciliği (data mining) ile çeşitli uygulamalar kullanarak, veri içerisinde gizli yatkınlık ve örüntüleri saptayabilir. İnsanların hızla büyüyen dijital veri hacimlerinden faydalı bilgiler (bilgi) elde etmelerine yardımcı olmak için yeni nesil bir hesaplama teorileri ve araçlarına acil olarak ihtiyaç duyulmaktadır. Bu teoriler ve araçlar, veri tabanlarında ortaya çıkan bilgi keşfi alanının konusudur (Fayyad ve diğerleri, 1996:37).

1.2 VERİ MADENCİLİĞİ TANIMLARI

Veri madenciliği, büyük miktardaki mevcut verilerden birbirleriyle anlamlı ilişki ve kurallar ortaya çıkarmak için uygun verilerin seçilmesi, birleştirilmesi, dönüştürülmesi, sınıflandırılması ve modellenmesi aşamalarından oluşur. Veri madenciliğinin birçok değişik tanımı yapılmaktadır. Bunlardan bazıları;

“Veri Madenciliği, veri tabanında bilgi keşfi işleminde, kabul edilebilir hesaplama etkinliği sınırlamaları altında, veriler üzerinde belirli bir model numaralandırması üreten veri analizi ve keşif algoritmalarının uygulanmasından oluşan bir adımdır (Fayyad ve diğerleri, 1996:54). ”

“Veri madenciliği, geniş veri tabanlarından bilgi çıkarma konusunu ele almak için makine öğrenmesi, örüntü tanıma, istatistik, veri tabanları ve görselleştirme tekniklerini bir araya getiren disiplinler arası bir alandır. (Cabena ve diğerleri, 1999:1).”

“Veri madenciliği, tek seferlik bir süreç değil, her seferinden bakılan perspektife göre değişen ve devam eden bir süreçtir. Veri madenciliğinin özü beklenmedik olanı keşfetme çabasıdır ve beklenmeyen, doğası gereği beklenmedik şekillerde ortaya çıkabilir (Hand, 1999:16).”

“Veri madenciliği, verileri işleyerek bilgiye ve bilgiyi yorumlayarak bazen bir karara, bazen de yeniden veri toplama aşamasına veya farklı karar verme tekniklerininim uygulanması aşamalarıdır(Oğuzlar, 2004:4).”

“Veri madenciliği, beklenmeyen ilişkileri bulmak ve verileri veri sahibi için hem anlaşılabilir hem de yararlı olan yeni yollarla özetlemek için genellikle büyük miktarda gözlemsel veri setlerinin analizidir. (Larose, 2005:11).”

“Veri madenciliği, veri tabanı sahibi için açık ve faydalı sonuçlar elde etmek amacıyla ilk olarak bilinmeyen düzenleri veya ilişkileri keşfetmek için büyük miktardaki verilerin seçilmesi, araştırılması ve modellenmesi sürecidir. (Giudici, 2003:2).”

“Veri madenciliği, veri tabanlarında, veri ambarlarında veya diğer bilgi depolarında depolanan çok büyük miktarda veriden ilginç bilgileri keşfetme sürecidir (Han ve Kamber, 2006:7).”

“Veri madenciliği yöntemleri, belirli varsayımların koşulları altında her zaman gözle görülür olmayan (ve dolayısıyla kolayca tanımlanamayan) kalıpları ve gizli ilişkileri tanımlamaya çalışır. (Gorunescu, 2011:5).”

Kısaca özetlemek gerekirse ham halde bulunan büyük miktardaki veriler içerisinden faydalı ve anlamlı bilgiyi çıkarmak için yapılan değiştirme, dönüştürme ve modelleme aşamalarının tümüne veri madenciliği denir.

1.3 VERİ MADENCİLİĞİNİN TARİHÇESİ VE LİTERATÜR TARAMASI

Geçmişten günümüze veriler her zaman yorumlanmış, bilgiye ulaşılmak istenmiş ve bunun için de çeşitli çalışmalar yapılmıştır. Bu sayede bilgi dünden bugüne taşınır hale gelmiştir.

İlk sayısal bilgisayar olan ENIAC (Electrical Numerical Integrator And Calculator)'ın ortaya çıkışı ile birlikte, matematikçiler 1950'li yıllarda mantık ve bilgisayar bilimleri alanlarında çalışarak, yapay zekâ (Artificial İntelligence) ve makine öğrenme (Machine Learning) ortaya çıkmıştır.

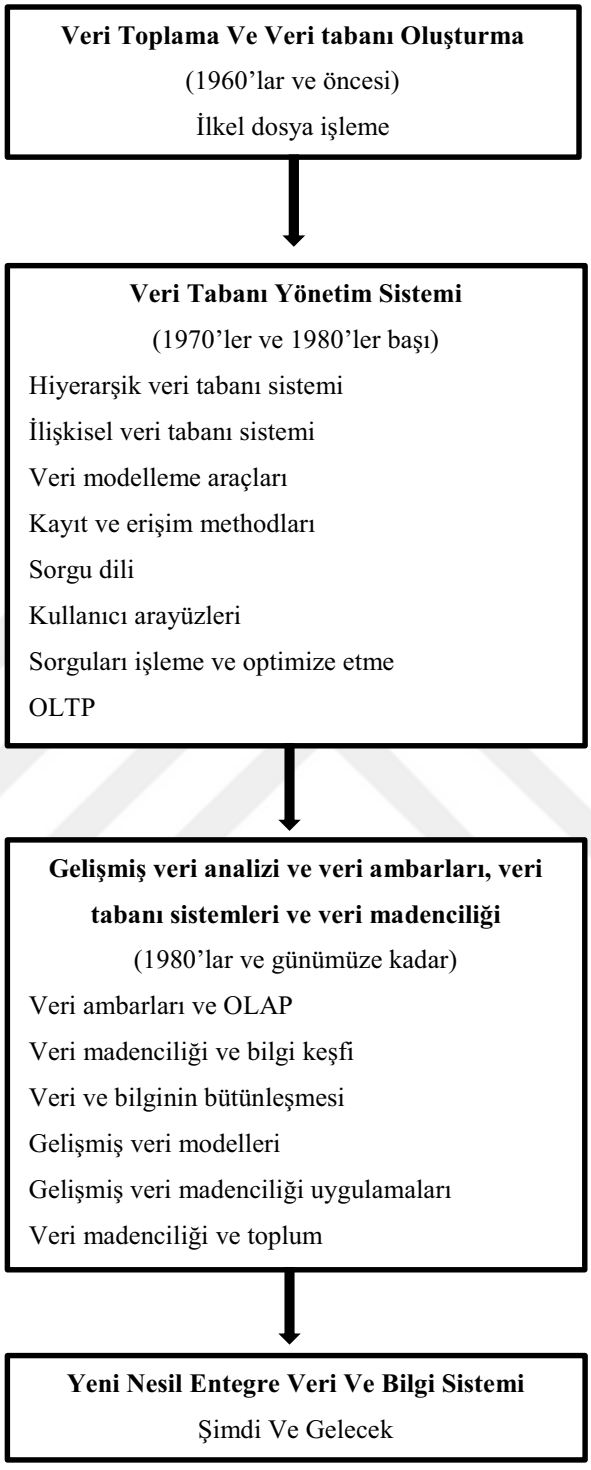
1960'lı yıllar ve öncesine dayanan veri bilimi 1970'li yıllarla birlikte veri tabanı sistemlerini geliştirmeye başlamıştır. Devamında algoritmalar geliştirilmiş modeller kurulmuş ve ilişkisel veri tabanı sistemleri oluşturulmuştur.

1980'lı yıllar ile beraber veri tabanında bilgi keşfinin ilk girişimleri başlamış ve büyük veri tabanları için veri ambarları geliştirilmiştir. Aynı süreç içerisinde yeni teknolojilerle birlikte veri madenciliği hızlı bir şekilde gelişerek yaygın olarak kullanılan standart bir işin parçası olmuştur. Veri ve bilgi bütünleşerek gelişmiş veri madenciliği uygulamaları uygulanmaya başlamıştır. Son yıllarda bilgisayar teknolojisinin de gelişmesi ile beraber güçlü ve büyük depolama hafızasına sahip bilgisayarların üretilmesi veri depolama, işlem yönetimi, bilgi alma ve veri analizi için çok sayıda veri tabanı ve bilgi deposu sunar.

Zaman içinde bilim ve teknolojinin gelişimi ile birlikte, bilgiye ulaşmak yorumlamak, kullanmak, değiştirmek, dönüştürmek tarihteki en yüksek noktaya ulaşmıştır. İnsanlık 21.yüzyılda artık bilim ve teknoloji sisteminin yol göstericiliğinde ilerlemektedir. Veri madenciliği de bu yol gösterici sistemlerin en önemlilerinden biridir.

Veri madenciliğinin tarihsel gelişimi Tablo 1 de özet halinde gösterilmektedir

Tablo 1: Veri Madenciliğinin Gelişimi.



Kaynak: Han ve Kamber, 2006:2.

Veri madenciliği haberleşme, astronomi, tıp, perakende, ekonomi sektörlerinde pazarlama yönetimi, güven yönetimi, piyasa analizi, sahtekârlık tespiti ve müşteriyi

elde tutmaya, üretim kontrolüne, bilim araştırmalarına müşteri ilişkileri ve risk araştırma vb. ihtiyaçlar doğrultusunda kullanılmaktadır. Tablo 2' de veri madenciliği kullanım alanları gösterilmektedir.

Tablo 2: Veri Madenciliği Kullanım Alanları

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

Kaynak: Gorunescu, 2011:41

1980 yılından itibaren veri miktarlarındaki hızlı artış ile beraber veri analizlerinin daha rahat ve kolay yapılmasını sağlayan veri madenciliği, istatistiksel yöntemleri de içinde barındırdığı gibi istatistiksel yöntemlerin yetersiz kaldığı (metin, ses vb.) bazı konularda da analiz yapabilmektedir. Yıllardan beri istatistikçiler veri tabanları konusunda yavaş bir şekilde ilerlerken istatistiksel açıdan önemli ilişkiler aramaktadırlar. Veri Madenciliği, bu süreci otomatik olarak gerçekleştirmektedir. İstatistiksel analiz ve veri madenciliğinin karşılaştırması ve farklılaştığı noktalar aşağıda Tablo 3'de gösterilmiştir.

Tablo 3: İstatistiksel Analiz Ve Veri Madenciliği

İstatistiksel Analiz	Veri Madenciliği
İstatistiksel analiz de genellikle hipotez kurulur	Veri madenciliğinde hipoteze kurulmaz
İstatistiksel analiz kendi eşitliğini geliştirmelidir.	Veri madenciliği algoritmaları eşitlikleri otomatik olarak geliştirir.
İstatistiksel analizler sayısal veriler kullanılarak yapılır	Veri madenciliğinde sayısal verilere ek olarak farklı türlerde veriler (örneğin metin, ses) de kullanır.
İstatistiksel analiz kirli veriyi analiz esnasında bulur ve filtre ederler.	Veri madenciliği temiz veri ile yapılır.
İstatistikçiler sonuçlarını kendileri bulur ve yorumlar	Veri madenciliğinin sonuçlarını yorumlamak çok zordur. Veri madenciliği sonuçlarını analiz etmede ve yorumlamada istatistikçiye ihtiyaç duyulmaktadır.

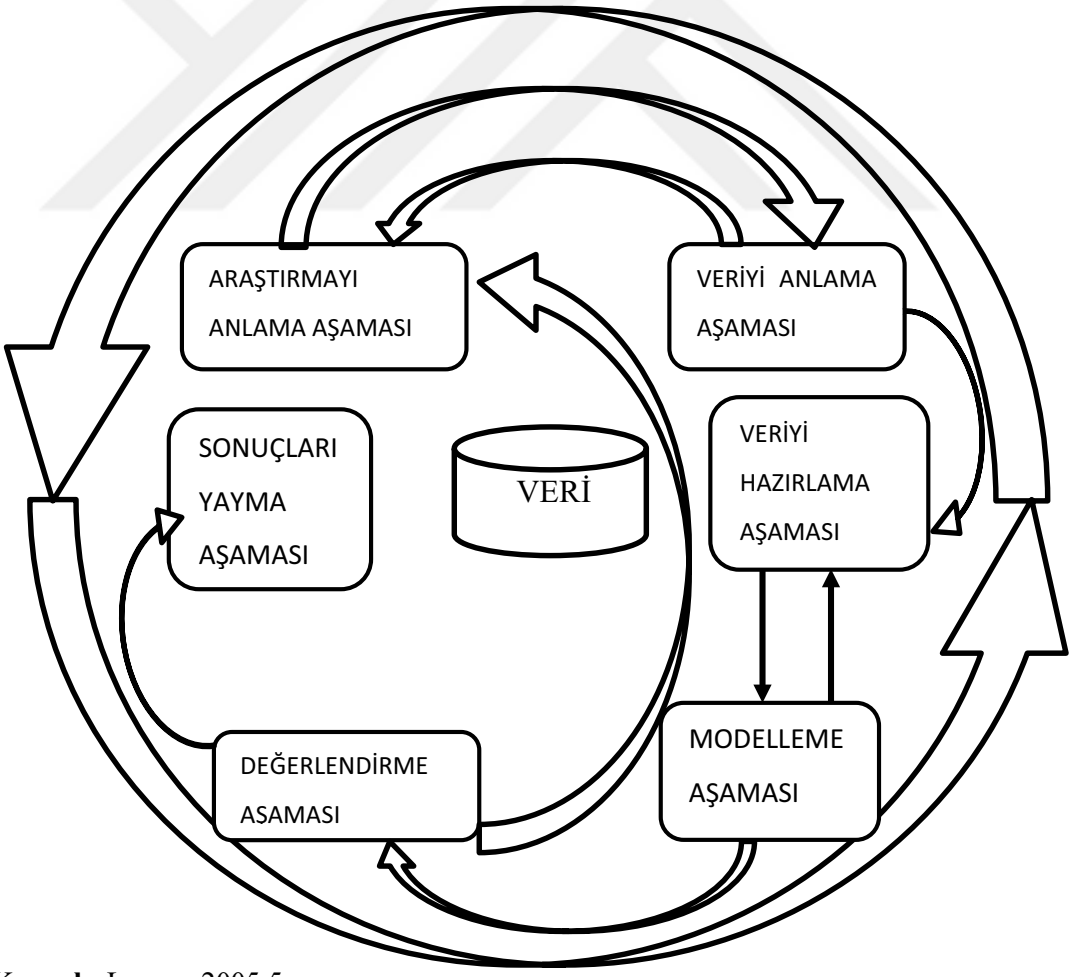
Kaynak: Moss and Atre, 2003:38

1.4 VERİ MADENCİLİĞİ SÜRECİ

VM süreci, büyük veri tabanlarından önemli, kullanılabilir, önceden bilinmeyen bilgiye çeşitli yöntemler uygulayarak erişmekteki adımlar olarak tanımlanabilir.

Veri madenciliği aşamaları uygulanırken en çok kullanılan model CRISP-DM (Cross-Industry Standard Process for Data Mining- Çapraz Endüstri Standart Prosedürü) modelidir. CRISP-DM süreç modeli, Daimler Chrysler AG, SPSS ve NCR gibi lider veri madenciliği kullanıcıları ve tedarikçilerinden oluşan bir konsorsiyum tarafından geliştirilmiştir (Göral, 2007:17). CRISP-DM'e göre, verilen veri madenciliği projesi Şekil 1'de gösterilen altı basamaklı bir yaşam döngüsüne sahiptir.

Şekil 1: CRISP-DM Süreç Modeli



Kaynak: Larose, 2005:5

Döngünün aşamaları;

1. Aşama: Araştırmanın Tanımlanması (Business understanding)

2. Aşama: Veriyi tanımlama (Data understanding)

3. Aşama: Veri hazırlama (Data preparation)

4. Aşama: Modelleme (Modeling)

5. Aşama: Değerlendirme (Evaluation)

6. Aşama: Uygulama ve Sonuçları Yayma (Deployment)

biçiminde şekillenmektedir (CRISP-DM Special Interest Group 2004).

Veri madenciliği süreci içerisinde karar vericiye katkı sağlayan karar destek sistemleri, değişik kaynaklardan topladığı bilgileri düzenleyen, kararı modelleyen, bilgileri analiz eden ve değerlendirme sonuçlarını sunan bilgisayar tabanlı sistemlerdir.

Karar destek sistemleri, karar vericinin daha önce öngörülmemiş bir bilgiye ihtiyaç duyduğu sırada kullanacağı verilerin toplanmasını, saklanması, analiz edilmesini, ulaşılabilmesi, planlamalarda, taktiklerin belirlenmesinde ve kritik kararlarının alınmasında kullanılmak üzere oluşturulan sistematik bir oluşumdur.

Minimum risk için tek yol bilgiye dayalı yönetimi öngören karar destek sistemleridir. İşletmelerin daha dinamik kararlar alması için oluşturulan karar destek sistemlerinde ihtiyaç duyulan eğilimlerin ve yöntem modellerinin ortaya çıkarılması mümkün olmaktadır.

Bir veri tabanı sistemi ise, birbiri ile ilişkili verilerin toplamını içeren, veriye ulaşmayı sağlayarak veriyi yönetmeye yardımcı olan yazılım programlarının hepsidir. Diğer deyişle veri tabanına, ilişkili verilerin toplandığı, kullanım amacına göre düzenlenmiş verilerin mantıksal ve fiziksel olarak tanımlandığı bilgi depoları da denilebilir.

Veri tabanı sistemleri yazılımları bilgisayar sistemlerinin önemli bir bileşeni olarak değerlendirilir. Veri tabanı, işletmeye ait bilgilerin bulunduğu bir ortamdır. Veri tabanı sistemleri, veri tabanlarının düzenli bir şekilde tutulduğu ve verilerin çeşitli yazılımlar aracılığıyla modellenerek kullanıldığı bir ortamdır.

Veri tabanları direk olarak veri madenciliğinde kullanılmak için uygun değildir. Temel olarak veri madenciliği çalışmaları için veri ve veri tabanı gerekmektedir. Belirli bir zamanı kapsayan, yapılacak çalışmanın içeriğine göre

konulara ayrılarak düzenlenmiş, birleştirilmiş ve temizlenmiş veri tabanlarına veri ambarları denilir.

Veri Ambarları, veri madenciliği sürecinin gerçekleştirildiği veriyi sağlayan özel bir veri tabanıdır. Pek çok farklı kaynaktan elde edilen veriyi aynı çatı altında analiz etme imkânı tanır.

OLTP (Online Transaction Processing), organizasyonlarda günlük hareket işlemlerin yapıldığı işlemsel veritabanı sistemleridir. Ayrıntılı bilgi içerir. Veriye kolaylıkla erişilebilir ve üzerinde oynama yapılabilir.

Veri ambarları üzerinde, çok büyük boyutlu analizler esnasında oluşan stratejik süreçler hakkında karar vermek amacıyla yapılan veri analizi işlemlerine OLAP (Online Analytical Processing) denilir.

OLAP, işlemsel sistemlerin ve ilişkisel veri tabanlarının tamamlanmasını sağlar. Veriler sadece istenildiği zaman değil her zaman toplanır. Bu sayede istatistiksel ölçümler çok daha sağlıklı yapılır. Büyük veri tabanlarında basit ilişkiler keşfedilmek isteniyorsa OLAP kullanılmalıdır. Son yıllardaki bilgi teknolojilerindeki gelişmeler OLAP uygulamalarını yaygınlaştırmıştır.

1.5 VERİ TABANLARINDA BİLGİ KEŞFİ

Veri Tabanlarında Bilgi Keşfi, veriden yararlı bilgiye ulaşılması sürecini betimlemektedir. Üzerinde inceleme yapılan işin ve verilerin özelliklerinin bilinmemesi durumunda, hiç bir veri madenciliği algoritmasının faydalı olması mümkün değildir.

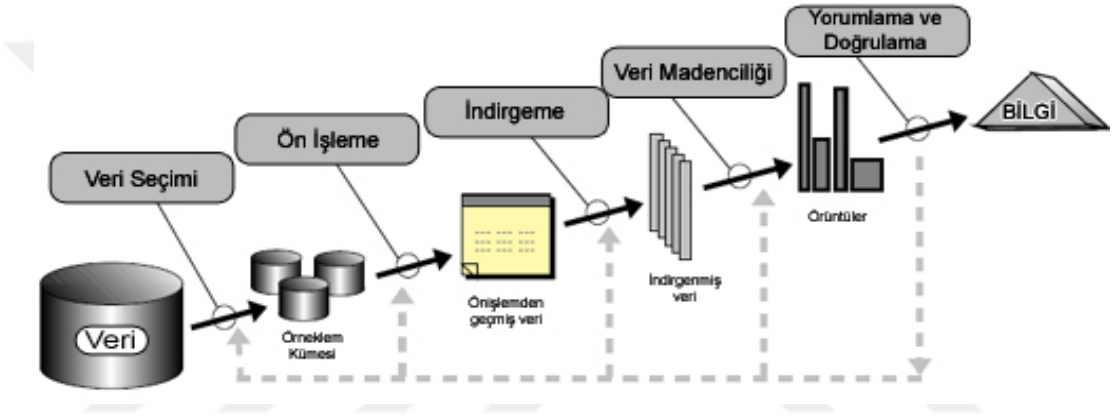
VTBK süreci, veri tabanlarını kullanarak veri tabanlarında istenilen seçim, ön işleme, alt örnekleme, dönüşüm, örüntülerin açığa çıkarılması için VM yöntemlerinin uygulanması ve açığa çıkarılan örüntülerin tanımlanması için VM ürünlerinin yorumlanması süreçlerini içermektedir. VTBK makine öğrenimi, örüntü tanıma, veri tabanları, istatistik, yapay zekâ, uzman sistemler, veri görselleştirme ve yüksek performanslı hesaplama gibi araştırma alanlarının kesişimi olarak gelişmiş ve gelişimine devam etmektedir. Verilerin nasıl depolanıp ulaşılabileceğine, algoritmaların çok büyük veri setlerini nasıl ölçekleyebileceğine ve hala etkin olarak çalışmalarına, sonuçların nasıl yorumlanabileceğine ve görselleştirilebileceğine ve

bütün insan-makine interaksiyonunun kullanışlı olarak nasıl modellenip, desteklenebileceğine olmak üzere veriden bilginin keşfinin tüm süreçleri üzerine odaklanır (Koyuncugil, 2006:39).

VTBK süreci interaktif, tekrar eden ve subjektif insan yargısının kullanılmasını gerektiren adımların birleşmesinden oluşmaktadır.

Veri ambarcılığı, veri kümelerine VTBK aşaması için veri temizleme ve veri erişimi konularında yardımcı olmaktadır. VTBK süreci şekil 3 de gösterilmektedir.

Şekil 2: Veri Tabanında Bilgi Keşif Süreci Aşamaları



Kaynak: Fayyad ve Diğerleri, 1996:41

Araştırma probleminin betimlenmesi sürecinde araştırmanın amacını, hali hazırdaki durumun değerlendirilmesini, veri madenciliğinin amaçlarını ve proje planlama aşamalarının belirlenmesini içermektedir. Bu safha veri madenciliği sürecinin taleplerinin ve yöntemlerinin belirlendiği aşamadır.

Problemin tanımlanmasında amaç, problem üzerine yoğunlaşılmış ve açık bir dille ifade edilmiş olmalı, elde edilecek sonuçların başarı düzeylerinin nasıl ölçüleceği de tanımlanmalıdır. (Akpınar, 2000:8).

Veri toplama, tanımlanan problem için gerekli olduğu düşünülen verilerin ve bu verilerin toplanacağı veri kaynaklarının belirlenmesi adımıdır. Daha sonra benzer verileri bir araya toplama, veri özelliklerini tanımlama, gizli bilgileri kümeleme ile sürece devam etmektedir.

Veriler nitel ve nicel veriler olarak iki şekilde gruplanır. Sayısal (nümerik) veriler nicel olarak değerlendirilirken, nitel veriler ise kategorik olarak

tanımlanmaktadır. Nitel veriler nominal (sırasız nitel) ve ordinal (sıralı nitel) veriler olmak üzere ikiye ayrılırlar. Değişkenlerin birbirine benzemediği ve bu farklılığında herhangi bir üstünlük yaratmadığı durumdaki değişkenler nominal veriler olarak değerlendirilir. Nominal veriler hem sayısal hem de karakter dizileri şeklinde olabilir (Şekeroğlu, 2010:25). Verilerin belli bir ölçüte göre büyükten küçüğe veya küçükten büyüğe sıralanması ordinal verileri oluşturmaktadır. Nicel veriler sayısal verilerdir. Aralık ve oran veriler olarak kullanılabilir.

Veriler farklı kaynaklardan elde edilmiş olabilir. Farklı kaynaklardan ulaşılan verilerin bir araya getirildiğinde belirli bir sistem içinde birleştirilmesi gerekmektedir. Verilerin içerisinde çok sayıda değişken yer almaktadır ve bu değişkenlerin saklanma şekilleri birbirlerinden farklılık gösterebilir. Belirli bir standarda dönüştürülmeyen veriler ileride büyük uyumsuzlukların yaşanmasına neden olabilmektedir. Veri madenciliğinde kişilerin işlevi, problemin çözümünü bulmak değil, tersine verileri kurulan modeller ile kendiliğinden bu sonuçlara ulaşmasına katkı sağlamaktır.

Verilerin hazırlanması aşaması kendi içerisinde;

- Birleştirme ve Temizleme,
- İndirgeme
- Dönüştürme

adımlarından meydana gelmektedir.

Veri birleştirme çok sık uygulanan veri madenciliği tekniklerinden biridir. Bir bilgi aynı anlama gelebilecek birden fazla kayıtta gereksiz yere tekrarlanmış olabilir. Örneğin, bir veri tabanında kayıtlı kişilerin hem doğum tarihleri hem de yaşları tutulmuşsa bunların ikisi de aynı anlama geldiğinden fazla kayıt oluşturur. Bazen elimizdeki değişkenlerin birleşmesi ve tek bir değişken gibi ele alınması gerekebilir. Bu hem veri madenciliği çalışması esnasındaki kullanılan zamanı azaltacak hem de elde edilecek sonuçların güvenirliğini ve kalitesini arttıracaktır.

Verilerin temizlenmesi adımında, gürültülü, tutarsız, hatalı, eksik ve aşırı uçta bulunan verilerin etkileri ayıklanmaya ve düzeltilmeye çalışılır.

Eksik veriler ile karşılaşıldığında yapılabilecek çözümler;

- Eksik veriler veri kümesinden silinebilir.
- Eksik veriler elle tek tek doldurulabilir.
- Tüm eksik verilere aynı bilgi girilebilir.

- Eksik olan verilere tüm verilerin ortalaması verilebilir.
- Diğer değişkenler yardımı ile eksik verilerin regresyon vb. tekniklerle tahmin edilebilir.

Bunun dışında verilerin temizlenmesi aşamasında yanlış veya tutarsız girilmiş veriler, gürültülü veriler ve aşırı uçlarda olan veriler için kullanılan yöntemler (Oğuzlar 2004:30):

- Kutulama yöntemiyle gürültünün temizlenmesi
- Sınırlar yardımıyla gürültünün temizlenmesi
- Kümeleme yöntemiyle gürültünün temizlenmesi
- Uçta bulunan veriler tahmin yöntemleri aracılığıyla, mevcut veriler kullanılarak düzgünleştirilebilir.

Veri indirgeme, büyük boyutlu veri kümesinden daha küçük boyutlu indirgenmiş veri kümesinin bir örneğinin elde edilmesi amacıyla uygulanır. İndirgenmiş veri kümesine veri madenciliği teknikleri uygulanarak daha etkin sonuçlar elde edilebilir. Veri indirgeme yöntemleri;

- Veri Birleştirme (Data Aggregation)
- Boyut İndirgeme (Dimension Reduction)
- Veri Sıkıştırma (Data Compression)
- Kesikli hale getirme (Discretization)

Veri madenciliği çalışmasında bazı veriler kullanılan algoritma, model veya teknikler ile çalışamamaktadırlar. Böyle durumlarda verileri, uygulanacak tekniğe uygun hale getirmek için dönüşümler yapmak gerekebilir. Örneğin yılın çeyrek dönemi olarak alınan bir değişken, yıllık değişken olarak daha üst düzey değişkene genelleştirileceği gibi bir veri kaynağında tarihler gg/aa/yy biçiminde yer alırken başka bir veri kaynağında aa.gg.yy biçiminde yer alıyorsa ortak bir tarih biçimine dönüştürülebilir.

Veriler, veri madenciliği için veri dönüştürme ile uygun biçimlere dönüştürülürler. Veri dönüştürme; düzeltme, birleştirme, genelleştirme ve normalleştirme gibi işlemlerden biri veya daha fazlasını içerebilir. Veri normalleştirme en sık kullanılan veri dönüştürme işlemlerinden birisidir. Veri normalleştirme tekniklerinden bazıları aşağıdaki biçimde sıralanabilir.

- Min-Max

- Z Skor
- Ondalık Ölçekleme

Min-max normalleştirmesi ile orijinal veriler yeni veri aralığına doğrusal dönüşüm ile dönüştürülürler. Bu veri aralığı genellikle 0-1 aralığıdır. Ayrıca alan değerinin minimum değerden ne kadar büyük olduğuna bakar ve bu farkları sıralar (Larose, 2005:36)

$$X_* = \frac{X - X_{min}}{(X_{max} - X_{min})}$$

Z skor normalleştirmesiyle normal rassal değişkenleri standart normale dönüştürmek imkânı vardır: Eğer $\chi \sim N(\mu, \sigma^2)$ ise, bu halde

$$Z = \frac{X - \mu}{\sigma}$$

bir standart normal rassal değişken olur; yani $Z \sim N(0,1)$

Ondalık ölçekleme ile normalleştirmede ise, ele alınan değişkenin değerlerinin ondalık kısmı hareket ettirilerek normalleştirme gerçekleştirilir. Hareket edecek ondalık nokta sayısı, değişkenin maksimum mutlak değerine bağlıdır.

Araştırma probleminin tanımlanmasının ardından problem için en uygun modelin seçilebilmesi, çok fazla model kurularak kurulan bu modellerin denenmesi ile sağlanabilir. Bu nedenle veri hazırlama ve model kurma aşamaları en uygun modele ulaşıncaya kadar tekrar eden bir süreçtir.

Verileri analiz edecek kişinin hedeflerine uygun olarak değişik modelleme teknikleri kullanılması mümkündür. Model kuruluş süreci denetimli (supervised) ve denetimsiz (unsupervised) öğrenimin kullanıldığı modellere göre farklılık göstermektedir (Giudici, 2003:8).

Örnekten öğrenme olarak da isimlendirilen denetimli öğrenimde, bir denetçi tarafından ilgili sınıflar önceden belirlenen bir kritere göre ayrılarak, her sınıf için çeşitli örnekler verilmektedir. Sistemin amacı verilen örneklerden hareket ederek her bir sınıfa ilişkin özelliklerin bulunması ve bu özelliklerin kural cümleleri ile ifade edilmesidir. Öğrenme süreci tamamlandığında, tanımlanan kural cümleleri verilen yeni örneklere uygulanır ve yeni örneklerin hangi sınıfa ait olduğu kurulan model tarafından belirlenir (Akpınar, 2000:11).

Denetimsiz yöntemler veriyi ayırt etmeye, bilmeye, tanımaya, keşfetmeye yönelik olarak kullanılır. Denetimli öğrenmede olduğu gibi bilgi ve sonuç çıkarmaktan

ziyade daha sonraki yöntemler için fikir vermeyi amaçlamaktadır. Gözlenen örneklerin nitelikleri arasındaki benzerliklere göre sınıflanması amaçlanmaktadır. Denetimsiz yöntemler otomatik olarak haber sınıflandırma (örneğin news.google.com), benzer haberleri aynı kümelere koyup aynı başlangıçtan kullanıcılara sunmak, pazar analizi ile müşterilerin tercihleri açıklama gibi uygulamalarda kullanılabilir.

Bu nedenle denetimsiz bir yöntemle elde edilen bir bilgi veya sonucu, eğer mümkünse denetimli bir yöntemle teyit etmek, elde edilen bulguların doğruluğu ve geçerliliği açısından önem taşımaktadır (Koyuncugil, 2006:56).

Modeli belirlerken hangi tekniğin en uygun olduğuna karar verebilmek güçtür. Bu sebeple farklı modeller deneyerek, doğruluk derecelerine göre en uygun modeli bulmak üzere sayısız deneme yapılabilir.

Veri madenciliğinde çalışılan veri kümesinden çıkarılan bilgi bir doğruluk derecesine sahip olup deterministik bir bilgi değildir. Oluşturulan modellerin başarım derecelerini belirleyen doğruluk, kesinlik, duyarlılık ve F-ölçütü gibi kriterler kullanılarak kullanılan algoritmaların başarıları değerlendirilir. Model başarımını değerlendirme ölçütleri modelin ne kadar doğru sınıflandırma yaptığını ölçer, hız ve ölçeklenebilirlik gibi özellikleri değerlendirmez.

Veri önişleme, parametre seçimi ve test kümesi seçimi veri madenciliği uygulamasında ortaya çıkacak olan modelin başarımını etkiler. Dolayısı ile algoritmaların karşılaştırılarak hangi algoritmanın daha iyi olduğunu bulmaya yönelik yapılan karşılaştırma sonuçları büyük ölçüde uygulamacıya bağlıdır.

Model başarısını değerlendirirken kullanılan belli başlı kavramlar hata oranı, kesinlik, duyarlılık ve F-ölçütüdür. Modelin başarısı, gerçek sınıfa atanan örnek sayısı ve yanlış sınıfa atanan örnek sayısı ilişkilidir. Modelin kullanımı sonucunda elde edilen sonuçların başarı bilgileri karışıklık matrisi (confusion matrix) ile gösterilebilir. Karışıklık matrisinde satırlar deneme kümesindeki örneklere ait doğru sayıları, sütunlar ise modelin tahminlemesini gösterir.

Tablo 4: Karışıklık Matris Örneği

	Tahmin Edilen Sınıf		
Doğru (gerçek) Sınıf	TP (True Positive)	FN (False Negative)	P
	FP (False Positive)	TN (True Negative)	N
	P'	N'	Toplam

Kaynak: Coşkun ve Baykal, 2011:57

1.5.1 Doğruluk – Hata oranı

Bir ölçüm sisteminin doğruluğu, bir niceliğin ölçüm değerinin asıl (gerçek) değerine olan yakınlık derecesidir. Bir ölçüm sisteminin kesinliği, (tekrarlana bilirliği veya yinelene bilirliği de denir), aynı şartlardaki ölçümlerin aynı sonucu verme derecesidir.

Doğruluk, doğru olarak sınıflandırılmış veri sayısının (TP +TN), toplam veri sayısına (TP+TN+FP+FN) oranıdır. Hata oranı ise bu değerin 1'den farkını ifade eder. Başka bir deyişle hatalı olarak sınıflandırılmış veri sayısının (FP+FN), toplam veri sayısına (TP+TN+FP+FN) oranıdır.

$$Doğruluk = \frac{TP + TN}{TP + FP + FN + TN}$$

$$Hata\ Oranı = 1 - Doğruluk = \frac{FP + FN}{TP + FP + FN + TN}$$

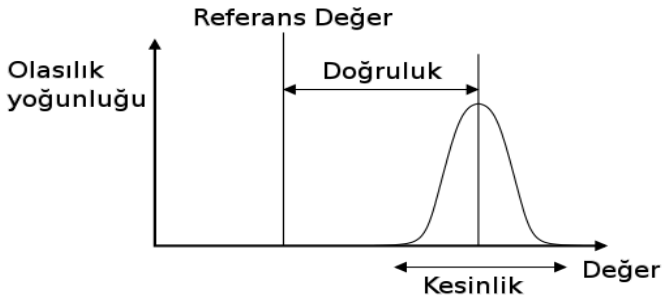
1.5.2 Kesinlik

Kesinlik, doğru pozitif örnek sayısının, toplam pozitif örnek sayısına oranıdır. Diğer bir ifadeyle tahminlerin içinden ne kadarını doğru tahmin ettiğimizin oranıdır.

$$Kesinlik = \frac{TP}{TP+FP}$$

Kesinlik dağılım, ortalama sapma, standart sapma, varyans, bağıl ortalama sapma terimleri ile ifade edilir.

Şekil 3: Doğruluk Ve Kesinlik



1.5.3 Duyarlılık (Sensitivity)

Doğru sınıflandırılmış pozitif örnek sayısının toplam pozitif örnek sayısına oranıdır.

$$Duyarlılık = \frac{TP}{TP + FN}$$

1.5.4 Özgüllük (Specifcity)

Özgüllük, testin gerçek negatif durumlar içinden negatif olan durumları ayırma yeteneğidir.

$$\text{Özgüllük} = \frac{TN}{TN + FP}$$

1.5.5 F-Ölçütü

Kesinlik ve duyarlılık ölçütleri tek başına anlamlı bir kıyaslama yapabilmemiz için yeterli değildir. Her iki ölçütü birlikte değerlendirmek daha güvenilir sonuçlara ulaşmamızı sağlar. Bu sonuçlara ulaşmak için f-ölçütü tanımlanmıştır. F-ölçütü, kesinlik ve duyarlılık ölçütlerinin harmonik ortalamasından elde edilir.

$$F - \text{Ölçütü} = \frac{2xDuyarlılıkxKesinlik}{Duyarlılık + Kesinlik}$$

Modelin doğruluğunun ölçülmesi için kullanılan en kolay yöntem, basit geçerlilik (Simple Validation) testidir. Bu yöntemde verilerin %33'lük (verilerin 1/3'ü) bir kısmı test verileri olarak ayrılır ve kalan %66 (verilerin 2/3'ü) kısım eğitim kümesi olarak seçilir. Fakat seçilen test kümeleri başarıyı çok fazla etkilediği için çok sağlıklı sonuçlar ortaya çıkmayabilir. Rastgele örnekleme (random sampling) yöntemi ile *k* kez farklı test kümeleri seçerek doğru tahmin arttırılabilir.

Kullanılabilecek diğer bir yöntem çapraz geçerlilik (Cross Validation) testidir. Bu yöntemde veri kümesi rastgele olarak *k=10* eşit parçaya ayrılır. İlk aşamada 1. parça test kümesi ve geri kalan 9 parça eğitim kümesi olarak alınır; ikinci aşamada ise 2. parça test kümesi ve geri kalan 9 parça eğitim kümesi olarak alınır. %10 test ve %90 eğitim kümesi olarak sistem çalıştırılır. Böylelikle 10 parçanın hepsi test kümesi ve hepsi eğitim kümesi olarak kullanılmış olur. Elde edilen hata oranlarının ortalaması kullanılır. Bu orana göre modelin başarı oranı belirlenebilir.

Bootstrapping, küçük veri kümeleri için modelin hata düzeyinin tahmininde kullanılan bir başka tekniktir. Çapraz geçerlilikte olduğu gibi model bütün veri kümesi üzerine kurulmaktadır. Daha sonra en az 200, bazen 1000'in üzerinde olmak üzere çok fazla sayıda öğrenim kümesi tekrarlı örneklemelerle veri kümesinden oluşturularak hata oranı hesaplanmaktadır (Akpınar, 2000:12).

Özellikle sınıflama problemleri için kurulan modellerin doğruluk derecelerinin değerlendirilmesinde basit ancak faydalı bir araç olan karışıklık matrisi kullanılmaktadır. Karışıklık matrisi daha ayrıntılı incelediğimizde algoritma test sonuçlarını özetlediğini görebiliriz. Bu matriste satırlarda gerçek, sütunlarda ise tahmini sınıflama değerleri yer almaktadır. Örneğin 8 kedi, 6 köpek ve 13 tavşandan oluşan 27 hayvanın olduğu bir örnek varsayarsak, ortaya çıkan karışıklık matrisinin aşağıda Tablo 5 de gibi göründüğünü varsayalım.

Tablo 5: Örnek Karışıklık Matrisi

		Tahmini sınıf		
		kedi	köpek	tavşan
Gerçek sınıf	kedi	5	3	0
	köpek	2	3	1
	tavşan	0	2	11

5 doğru pozitif gerçekte kedi olup doğru bir şekilde kedi olarak sınıflandırılanlar	2yanlış pozitif gerçekte köpek olup yanlış bir şekilde kedi olarak sınıflandırılanlar
3 yanlış negatif gerçekte kedi olup yanlış bir şekilde köpek olarak sınıflandırılanlar	17 doğru negatif kedi olmayan olarak sınıflandırılan geri kalan bütün hayvanlar

Modelin başarı oranını kedi, köpek ve tavşanı doğru sınıflandırıp sınıflandırmadığına bakarak bulmaya çalışalım. Örneğin gerçekte 8 kedi varken, kurulan modelin bu kedilerin 3'ünü köpek olarak sınıflandırıldığı ve 6 köpeğin, 1'ini tavşan ve 2'sini kedi olarak sınıflandırıldığı görülmektedir. Söz konusu modelde kedi ve köpekler arasında ayrım sorunu vardır, ama tavşan ve diğer hayvanların türleri arasındaki oldukça iyi ayrım yapabilmiştir. Tüm doğru tahminler tablonun köşegenindeki değerlerdir, tablo incelendiğinde köşegen dışında kalan değerlerin hataları temsil ettiği kolayca görülmektedir. Modelin karışıklık matrisinden alınan bilgilere göre %81 doğru olduğu ve %67 başarı oranı olduğu görülebilir.

$$Doğruluk = \frac{TP+TN}{TP+FP+FN+TN} = \frac{22}{27} = 0{,}81$$

$$Duyarlılık = \frac{TP}{TP+FN} = \frac{5}{8} = 0{,}625$$

$$Hata\ Oranı = \frac{FP+FN}{TP+FP+FN+TN} = \frac{5}{27} = 0{,}185$$

$$Kesinlik = \frac{TP}{TP+FP} = \frac{5}{7} = 0{,}714$$

$$F - \text{Ölçütü} = \frac{2xDuyarlılıkxKesinlik}{Duyarlılık+Kesinlik} = \frac{2x0{,}625x0{,}714}{0{,}625+0{,}714} = 0{,}667$$

Kurulan ve geçerliliği kabul edilen model doğrudan bir uygulama olabileceği gibi, bir başka uygulamanın alt parçası olarak kullanılabilir. Kurulan modeller risk analizi, kredi değerlendirme, dolandırıcılık tespiti gibi işletme uygulamalarında doğrudan kullanılabileceği gibi, promosyon planlaması simülasyonuna entegre edilebilir (Akpınar, 2000:14).

Modeli belirlerken hangi algoritmanın en uygun olduğuna;

- Modelin kesinliğine yani kaç örneğin kaçının doğru sınıflandırıldığına
- Modelin sonuçlanma hızına,
- Eksik ve gürültülü verilerden etkilenme durumuna,
- Ölçeklendirilebilirliğine
- Yorumlanabilirliğine

göre karar verilebilir.

Zamanın ilerlemesiyle birlikte sistemlerin özelliklerinde ve ürettiği verilerde değişiklikler meydana gelmektedir. Kurulan modeller sürekli olarak takip edilmeli ve herhangi bir değişim söz konusu ise güncellenmelidir. Modelin güncelliğini kaybedip kaybetmediğini anlamak için kullanılan, tahmin edilen ve gözlenen değişkenler arasındaki farklılığı gösteren grafikler en faydalı yöntemlerden biridir.

Bilimin ve teknolojinin gelişmesi ile birlikte kullanılan bilgisayar programları ve modeller güncelliğini kaybedebilir. Bu sebeple kullanılan programlar ve yeni geliştirilen modeller sürekli takip edilmeli sistemler sürekli güncellenmelidir.

İKİNCİ BÖLÜM
VERİ MADENCİLİĞİ ALGORİTMALARI VE MODELLERİ

2.1 VERİ MADENCİLİĞİ MODELLERİ

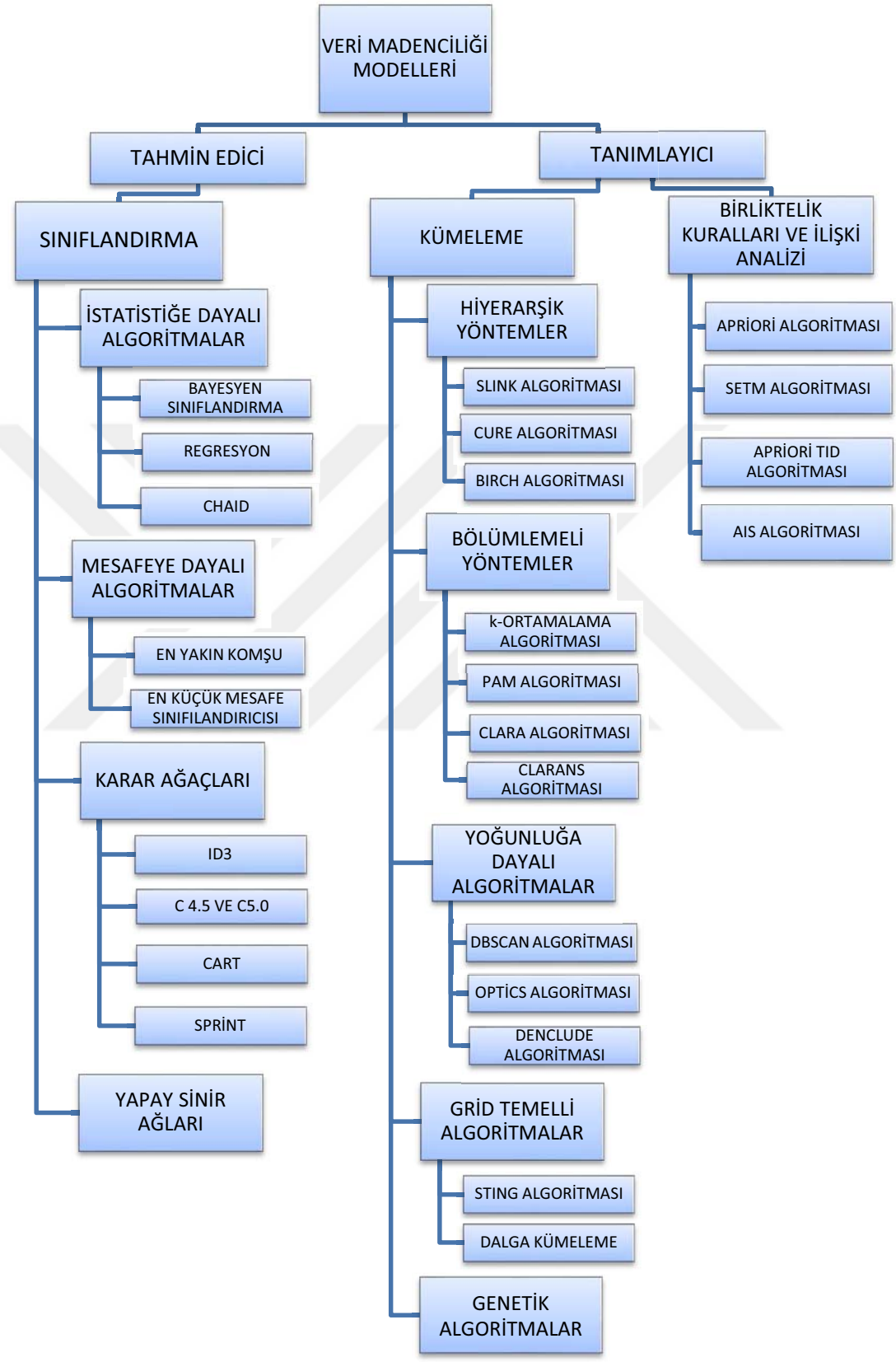
Veri madenciliğinin uygulama aşaması için türetilmiş çok sayıda algoritma mevcuttur. Uygulanacak algoritmanın seçilmesi, verilerin niteliğine ve kullanıcının seçimine göre belirlenmektedir. Veri madenciliği modelleri, kullanıcıların farklı model arayışına öncülük etmek ve hedefe ulaşmak için ipuçları belirlemelerine de olanak sağlamalıdır.

Veri madenciliği aynı zamanda deneysel bir çalışma olduğu için birden fazla algoritmanın ayrı ayrı sınanmasında fayda vardır. Birçok algoritmanın denenmesi sonucunda çalışılan verilerde en başarılı sonuçları veren algoritma ya da algoritmalar seçilebilir. Bu algoritmalar kullanılarak kullanılacak model belirlenebilir. Ayrıca veriler zaman içinde farklılaşma göstereceğinden modellerinde zaman içinde yenilenmesi ve geliştirilmesi gerekebilir.

Veri madenciliği modelleri tanımlayıcı ve tahmin edici olarak ikiye ayrılır. Tanımlayıcı modeller, veri tabanındaki verilerin genel özelliklerini karakterize eder. Tahmin edici modeller, mevcut veriler üzerinden öngörülerde bulunurlar. Tahmin edici modeller sınıflandırma algoritmaları olarak istatistiğe dayalı ve mesafeye dayalı algoritmalar, karar ağaçları, yapay sinir ağları modellerinden oluşur. Tanımlayıcı modeller kümeleme ve birliktelik kuralları olarak ikiye ayrılır. Kümeleme algoritmaları hiyerarşik yöntemler, bölünmeli yöntemler, yoğunluğa dayalı, grid temelli, genetik algoritmalardan oluşur.

Algoritmalar ve uygulama alanlarına göre dağılımları Şekil 4 de gösterilmiştir.

Şekil 4: Veri Madenciliği Modelleri



2.2 SINIFLANDIRMA TEKNİK VE ALGORİTMALARI

Verinin içerdiği ortak özelliklere göre ayrıştırılması işlemine sınıflandırma denir. Başka bir ifadeyle hangi sınıfa ait olduğu bilinmeyen bir kayıt için bir sınıf belirleme sürecidir.

Sınıflandırma en çok bilinen veri madenciliği tekniklerinden birisidir; resim, örüntü tanıma, hastalık tanıları, dolandırıcılık tespiti, kalite kontrol çalışmaları ve pazarlama konuları sınıflandırma tekniklerinin en çok kullanıldığı alanlardır. Sınıflandırma algoritmaları tahmin edici bir algoritmadır; birisi yaşınızı tahmin ediyor ya da bir kavanozdaki misket sayısını tahmin ediyorsa, bunlar aslında sınıflandırma problemleridir(Dunham, 2003:75).

Sınıflandırma kavramı, basitçe bir veri kümesi (data set) üzerinde tanımlı olan çeşitli sınıflar arasında veriyi dağıtmaktır. Sınıflandırma algoritmaları, verilen eğitim kümesinden bu dağılım şeklini öğrenirler ve daha sonra sınıfının belirli olmadığı test verileri geldiğinde doğru şekilde sınıflandırmaya çalışırlar. Veri kümesi üzerinde verilen bu sınıfları belirten değerlere etiket (label) ismi verilir ve gerek eğitim gerekse test sırasında verinin sınıfının belirlenmesi için kullanılırlar.

Veri madenciliğinde modeller, tanımlayıcı ve tahmine dayalı modeller olmak üzere temel olarak ikiye ayrılır. Tanımlayıcı modellerde amaç, veri kümesindeki değişkenler arasındaki ilişkileri özetleyen örüntüleri ortaya çıkarmakken, tahmine dayalı modellerde belirli bir değişkenin değerini diğer değişkenlerden yararlanarak tahmin etmek amaçlanır (Tan ve diğerleri 2006:147). Sınıflama modelleri, bağımlı değişkenin kategorik olduğu tahmine dayalı modellerdir. Bu modellerde bağımlı değişken "sınıf değişkeni" olarak da adlandırılabilir. Sınıflama modellerinde amaç, sınıflar arasındaki sınırların belirlenerek bağımsız değişkenin verilen değerlerine göre bağımlı değişken değerini doğru bir şekilde tahmin etmektir.

Verilerin sınıflandırma süreci iki adımdan oluşur (Han ve Kamber, 2006:30).

İlk başta, veri tabanına uygun model seçilmelidir. Bu model, veri tabanındaki kayıtların özellikleri veya alan isimleri kullanılarak gerçekleştirilir. Sınıflandırma modelinin kurulması aşamasında veri tabanından bir kısmı rasgele olarak seçilen veriler eğitim verileri olarak ele alınır. Kalan veriler ise test verileri olur. Daha sonra eğitim verileri üzerinde bir algoritma deneyerek bir sınıflama modeli oluşturulur

İkinci adımda, test verileri olarak ayrılan veriler ile sınıflandırma kuralları belirlenir. Aynı kurallar bu sefer test verilerine uygulanarak denenir. Test verilerine uygulanan sınıflandırma kuralları sonucu elde edilen modelin doğruluğu geçerli görülürse, bu model sınıflama modeli olarak belirlenir.

Sınıflandırma algoritmaları istatistiğe dayalı ve mesafeye dayalı algoritmalar, karar ağaçları, yapay sinir ağları gibi yöntemlerle uygulanabilmektedir.

2.2.1 İstatistiğe Dayalı Algoritmalar

Sınıflandırma tekniklerinden biri olan istatistiğe dayalı algoritmalar kullanılırken istatistiksel yöntemlerden yararlanılır. Bağımlı ve bağımsız değişkenler belirlenerek kullanılacak model buna göre oluşturulur. İstatistiğe dayalı algoritmalardan en bilineni "*CHAID Algoritması"*dır.

2.2.1.1 Bayesyen Sınıflandırma

Olasılık teorisi içinde incelenen bir olay olarak B olayına koşullu bir A olayı (yani B olayının bilindiği halde A olayı) için olasılık değeri, A olayına koşullu olarak B olayı (yani A olayı bilindiği haldeki B olayı) için olasılık değerinden farklıdır. Ancak bu iki birbirine ters koşulluluk arasında çok belirli bir ilişki vardır ve bu ilişkiye (ilk açıklayan istatistikçi İngiliz Thomas Bayes (1702–1761) adına atfen Bayes Teoremi denilmektedir.

Bu teorem bir rassal değişken için olasılık dağılımı içinde koşullu olasılıklar ile marjinal olasılıklar arasındaki ilişkiyi gösterir.

Naive Bayes Algoritması her kriterin sonuca olan etkilerinin olasılık olarak hesaplanması temeline dayanmaktadır. Yani elimizde olan sınıflandırılmış verileri kullanarak yeni bir verinin var olan sınıflardan herhangi birine girme olasılığını hesaplayan yöntemdir. Bir örneğin sınıf üyelik olasılığını kestirir.

Naive Bayes Algoritmasının Avantajları

- Algoritmayı kodlamasının ve sonucunda ne çıkacağını tahmin etmesinin kolay olması

- Elde edilen sonuçlar birbirinden bağımsız değişkenler kullanıldığında karmaşık algoritmalar ile eşdeğer nitelikte başarı sağlaması

  Naive Bayes Algoritmasının Dezavantajları

- Gerçek verilerdeki değişkenlerin birbirinden bağımsız verilerini bulmanın çok zor olması
- Değişkenler arasında ilişkiyi bu algoritma ile gösterilemiyor olması yatmaktadır.

x_i sınıf üyeliği bilinmeyen veri örneği ve örnek $x_i = \{ x_1, x_2, \ldots, x_m \}$ nitelik değerlerinden oluşsun. Bu örnek sınıfta n adet sınıf olduğunu varsayılırsa ve C_1 , C_2 ,...,C_n sınıf değerleri olduğunu düşünülerse x_i nin C_i sınıfında olma olasılığını aşağıdaki gibi formüle edilir ve

$$P(C_i/x_i) = \frac{P(x_i/C_i)P(C_i)}{P(x_i)}$$

olarak hesaplanabilir. $P(x_i)$, x_i değerinin, $P(C_i)$ ise C_i sınıfının veri tabanında bulunma olasılığıdır. Hesaplardaki işlem yükünü azaltmak ve yeni veriyi sınıflandırmak için payda da yer alan $P(x_i)$ değerleri birbirine eşit olduğundan sadece pay değerlerinin karşılaştırmak yeterlidir. Bu değerler içinden en büyük olanı seçilerek bilinmeyen örneğin bu sınıfa ait olduğu belirlenmiş olur.

$$\underset{C_i}{\operatorname{argmax}} \{P(x_i/C_i)\ P(C_i)\ \}$$

Bayes algoritmasını bir örnek ile açıklanabilir. Aşağıdaki tablo 6 bulunan 15 adet örnek veriyi kullanarak x= { Erkek, 172cm, 65kg} olarak verilen yeni verinin küçük, orta ve büyük beden sınıflarından hangisine dâhil olduğunu Bayes kuralı ve algoritmasına ile gösterilmek istenirse;

Tablo 6: Cinsiyet, Kilo, Boy ve Beden Verileri

CİNSİYET	KİLO	BOY	BEDEN
K	48	170	ORTA
K	49	151	KÜÇÜK
K	52	158	ORTA
K	56	165	ORTA
E	59	160	KÜÇÜK
K	61	159	ORTA
E	62	162	KÜÇÜK
E	63	174	ORTA
K	68	168	ORTA
K	69	177	BÜYÜK
E	72	170	ORTA
E	74	165	KÜÇÜK
E	85	175	ORTA
E	85	190	BÜYÜK
E	95	190	BÜYÜK

CİNSİYET	KİLO	BOY	BEDEN
K	1	3	ORTA
K	1	1	KÜÇÜK
K	1	1	ORTA
K	2	3	ORTA
E	2	2	KÜÇÜK
K	2	1	ORTA
E	2	2	KÜÇÜK
E	2	3	ORTA
K	3	3	ORTA
K	3	4	BÜYÜK
E	3	3	ORTA
E	3	3	KÜÇÜK
E	4	4	ORTA
E	4	5	BÜYÜK
E	5	5	BÜYÜK

Kaynak**:** Silahtaroğlu, 2013:71-99

Öncelikle boy ve kilo sürekli değişkenleri aralık değerleri olarak ifade edilebilir ve boy ve kilo değerleri aşağıdaki gibi etiketlenebilir.

BOY	KİLO
151-159.99 arası 1. Grup	[45-55,99] arası 1. Grup
160-164.99 arası 2. Grup	[56-64,99] arası 2. Grup
165-174.99 arası 3. Grup	[65-75,99] arası 3. Grup
175-184.99 arası 4. Grup	[76-85,99] arası 4. Grup
185- üzeri de 5. Grup	[86-100] arası 5. Grup

Dolayısıyla x= {E, 172, 65} verilerin yeni haline uygun olarak x={E,3,3} şeklinde ifade edilebilir. Yani

x_1 : cinsiyet= erkek

x_2 : kilo = 3. Grup

x_3 : boy = 3. Grup

BEDEN =?

Bayes olasılık değerlerinde yardımcı olması amacı ile verilerin olasılık değerlerini tabloda düzenlenebilir.

Tablo 7 : Bayes Olasılık Değerleri

		BEDEN					
Nitelik	Değeri	KÜÇÜK		ORTA		BÜYÜK	
		Sayı	Olasılık	Sayı	Olasılık	Sayı	Olasılık
Cinsiyet	KADIN	1	1/4	5	5/8	1	1/3
	ERKEK	3	3/4	3	3/8	2	2/3
KİLO	[45-55,99] - 1. Grup	1	1/4	2	2/8	0	0
	[56-64,99] - 2. Grup	2	2/4	3	3/8	0	0
	[65-75,99] - 3. Grup	1	1/4	2	2/8	1	1/3
	[76-85,99] - 4. Grup	0	0	1	1/8	1	1/3
	[86-100] - 5. Grup	0	0	0	0	1	1/3
BOY	151-159.99 - 1. Grup	1	1/4	2	2/8	0	0
	160-164.99 - 2. Grup	2	2/4	0	0	0	0
	165-174.99 - 3. Grup	1	1/4	5	5/8	0	0
	175-184.99 - 4. Grup	0	0	1	1/8	1	1/3
	185- üzeri - 5. Grup	0	0	0	0	2	2/3

P(C_j) değerleri yani küçük, orta ve büyük beden değerlerinin örnek veri tabanında bulunma olasılığı hesaplandığında;

P(küçük) = P($C_1 = küçük\ beden$) = 4/15 = 0,267

P(orta) = P($C_2 = orta\ beden$) = 8/15 = 0,534

P(büyük) = P($C_3 = büyük\ beden$) = 3/15 = 0,2

P (x/C_j) değeri yani P(x/beden= küçük) koşullu olasılığını hesaplana bilmesi için yeni verimizin cinsiyet, kilo, boy değerlerinin ayrı ayrı koşullu olasılık değerleri bulunabilir;

$P(x_1/C_1)$ = P(cinsiyet= erkek / beden=küçük) = 3/4

$P(x_2/C_1)$= P(kilo=3. Grup / beden=küçük) = 1/4

$P(x_3/C_1)$= P(boy=3. Grup / beden=küçük) = 1/4

$P(X/C_1) = P(X/beden = küçük)$ = (3/4)(1/4)(1/4) = 3/64

P(küçük) = P($C_1 = küçük\ beden$) = 4/15

$P(X/C_1)\ P(C_1)$ = 3/64 x 4/15 =0,0125

Aynı şekilde P(x/beden= orta) koşullu olasılığını hesaplanabilir;

$P(x_1/C_2)$ = P(cinsiyet= erkek / beden=orta) = 3/8

$P(x_2/C_2)$= P(kilo=3. Grup / beden=orta) = 2/8

$P(x_3/C_2)$= P(boy=3. Grup / beden=orta) = 5/8

$P(X/C_2) = P(X/beden = orta)$ = (3/8)(2/8)(5/8) = 30/512

P(orta) = P($C_2 = orta\ beden$) = 8/15

$P(X/C_2)\ P(C_2)$ = 30/512 x 8/15 = 0,03125

P(x/beden= büyük) koşullu olasılığı ise;

$P(x_1/C_3)$ = P(cinsiyet= erkek / beden=büyük) = 2/3

$P(x_2/C_3)$= P(kilo=3. Grup / beden=büyük) = 1/3

$P(x_3/C_3)$= P(boy=3. Grup / beden=büyük) = 0

$P(X/C_3) = P(X/beden = büyük)$ = (2/3)(1/3)(0) = 0

P(büyük) = P($C_3 = büyük\ beden$) = 3/15

$P(X/C_3)\ P(C_3)$ = 0 x 3/15 = 0

Bilinmeyen örnek X'i sınıflandırmak için $\underset{C_i}{\text{argmax}}\{P(X/C_i)\ P(C_i)\ \}$ seçimi yapılır.

$\underset{C_i}{\text{argmax}}\{P(X/C_i)\ P(C_i)\ \}$ = {0.0125, 0.03125, 0} = 0.03125

Bu koşullarda yeni veri olan {Erkek, 172cm, 65 kg} kişinin, örneğin 0.03 olasılığı ile ilgili olan sınıfa yani Orta beden sınıfına dâhil olduğu görülmüştür.

2.2.1.2 Regresyon

Regresyon analizi herhangi bir değişkenin bir veya daha fazla başka değişkenler arasındaki ilişkinin matematiksel bir denklem şeklinde yazılmasıdır, yazılan bu denkleme regresyon denklemi adı verilir.

Veri madenciliği için düşünmek gerekirse y sınıfları göstermektedir. x ise bağımsız değişkeni yani özellikleri temsil etmektedir. Verilen x değişken değerlerine göre y nin hangi sınıfa ait olacağı tahmin edilmeye çalışılır. Birden fazla değişkenden yani birden fazla x değerinden oluşuyorsa; çoklu regresyon olarak adlandırılır.

Regresyon analizi bir örnek ile açıklanabilir. Değişkenler cinsiyet, kilo, boy ve beden olsun ve cinsiyet, boy ve kilo bağımsız değişkenler yani x leri ifade ettiği düşünülsün. Beden ise x e bağlı değişken yani y olarak ele alınabilir. Regresyon analizi diğer sınıflandırma modellerindeki gibi öğrenme aşaması ile başlar. Bu adımda regresyon analiz denklemi elde edilir. Daha sonraki süreçte ise hangi bedene ait olduğu bilinmeyen bir örneğin; boy, kilo ve cinsiyeti verilmiş olsun hangi sınıfa yani hangi bedene sahip olduğu bulunabilir.

Regresyon analizinin avantajları;

- Regresyon analizi, bir bağımlı değişkenin başka açıklayıcı değişken değişkenlere olan bağımlılığını tahmin etmeyi amaçlar.
- Regresyon analizi elimizde bulunan geçmişe ait veriler yardımıyla geleceğe dair tahminler yürütmemize olanak sağlar.
- Regresyon analizi, aralarında sebep-sonuç ilişkisi bulunan iki veya daha fazla değişken arasındaki ilişkiyi belirlemek için kullanılır.

Regresyon analizini de bir örnek ile açıklanabilir. Bayes algoritmasını açıklarken kullanılan 15 adet örnek veri kullanılmak istenirse x= { Erkek, 172cm, 65 kg} olarak verilen yeni verinin küçük, orta ve büyük beden sınıflarından hangisine dâhil olduğunu regresyon analizi yardımıyla gösterilebilir;

Tablo 8: Regresyon Analizi İçin Örnek Veri Değerleri

CİNSİYET	KİLO	BOY	BEDEN
1	1	3	ORTA=2
1	1	1	KÜÇÜK=1
1	1	1	ORTA=2
1	2	3	ORTA=2
2	2	2	KÜÇÜK=1
1	2	1	ORTA=2
2	2	2	KÜÇÜK=1
2	2	3	ORTA=2
1	3	3	ORTA=2
1	3	4	BÜYÜK=3
2	3	3	ORTA=2
2	3	3	KÜÇÜK=1
2	4	4	ORTA=2
2	4	5	BÜYÜK=3
2	5	5	BÜYÜK=3

$$x_i = \begin{vmatrix} 1 & 1 & 1 & 3 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 2 & 3 \\ 1 & 2 & 2 & 2 \\ 1 & 1 & 2 & 1 \\ 1 & 2 & 2 & 2 \\ 1 & 2 & 2 & 3 \\ 1 & 1 & 3 & 3 \\ 1 & 1 & 3 & 4 \\ 1 & 2 & 3 & 3 \\ 1 & 2 & 3 & 3 \\ 1 & 2 & 4 & 4 \\ 1 & 2 & 4 & 5 \\ 1 & 2 & 5 & 5 \end{vmatrix} \quad Y = \begin{vmatrix} 2 \\ 1 \\ 2 \\ 2 \\ 1 \\ 2 \\ 1 \\ 2 \\ 2 \\ 3 \\ 2 \\ 1 \\ 2 \\ 3 \\ 3 \end{vmatrix}$$

Cinsiyet değişkenini olan kadın=1, erkek=2 aynı şekilde beden değerleri de küçük=1, orta=2 büyük=3 olarak kodlanabilir modelin bağımsız değişkenleri cinsiyet, kilo, boy ve bağımlı değişkeni beden olan çoklu regresyon analizi ile oluşturulabilir.

x_i'ler bağımsız değişkenleri ve Y'de bağımlı değişkeni göstermek üzere en genel çoklu regresyon denklemi;

$$Y = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \cdots + b_nx_n + e_i$$

şeklinde tanımlanabilir. Tahmin denklemi oluşturmak için matris yöntemiyle hesaplamalar yapılabilir. Bağımsız x_i değişkenlerine ve *Y'*de bağımlı değişkenine ait matrisler;

$$X^TX = \begin{vmatrix} n & \Sigma x_1 & \Sigma x_2 & \Sigma x_3 \\ \Sigma x_1 & \Sigma x_1^2 & \Sigma x_1x_2 & \Sigma x_1x_3 \\ \Sigma x_2 & \Sigma x_2x_1 & \Sigma x_2^2 & \Sigma x_2x_3 \\ \Sigma x_3 & \Sigma x_3x_1 & \Sigma x_3x_2 & \Sigma x_3^2 \end{vmatrix} = \begin{vmatrix} 15 & 23 & 38 & 43 \\ 23 & 39 & 63 & 70 \\ 38 & 63 & 116 & 127 \\ 43 & 70 & 127 & 147 \end{vmatrix}$$

Denklemdeki *b* değerleri matris yardımı ile aşağıdaki şekilde hesaplanır;

$$b = (X^TX)^{-1}X^TY$$

$$(X^TX)^{-1} = \begin{vmatrix} 0{,}779 & -0{,}366 & 0{,}041 & -0{,}090 \\ -0{,}366 & 0{,}386 & -0{,}106 & 0{,}014 \\ 0{,}041 & -0{,}106 & 0{,}196 & -0{,}131 \\ -0{,}090 & 0{,}014 & -0{,}131 & 0{,}140 \end{vmatrix}$$

$$X^TY = \begin{vmatrix} 29 \\ 44 \\ 80 \\ 92 \end{vmatrix} \qquad b = (X^TX)^{-1}X^TY = \begin{vmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{vmatrix} \Longrightarrow \begin{vmatrix} 1{,}487 \\ -0{,}822 \\ 0{,}153 \\ 0{,}406 \end{vmatrix}$$

Olarak bulunur. Modelde;

$Y = b_0 + b_1x_1 + b_2x_2 + b_3x_3 = 1{,}487 - 0{,}822x_1 + 0{,}153x_2 + 0{,}406x_3$ olur.

Yeni veri olarak hangi beden kategorisinde olduğu bulunmak istenen x={E,172, 65} değerini x = {E=2, 65=3, 172=3} olarak grupların değeri ile yazılırsa x={2,3,3} olur. Model de değerler yerine konulduğunda;

$Y = 1{,}487 - 0{,}822 * 2 + 0{,}153 * 3 + 0{,}406 * 3 = 1{,}52 \approx 2$ bulunur.

Orta beden değerini 2 olarak sınıflandırıldığından bu yeni değerlere sahip kişi orta beden kategorisinde olacaktır.

2.2.1.3 CHAID algoritması

CHAID algoritması, 1980'de Kaas tarafından en iyi bölmeyi hesaplamak için istatistik olarak anlamlı bir farklılığın olmadığı, hedef değişkene uyan çiftlerde tahmin değişkeninin olası kategori çiftini birleştirmesiyle oluşturulmuştur.

Chaid analizi, kategorik değişkenlere ilişkin veri kümesini ve bağımlı değişkeni en iyi açıklayabilecek şekilde ayrıntılı homojen alt gruplara böler. Bu alt kümeler, küçük tahmin edici gruplardan oluşur. En iyi tahmin sonucunu elde etmek için başlangıç değişkenleri bağımsız olarak yeniden kategorileştirilir. Bunun için Ki kare analizi kullanılır. Adımsal olarak uygulanan benzer kategorileri birleştirme işlemi, değişkenler arasında daha fazla birleştirme sağlanamayacağına istatistiksel olarak karar verilinceye kadar devam eder. Değişkenlerin bölünmeye uygun olup olmadığına Benferroni düzeltilmiş "p" değeri kullanılarak karar verilir.

En uygun bölümleri seçmek için kullanılan *entropi* veya *gini metrikleri* yerine *ki-kare (chisquare)* testi kullanılmaktadır (Thomas and Galambos, 2004:257). Hedef değişkeni, tahmin edici değişkenlerle ilişki düzeyine göre sınıflandırma amacıyla; diğer karar ağacı algoritmalarından farklı olarak ikiden fazla gruba ayırarak dallanan algoritma, tüm olası alt grupları ağaç biçiminde kolay anlaşılır biçimde göstermektedir. Bu analiz sonucunda bağımlı değişken üzerine etkisi istatistiksel olarak önemli bulunan bağımsız değişkenler içinde en yüksek *F* değerine sahip olan değişken, CHAID diyagramında ilk sırayı almaktadır (Koyuncugil,2006:75).

CHAID, *ki-kare* metriği vasıtasıyla ilişki düzeyine göre farklılık rastlanan grupları ayrı ayrı sınıflamakta ve ağacın yaprakları, ikili değil, verideki farklı yapı sayısı kadar dallanmaktadır.

Algoritmanın adımları gereğince her bir alt grup için elde edilmesi gerekli olan ki-kare (χ^2)değerleri ise;

$$\chi^2 = \sum_{j=1}^{c} \sum_{i=1}^{r} \frac{\left(G_{ij} - B_{ij}\right)^2}{B_{ij}}$$

formülü ile hesaplanmaktadır.

G_{ij} = j. sütun ve i. satırdaki gözlenen değer

B_{ij} = j. sütun ve i. satırdaki beklenen değer

c = sütun değişkeninin düzey sayısı

r = satır değişkeninin düzey sayısı

$$B_{ij} = \frac{(n_i)(n_j)}{n} \qquad s.d. = (r\text{-}1)\,(c\text{-}1)$$

CHAID algoritması aşağıda verilmektedir:

1. Her bir tahmin edici değişken X için, X'in, Y hedef değişkenini dikkate alan en az öneme sahip kategori çiftini bulunarak (en büyük p değerine sahip olan değerdir). Y'nin ölçüm düzeyine bağlı olarak p değerleri farklı hesaplanır.

a. Y sürekli ise F testi kullanılır.

b. Y isimsel ise X'in kategorileri satırlarda ve Y'nin kategorileri sütunlarda olacak biçimde iki yönlü çapraz tablo düzenlenir. Pearson ki-kare testi veya olabilirlik oranı testi kullanılır.

c. Y sıralı ise bir Y birliktelik modeli uydurulur ve olabilirlik oranı testi kullanılır.

2. En büyük p değerine sahip X'in kategori çifti için, p değerini önceden belirlenmiş alfa düzeyi $\alpha_{birleş}$ ile kıyaslanır.

a. Eğer p değeri $\alpha_{birleş}$'den büyük ise bu çifti bir tek kategori altında birleştirilir. X'in yeni kategori kümesi için süreci Adım 1'den başlatılır.

b. Eğer p değeri $\alpha_{birleş}$'den küçük ise Adım 3'e gidilir.

3. X'in ve Y'nin kategori kümesi için uygun Bonferroni düzeltmesi kullanılarak, düzeltilmiş p değerini hesaplanır.

Bonferroni çarpanı;

- Eğer açıklayıcı değişken kategorileri nominal ise;

$$B_{serbest} = \sum_{i=0}^{r-1}(-1)^i \frac{(r-i)^c}{r!(r-i)!}$$

- Eğer açıklayıcı değişkenin kategorileri sıra belirten bir ölçek düzeyinde ise;

$$B_{monoton} = \frac{(c-1)}{(r-1)}$$

formülleri ile hesaplanmaktadır. (c kategori sayısını, r ise $1 \leq r \leq c$ olmak üzere açıklayıcı değişkenin grup sayısını ifade etmektedir.)

4. En küçük düzeltilmiş p değerine sahip X tahmin edici değişkenini seçilir. Bunun p değeri önceden tanımlanılmış alfa düzeyi $\alpha_{bölünmüş}$ ile kıyaslanır.

a. Eğer p değeri, $\alpha_{bölünmüş}$ değerinden küçük veya eşit ise düğüm X'in kategori kümesini temel alarak bölünür.

b. Eğer p değeri, $\alpha_{bölünmüş}$ değerinden büyük ise düğüm bölünmez. Bu düğüm uç düğüm olarak alınır.

5. Ağaç büyütme sürecini durma kuralları görülene kadar sürdürülür(SPSS 2002).

Bir örnekle açıklanmak gerekirse; bir yarış atının pistin durumu, havanın yağışlı olup olmaması ve koşunun sabah, öğle, akşamüzeri gibi zamanlarda yapılması durumunda atın koşu derecesine etkisinin olup olmadığı ve bu etmenlerin nasıl bir düzen içinde etki ettiği araştırılmak istenilmiş olursa(Silahtaroğlu, 2013:106);

Tablo 9: CHAID Algoritması İçin Koşu Verileri

Yarış	Pist	Yağı	Zaman	Sır	Yarış	Pist	Yağış	Zaman	Sır
1	Cim	Var	Sabah	1	47	Tarta	Yok	Aksamü	1
2	Kum	Yok	Öğle	1	48	Cim	Var	Sabah	3
3	Tarta	Yok	Aksamüze	2	49	Tarta	Var	Aksamü	2
4	Cim	Var	Sabah	2	50	Cim	Var	Sabah	2
5	Kum	Yok	Öğle	2	51	Tarta	Var	Aksamü	1
6	Tarta	Yok	Aksamüze	2	52	Cim	Var	Sabah	2
7	Cim	Var	Sabah	3	53	Cim	Var	Sabah	1
8	Kum	Yok	Öğle	1	54	Tarta	Yok	Aksamü	3
9	Tarta	Var	Aksamüze	1	55	Cim	Var	Sabah	2
10	Cim	Var	Sabah	3	56	Tarta	Yok	Aksamü	2
11	Kum	Yok	Öğle	2	57	Cim	Var	Sabah	1
12	Tarta	Var	Aksamüze	3	58	Tarta	Var	Aksamü	1
13	Cim	Var	Sabah	3	59	Cim	Var	Sabah	1
14	Kum	Yok	Öğle	3	60	Tarta	Var	Aksamü	3
15	Tarta	Yok	Aksamüze	1	61	Cim	Var	Sabah	3
16	Cim	Var	Sabah	3	62	Tarta	Yok	Aksamü	1
17	Kum	Yok	Öğle	3	63	Cim	Var	Sabah	3

18	Tarta	Var	Aksamüze	2	64	Tarta	Var	Aksamü	2
19	Cim	Var	Sabah	2	65	Cim	Var	Sabah	2
20	Kum	Yok	Öğle	3	66	Tarta	Var	Aksamü	1
21	Tarta	Var	Aksamüze	1	67	Cim	Var	Sabah	2
22	Cim	Var	Sabah	2	68	Cim	Var	Sabah	1
23	Cim	Var	Sabah	1	69	Tarta	Yok	Aksamü	2
24	Tarta	Var	Aksamüze	2	70	Cim	Var	Sabah	2
25	Tarta	Var	Sabah	2	71	Tarta	Yok	Aksamü	2
26	Tarta	Yok	Aksamüze	2	72	Cim	Var	Sabah	1
27	Tarta	Var	Sabah	3	73	Tarta	Var	Aksamü	1
28	Tarta	Var	Aksamüze	1	74	Cim	Var	Sabah	1
29	Tarta	Var	Sabah	3	75	Tarta	Var	Aksamü	3
30	Tarta	Var	Aksamüze	3	76	Cim	Var	Sabah	3
31	Cim	Var	Sabah	3	77	Tarta	Yok	Aksamü	1
32	Tarta	Yok	Aksamüze	1	78	Cim	Var	Sabah	3
33	Cim	Var	Sabah	3	79	Tarta	Var	Aksamü	2
34	Tarta	Var	Aksamüze	2	80	Cim	Var	Sabah	2
35	Cim	Var	Sabah	2	81	Tarta	Var	Aksamü	1
36	Tarta	Var	Aksamüze	1	82	Cim	Var	Sabah	2
37	Cim	Var	Sabah	2					
38	Cim	Var	Sabah	1					
39	Tarta	Yok	Aksamüze	2					
40	cim	Var	Sabah	2					
41	Tarta	Yok	Aksamüze	2					
42	Cim	Var	Sabah	3					
43	Tarta	Var	Aksamüze	1					
44	Cim	Var	Sabah	3					
45	Tarta	Var	Aksamüze	3					
46	Cim	Var	Sabah	3					

Kaynak : Silahtaroğlu, 2013:108

Algoritmanın nasıl hareket edeceği adım adım ele alınırsa:

CHAID Adımları;

Birinci sütunu ele alınırsa;

- Çim ve kum için hesaplama yapılabilir
- Çim ve tartan için hesaplama yapılabilir
- Tartan ve kum için hesaplama yapılabilir
- En büyük p değeri veren değişkenler birleştirilir ve ikinci adımı uygulanır

Örnek olarak çim ile kum en büyük çıktığı düşünülürse

- Çim ve kum + Tartanı hesaplama yapılabilir

İkinci sütun için işlem zaten iki farklı veri tipi olduğundan;

- Var/yok için hesaplama yapılabilir

Zaman sütunu da 3 tip olduğundan 1. Adımdaki gibi;

- Sabah/öğle hesaplama yapılabilir
- Sabah/akşamüzeri hesaplama yapılabilir
- Akşamüzeri/öğle hesaplama yapılabilir
- En büyük p değerlileri birleştirilir.

Birleştirdikten sonra ikili hesaplama yapılır. Üç ayrı sonuçtan en küçük değer ikili düğüm olarak atanmalıdır. Kök(düğüm) olan sütunu veri kümesinden çıkarılır ve ağaç tamamlanana kadar devam edilir.

Yukardaki adımları hesaplama yaparak tekrar edilmek gerekirse, önce birinci sütun ele alınabilir

Tablo 10: Çim Ve Kum İçin Ki-Kare Hesaplamaları

	Sıralama			
Pist	1	2	3	**T**
Çim	9	14	1	37
Tart	15	15	8	38
Top	24	29	2	75

Gözlenen	Beklenen	$\|G - B\|$	$(G - B)^2$	$(G - B)^2/B$
9	9,25	0,25	0,06	0,01
14	13,45	0,55	0,30	0,02
14	14,30	0,30	0,09	0,01
2	1,75	0,25	0,06	0,04
2	2,55	0,55	0,30	0,12
3	2,70	0,30	0,09	0,03
			Kikare	0,22
			p	0,90

Tablo 11: Çim Ve Tartan İçin Ki-Kare Hesaplamaları

	Sıralama			
Pist	1	2	3	**Top**
Çim	9	1	14	37
Kum	2	2	3	7
Topl	1	1	17	44

Gözlenen	Beklene	$\|G - B\|$	$(G - B)^2$	$(G - B)^2/B$
9	11,84	2,84	8,07	0,68
14	14,31	0,31	0,09	0,01
14	10,85	3,15	9,90	0,91
15	12,16	2,84	8.07	0,66
15	14,69	0,31	0,09	0,01
8	11,15	3,15	9,90	0,89
			Kikare	3.16
			p	0,21

Tablo 12: Tartan Ve Kum İçin Ki-Kare Hesaplamaları

	Sıralama			
Pist	1	2	3	**Topl**
Kum	2	2	3	7
Tart	1	15	8	38
Topl	1	17	1	45

Gözlenen	Beklenen	$\|G - B\|$	$(G - B)^2$	$(G - B)^2/B$
2	2,64	0,64	0,42	0,16
2	2,64	0,64	0,42	0,16
3	1,71	1,29	1,66	0,97
15	14,36	0,64	0,42	0,03
15	14,36	0,64	0,42	0,03
8	9,29	1,29	1,66	0,18
			Kikare	1,52
			p	0,47

En büyük *p* değeri veren değişkenler birleştirilir ve ikinci adımı uygulanır; *p* = 0,90 ile en büyük değer çim ve kum arasındadır. Bu durumda iki değer tek değer olarak kullanılabilecektir. Artık çim ve kum diye iki ayrı değer yerine "çim veya kum" adında yeni bir değer oluşabilir ve veri kümesinde hem çim hem kum olan tüm yerlere "çim veya kum" değeri gelebilir.

Tablo 13 : "Çim Veya Kum" Ve Tartan Arasındaki Ki-Kare Değeri Hesaplamaları

	Sıralama			
	1	2	3	
Pist				**To**
ÇimKu	11	16	17	44
Tartan	15	15	8	38
Toplam	26	31	25	82

Gözlene	Beklene	$\|G - B\|$	*(G - B)²*	*(G - B)²/B*
11	13,95	2,95	8,71	0,62
16	16,63	0,63	0,40	0,02
17	13,41	3,59	12,85	0,96
15	12,05	2,95	8,71	0,72
15	14,37	0,63	0,40	0,03
8	11,59	3,59	12,85	1,11
			Kikare	3,47
			p	0,18

İkinci değişken (yağış) için işlem: zaten iki farklı veri tipi olduğundan yalnızca bir aşama yapılabilir.

Tablo 14: Var/Yok İçin Ki-Kare Değeri Hesaplamaları

	Sıralama			
	1	2	3	
Yağı				**Top**
Var	19	2	2	61
Yok	7	1	4	21
Topl	26	3	2	82

Gözlenen	Beklenen	$\|G - B\|$	*(G - B)²*	*(G - B)²/B*
19	19,34	0,34	0,12	0,01
21	23,06	2,06	4,25	0,18
21	18,60	2,40	5,77	0,31
7	6,66	0,34	0,12	0,02
10	7,94	2,06	4,25	0,54
4	6,40	2,40	5,77	0,90
			Kikare	1,95
			p	0,38

Zaman sütunu da 3 değişkenli olduğundan 1. Adımdaki gibi ilk önce Sabah/öğle hesaplama yapılabilir. Daha sonra sabah/akşamüzeri ve öğle/akşamüstü için hesaplamalar yapılabilir. En büyük *p* değeri veren değişkenler birleştirilebilir. En büyük p değerini veren değişkenler birleştirildikten sonra yeniden oluşan iki değişken arasında hesaplamalar yapılabilir.

Tablo 15: Sabah/Öğle Ki-Kare Değeri Hesaplamaları

	Sıralama			
	1	2	3	
Zama				**To**
Sabah	9	15	16	40
Öğle	2	2	3	7
Topla	11	17	19	47

Gözlenen	Beklenen	$\|G$	$(G - B)^2$	$(G - B)^2/B$
9	9,36	0,36	0,13	0,01
15	14,47	0,53	0,28	0,02
16	16,17	0,17	0,03	0,00
2	1,64	0,36	0,13	0,08
2	2,53	0,53	0,28	0,11
3	2,83	0,17	0,03	0,01
			Kikare	0,24
			p	0,89

Tablo 16: Sabah/Akşamüstü Ki-Kare Değeri Hesaplamaları

	Sıralama			
	1	2	3	
Zaman				**Topla**
Sabah	9	1	16	40
Akşamü	1	1	6	35
Toplam	2	2	22	75

Gözle	Beklen	$\|G - B\|$	$(G - B)^2$	$(G - B)^2/B$
9	12,80	3,80	14,44	1,13
15	15,47	0,47	0,22	0,01
16	11,73	4.27	18,20	1,55
15	11,20	3,80	14,44	1,29
14	13,53	0,47	0,22	0,02
6	10,27	4,27	18,20	1,77
			Kikare	5,77
			p	0,06

Tablo 17: Öğle /Akşamüstü Ki Kare Değeri Hesaplamaları

	Sıralam			
	1	2	3	
Zaman				**Topl**
Öğle	2	2	3	7
Akşamü	1	1	6	35
Toplam	1	1	9	42

Gözlenen	Beklenen	$\|G - B\|$	$(G - B)^2$	$(G - B)^2/B$
2	2,83	0,83	0,69	0,25
2	2,67	0,67	0,44	0,17
3	1,50	1,50	2,25	1,50
15	14,17	0,83	0,69	0,05
14	13,33	0,67	0,44	0,03
6	7,50	1,50	2,25	0,30
			Kikare	2,29
			p	0,32

En büyük p değerlileri birleştirilir. Bu değişken için p=0,89 ile en büyük p değeri sabah öğle arasındaki ki-kare olasılık değeridir. Pist değişkeninde olduğu gibi zaman değişkeninde de bu iki değişken birleştirilerek "sabah veya öğle" şeklinde etiketlenebilir.

Tablo 18: "Sabah Veya Öğle" /Akşamüstü Ki-Kare Değeri Hesaplamaları

	Sıralama			
	1	2	3	
Zaman				**Top**
Sabahveya öğle	2	2	3	7
Akşamüze	1	1	6	35
Toplam	1	1	9	42

Gözlenen	Beklenen	$\|G - B\|$	$(G - B)^2$	$(G - B)^2/B$
11	14,90	3,90	15,23	1,02
17	17,77	0,77	0,59	0,03
19	14,33	4,67	21,82	1,52
15	11,10	3,90	15,23	1,37
14	13,23	0,77	0,59	0,04
6	10,67	4,67	21,82	2,04
			Kikare	6,04
			p	0,05

Üç ayrı değişken için değerler şöyledir;

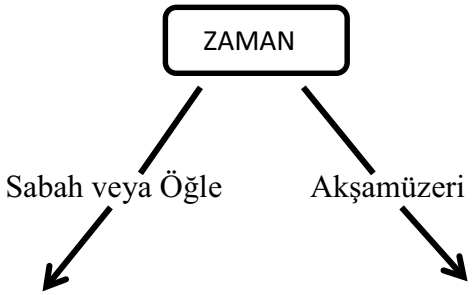
Pist değişkeni("çim veya kum" + tartan) için p = 0,18

Yağış değişkeni için p = 0.38

Zaman değişkeni ("sabah veya öğle" + akşamüzeri) için p = 0,05

Bu üç değişkenden en küçük p değeri (0,05) zaman değişkenin de ortaya çıkmıştır. Bu durumda ilk düğüm zaman değişkeni olur ve karar ağacı aşağıdaki şekilde oluşmaya başlar.

Şekil 5: Karar Ağacı



Kök olan sütunu veri kümesinden çıkarılır ve ağaç tamamlanana kadar bu işlemlere devam edilir. Bu aşamadan sonra sırasıyla "sabah veya öğle" ve akşamüzeri dalları için işlemler benzer şekilde devam ettirilebilir.

2.2.2 Mesafeye Dayalı Sınıflandırma Algoritmaları

Sınıflandırma yapılırken verilerin birbirlerine olan uzaklık-yakınlık durumundan faydalanarak sınıflamanın yapılabilmesi mesafeye dayalı sınıflandırma teknikleri olarak adlandırılır. Veriler arasındaki mesafe genellikle *Euclid* mesafe ölçümü kullanılarak hesaplanır. Mesafeye dayalı algoritmalardan yaygın olarak kullanılanı "*K-en yakın komşu algoritmasıdır"*.

2.2.2.1 K- En Yakın Komşu Algoritması

Örüntü tanıma alanında, "k NearNeighbor" (k-nn) algoritması, yeni bir verinin en yakın (k) komşu verilerine göre etiketlendiği sınıflandırma yöntemi olarak tanımlanabilir(Gorunescu, 2011:256).

Algoritmada *k* değeri önceden seçilir; değerinin yüksek olması birbirlerine benzemeyen noktaların bir araya toplanmasına, çok küçük seçilmesiyse birbirine benzediği, yani aynı sınıfta oldukları halde, bazı noktaların ayrı sınıflara konulmasına ya da o tür noktalar için ayrı sınıfların açılmasına neden olur. Tipik *k* değeri 3.5 ve 7 dir. (Khan ve diğerleri, 2002:517)

Avantajları;

- Gürültüye sahip öğrenme verilerinde de güçlü sonuçlar ortaya koyması,
- Öğrenme verilerinin çok olması durumunda etkili sonuç vermesi
- Kolay anlaşılabilir bir algoritma olduğu için uygulaması basit olması
- Ayrıca gürültüye sahip veriler için de olumlu sonuçlar ortaya koyması

Dezavantajları;

- k sayısının belirlenmesinin gerekliliği,
- Uzaklığa bağlı bir sınıflandırma yöntemi olduğu için hangi uzaklık ölçütünün kullanılacağına dair bir kesinlik olmaması,
- Eğitim verilerinin, öğrenme verilerine olan uzaklıkları tek tek hesaplanacağı için hesaplama zamanının ve masrafının fazla olması
- Tahmin bölgesel bilgilere dayanmaktadır, bu nedenle aşırı değerlerden / aykırı değerlerden etkilenmesi muhtemeldir(Gorunescu, 2011:258).

Algoritmanın çalışma ilkesi aşağıda verilen adımlarla özetlenebilir:

- Uygun bir mesafe ölçüm uzayı belirlenir

  Bazı uzaklık fonksiyonları;

  1- Manhattan Uzaklık Fonksiyon

$$d(i,j)=\left(\left|x_{i1}-x_{j1}\right|+\left|x_{i2}-x_{j2}\right|+\cdots+\left|x_{ip}-x_{jp}\right|\right)$$

  2- Minkowski Uzaklık Fonksiyonu:

$$d(i,j)=\left[\left|x_{i1}-x_{j1}\right|^{m}+\left|x_{i2}-x_{j2}\right|^{m}+\cdots+\left|x_{ip}-x_{jp}\right|^{m}\right]^{1/m}$$

  3- Öklid Uzaklık Fonksiyonu:

$$d(i,j)=\sqrt{\left(x_{i1}-x_{j1}\right)^{2}+\left(x_{i2}-x_{j2}\right)^{2}+\ldots+\left(x_{ip}-x_{jp}\right)^{2}}$$

Uzaklık Fonksiyonun Özellikleri

1. $d(i,j)\geq 0$; Uzaklık negatif değil
2. $d(i,i)=0$; Her birim kendisine olan uzaklığı sıfırdır.
3. $d(i,j)=d(j,i)$; Uzaklık fonksiyonu simetriktir.
4. $d(i,j)\leq d(i,h)+d(h,j)$; iki birimin arasındaki uzaklık bu iki birimin üçüncü bir birime olan uzaklıkları toplamından küçük olamaz (üçgen eşitsizliği)

- Birbirine en yakın k adet nokta belirlenir
- Belirlenen grubun en çok rastlandığı sınıf belirlenir
- Bu gruba belirlenen sınıfın ismi atanır

Tablo 19: k-En Yakın Komşu Algoritma Örneği

X_1	X_2	Y	*Öklid uzaklık fonksiyonu*	*Mesafe*	X_1	X_2	*Mesafe*
2	4	KÖTÜ	$d(i,j)=\sqrt{(2-8)^2+(4-4)^2}$	6	2	4	6
3	6	İYİ	$d(i,j)=\sqrt{(3-8)^2+(6-4)^2}$	5,39	3	6	5,39
3	4	İYİ	$d(i,j)=\sqrt{(3-8)^2+(4-4)^2}$	5	3	4	5
4	10	KÖTÜ	$d(i,j)=\sqrt{(4-8)^2+(10-4)^2}$	7,21	4	10	7,21
5	8	KÖTÜ	$d(i,j)=\sqrt{(5-8)^2+(8-4)^2}$	5	5	8	5
6	3	İYİ	$d(i,j)=\sqrt{(6-8)^2+(3-4)^2}$	2,24	6	3	2,24
7	9	İYİ	$d(i,j)=\sqrt{(7-8)^2+(9-4)^2}$	5,10	7	9	5,10
9	7	KÖTÜ	$d(i,j)=\sqrt{(9-8)^2+(7-4)^2}$	3,16	9	7	3,16
11	7	KÖTÜ	$d(i,j)=\sqrt{(11-8)^2+(7-4)^2}$	4,24	11	7	4,24
10	2	KÖTÜ	$d(i,j)=\sqrt{(10-8)^2+(2-4)^2}$	2,83	10	2	2,83

10 tane veri verilmiş olsun. X_1=8 ve X_2=4 değerleri için en yakın komşu bulma algoritmasını kullanarak hangi sınıfa ait olduğunu bulunmak istenirse ve k=4 olarak alınırsa;

Hesaplamalar yapılıp en yakın 4 nokta alınarak değerlendirilir. Hesaplamalar sonucunda 2.24, 2.83, 3.16, 4.24 sonuçlarının tabloda karşılık geldiği değerler İYİ, KÖTÜ, KÖTÜ, KÖTÜ olarak görünüyor. 3 tane KÖTÜ 1 tane İYİ olduğundan en çok olan sınıfa dâhil edilir. Yani yeni değişkenimiz KÖTÜ sınıfına dâhil olmuş olur.

2.2.2.2 En Küçük Mesafe Sınıflandırıcısı

Bu algoritma, en yakın komşu algoritmasındaki gibi mesafeye bağlı olarak sınıf tahmininde bulunur. Veri tabanındaki her bir sınıfa ait noktanın merkezi ile tahmin edilmesi gereken veriye olan mesafesi göz önüne alınarak işlem yapılır.(Silahtaroğlu,2013:119)

Adımlar;

1 $X_0 = \frac{\sum_{i=1}^{N} x_{\min}}{N}$

2 $dz(x)=X^T X_0 - \frac{1}{2}(X_0^T X_0)$

3 X verisini en büyük sınıfa ata

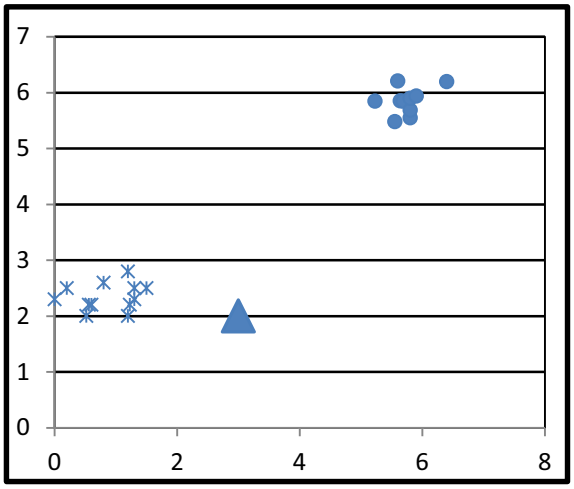
Aşağıdaki tabloda verilen x ve y verilerinin değerlerine göre aldıkları sınıf isimleri Nokta ve Yuvarlak olarak belirlenmiştir. x ve y değerleri {3,2} olan bir K noktasının yuvarlak sınıfına mı yoksa nokta sınıfına mı ait olduğunu En Küçük Mesafe Sınıflandırıcısı algoritmasına göre hesaplamak istenirse(Silahtaroğlu, 2013:120);

Tablo 20: En Küçük Mesafe Sınıflandırma Örneği

x	y	Sınıf	x	y	Sınıf
0,56	2,2	Nokta	5,8	5,55	Yuvarlak
0,8	2,6	Nokta	5,55	5,4	Yuvarlak
1,5	2,5	Nokta	5,6	6,21	Yuvarlak
1,2	2,8	Nokta	5,64	5,854	Yuvarlak
1,23	2,2	Nokta	5,23	5,85	Yuvarlak
1,3	2,3	Nokta	5,69	5,85	Yuvarlak
0	2,3	Nokta	6,4	6,2	Yuvarlak
0,2	2,5	Nokta	5,8	5,69	Yuvarlak
0,52	2	Nokta	5,8	5,9	Yuvarlak
0,6	2,2	Nokta	5,9	5,94	Yuvarlak
1,2	2	Nokta			
1,3	2,5	Nokta			

Kaynak: Silahtaroğlu, 2013:120

Şekil 6: En Küçük Mesafe Sınıflandırma Örneği Grafiği



$$X_0^{NOKTA} = \frac{\sum_{i=1}^{N} x_{min}}{N} = [0.86,2.34]$$

$$X_0^{YUVARLAK} = \frac{\sum_{i=1}^{N} x_{min}}{N} = [5.74,5.84]$$

$$dz(K)^{NOKTA} = X^T X_0 - \frac{1}{2}(X_0^T X_0) = \begin{bmatrix}3\\2\end{bmatrix}[0.86\ 2.34] - \frac{1}{2}\left([0.86 \quad 2.34]\begin{bmatrix}0.86\\2.34\end{bmatrix}\right)$$

$$= 4.15$$

$$dz(K)^{YUVARLAK} = X^T X_0 - \frac{1}{2}(X_0^T X_0) = \begin{bmatrix}3\\2\end{bmatrix}[5.74\ 5.84] - \frac{1}{2}\left([5.74 \quad 5.84]\begin{bmatrix}5.74\\5.84\end{bmatrix}\right)$$

$$= -4.66$$

Bu durumda, *K* noktası 4.15 (en büyük değer) ile "Nokta" sınıfına daha yakın olduğu için "Nokta" sınıfına atanacaktır.

2.2.3 Karar Ağaçları

Karar ağacı yaklaşımı, hedeflenen bilgiyi yaklaşık olarak hesaplamak için kullanılan ve öğrenme fonksiyonunun karar ağacı ile gösterildiği bir yöntemdir. Bir karar ağacı, ağaç görünümünde tanımlayıcı ve tahmin edici bir modeldir. Bu model, karar alıcıya karar alırken hangi faktörlerin göz önüne alınması ve her bir faktörün kararın farklı çıktıları ile geçmişte nasıl ilişkili olduğunun belirlenmesi konularında yardımcı olur (Bounsaythip ve Runsala, 2001:20). Karar ağacına dayalı analizler (www.spss.com/ classification_trees);

- Belirli bir sınıfın olası üyesi olacak elemanların belirlenmesinde (segmentation),
- Çeşitli vakaların yüksek, orta, düşük risk grupları gibi çeşitli kategorilere ayrılmasında (stratification),

- Gelecekteki olayların tahmin edilebilmesi için kurallar oluşturulmasında,
- Parametrik modellerin kurulmasında kullanılmak üzere çok miktardaki değişken ve veri kümesinden faydalı olacakların seçilmesinde,
- Sadece belirli alt gruplara özgü olan ilişkilerin tanımlanmasında,
- Kategorilerin birleştirilmesinde ve sürekli değişkenlerin kesikli değişkene dönüştürülmesinde yaygın olarak kullanılmaktadır.

Karar ağacı algoritmalarında iki temel işlem gerçekleştirilmektedir. Bu işlemler; bölme (splitting) ve budama (pruning) işlemleridir. Algoritmaların sonlanması ise uygulanan durdurma kriterine göre olmaktadır. Bunlar kısaca şöyle açıklanabilir (Bounsaythip ve Runsala, 2001:18-19):

- Bölme: Bu işlem, verilerin daha küçük alt kümelere ayrılmasını içeren yinelemeli bir süreçtir. İlk yineleme tüm veriyi içeren kök düğümünü ele alır. Bundan sonraki yinelemeler, verinin alt kümelerini içeren türev düğümler üzerinde işlem yapmaktadır. Her bölme işleminde, değişkenler analiz edilir ve en iyi bölme seçilir.
- Budama: Bir ağaç oluşturulduktan sonra, istenmeyen alt ağaçlar veya düğümler içerebilir. Budama işlemi ile bunlar ayıklanır. Bir başka ifade ile budama işlemi bir karar ağacını daha genel yapmak için kullanılmaktadır.
- Durdurma kriteri: Ağaç oluşturma algoritmaları çeşitli durdurma kuralları içerirler. Bu kurallar genellikle, maksimum ağaç derinliği, bir düğümde bölme için ele alınan minimum eleman sayısı ve yeni bir düğümde olması gereken minimum eleman sayısı gibi çeşitli faktörlere dayanır(Bounsaythip ve Runsala, 2001:20).

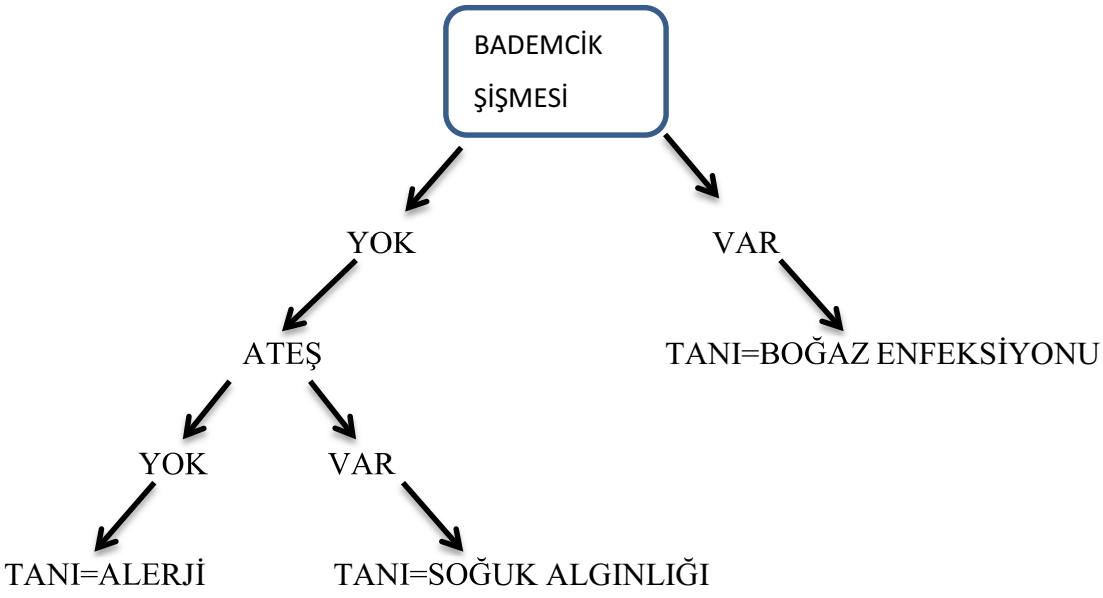
Karar ağaçlarına örnek olarak, Tablo 21'de kategorik verileri içeren bir hasta veri tabanı, Şekil 7'de buna ait bir karar ağacı ve bu ağaçtan elde edilen kurallar verilmiştir.

Tablo 21: Hasta Veritabanı

HASTA SIRA NO.SU	BOĞAZ AĞRISI	ATEŞ	BADEMCİK ŞİŞMESİ	KAN TOPLAMASI	BAŞ AĞRISI	TANI
1	Var	Var	Var	Var	Var	Boğaz Enfeksiyonu
2	Yok	Yok	Yok	Var	Var	Alerji
3	Var	Var	Yok	Var	Yok	Soğuk algınlığı
4	Var	Yok	Var	Yok	Yok	Boğaz Enfeksiyonu
5	Yok	Var	Yok	Var	Yok	Soğuk algınlığı
6	Yok	Yok	Yok	Var	Yok	Alerji
7	Yok	Yok	Var	Yok	Yok	Boğaz Enfeksiyonu
8	Var	Yok	Yok	Var	Var	Alerji
9	Yok	Var	Yok	Var	Var	Soğuk algınlığı
10	Var	Var	Yok	Var	Var	Soğuk algınlığı

Kaynak: Emel ve Taşkın 2005:227

Şekil 7: Hasta Veri Tabanı İçin Bir Karar Ağacı ve Kurallar



Karar ağaçları oluşturulurken hangi algoritmadan yararlanıldığı önemlidir. Çünkü seçilen algoritmaya göre ağacın şekli farklılık göstermekte ve değişik ağaç yapıları ile farklı sınıflandırma sonuçları elde edilmektedir. Kök düğüm değiştiği zaman, en uçtaki yapraklara ulaşmada izlenecek yol da değiştirmekte ve dolayısıyla sınıflandırma da değişmektedir. Kök düğümün belirlenirken veri tabanının eşit parçalara ayrılarak, dallara ayrılmış olması en büyük kıstaslardan biridir. Ağacın alt dallarını oluşturmak için kök düğümden sonra oluşturulacak düğümler belirlenir. Bu düğümlerin oluşturulması aşamasında da kök düğümde uygulanan yöntemler uygulanır. Önceden de ifade edildiği gibi her ağaç kendi sınıflandırma sistemini şekillendireceği için düğümler belirlenirken kullanılan algoritmalar çok önemlidir.

Karar ağacı yöntemi temel alınarak geliştirilen çok sayıda algoritma vardır; bu algoritmalar kök, düğüm ve dallanma kriterlerini tanımlarken takip ettikleri süreç bakımından birbirlerinden ayrılmaktadırlar. Dallanmanın hangi kritere göre yapılacağının belirlenmesi karar ağaçlarının en önemli problemlerinden biridir. Karar ağacı yöntemi temel alınarak kullanılan algoritmalar üç gruba ayrılabilir. Entropiye dayalı algoritmalar, Sınıflandırma ve regresyon ağaçları (CART) ve bellek tabanlı sınıflandırma algoritmaları olarak gruplandırılabilir. ID3 ve onun daha gelişmiş biçimi olan C4.5 ve C5.0 algoritmaları entropiye dayalı bölümlemeyi kullanan algoritmalardır. Bellek tabanlı sınıflandırma yöntemleri arasında k-en yakın komşu algoritması sayılmaktadır (Özkan, 2013:54).

Entropi; beklentisizliğin maksimumlaşmasıdır (Fiske, 2002:12). Aynı zamanda entropi, bir sistemdeki belirsizliğin ölçüsüdür şeklinde de tanımlanabilir. Örneğin elimizdeki bütün verilerin toplamı aynı gruba dahil olduğunda; aynı zamanda aynı sınıfa ait olmuş oluyorlar. Bu gruptan rastgele seçilen bir değişkenin hangi sınıfa dahil olduğunu sorduğumuzda aldığımız cevap bizi şaşırtmayacaktır. Bu durumda entropi 0'dır. Entropi 0 ila 1 arasında değer alır. Bütün olasılıklar eşit olduğunda entropi en büyük değeri olan 1 olacaktır.

Entropinin hesaplanması aşağıda ifade edilmektedir;

S bir kaynak olsun. Bu kaynağın $\{m_1, m_2, \ldots\ldots\ldots, m_n\}$ olmak üzere n adet bildiri ürettiğini varsayalım. Tüm bildiriler birbirlerinden bağımsız olarak üretilmektedir ve m_i bildiri üretilme olasılıkları p_i'dir. $P=\{p_1, p_2, \ldots\ldots\ldots p_i\}$ Olasılık

dağılımına sahip bildirileri üreten S kaynağının entropisi H(S) şeklinde gösterilmektedir. Entropi miktarı formülü; (Shannon, 1948:11).

$$H(S) = \sum_{i=1}^{n} p_i \log(1/p_i)$$

2.2.3.1 ID3 Algoritması

ID3 algoritması diğer değişkenler içerisinde sınıflamada en ayırıcı özelliğe sahip değişkeni bulurken entropi kavramından yararlanır. Entropi kavramı eldeki bilginin sayısallaştırılmasıdır (Dunham, 2003:97).

ID3 algoritmasında nitelikler birbirinden bağımsızdır. Bu algoritmada amaç, ağaç oluşturulurken öğrenme kümesindeki veriler mümkün olduğunca birbirine benzer hale getirilir ve böylece ağaç derinliği minimum seviyede tutulur. Karmaşıklık minimuma indirilir, kazanç maksimum seviyede olur.

ID3 tarafından kullanılan temel strateji, ilk önce en yüksek bilgi kazancı olan bölme niteliklerini seçmektir. ID3 algoritması, bölünmeden önce ve sonra bilgi kazancı arasındaki farktan faydalanarak, en yüksek kazançlı düğümü öncelikli düğüm olarak seçer. Bu durum bölünmüş büyük miktardaki verinin ihtiyaç duyduğu bilgiyi azalmaktadır. Orijinal veri setinin entropileri ile alt bölümlenmiş veri setlerinin her birinin entropilerin ağırlıklı toplamları arasındaki farklar belirlenir. Fark alt bölümlerden hangisi için büyükse dallanma o bölüm üzerinden ilerler. Kazanç formülü;

$$Kazanç(D;S) = H(D) - \sum_{i=1}^{n} P(D_i)\, H(D_i)$$

$H(D)$ genel durum entropisi, $P(D_i)$ her bir alt niteliğin veri tabanında bulunma olasılığı, $H(D_i)$ alt niteliğin entropisidir.

ID3 kullanım Avantajları;

- Anlaşılabilir tahmin kuralları eğitme verileri tarafından oluşturulur.
- En hızlı ve Kısa ağacı oluşturur.
- Sadece yeterli nitelikler bütün veriler sınıflandırılana kadar gereklidir.
- Yaprak düğümleri bulmak test sayılarını azaltmak için test verilerini budamayı mümkün kılar.

- Bütün veri kümesi ağaç oluşturmak için araştırılır.

ID3 kullanımı Dezavantajları

- Eğer küçük örnekler test edildiyse veri başarısız çıkmıştır veya çok doludur.
- Aynı zamanda sadece bir yükleme test edilebilir bir karar verilirken veri sınıflandırması çok pahalıya gelebilir ve bunu engellemek için sürekli bir kırılmaya ihtiyaç vardır.

Tablo 22: ID3 algoritması örnek verileri

CİNSİYET	KİLO	BOY	BEDEN
K	48	170	ORTA
K	49	151	KÜÇÜK
K	52	158	ORTA
K	56	165	ORTA
E	59	160	KÜÇÜK
K	61	159	ORTA
E	62	162	KÜÇÜK
E	63	174	ORTA
K	68	168	ORTA
K	69	177	BÜYÜK
E	72	170	ORTA
E	74	165	KÜÇÜK
E	85	175	ORTA
E	85	190	BÜYÜK
E	98	190	BÜYÜK

Kaynak: Silahtaroğlu, 2013:71

Yukarıda tablodaki veriler kullanarak bir karar ağacı oluşturulmak istenirse kök düğüm ID3 algoritmasına göre hesaplanmak istenirse(Silahtaroğlu, 2013:76);

Toplam 15 adet veri ve hedef olarak beden belirlenirse eğer örnek verilerden 4 tanesi küçük, 8 tanesi orta ve 3 tanesi büyük beden sınıfında olduğu görülmektedir. İlk başta genel durum entropisini hesaplanır;

$H(beden) = \sum_{i=1}^{n} p_i \log\left(\frac{1}{p_i}\right)$ =4/15 log(15/4)+8/15 log(15/8)+3/15 log(15/3)

= 0,4384

Cinsiyet değişkeni için entropi hesaplanırsa;

H(kadın) = 1/7 log(7/1)+5/7 log(7/5)+1/7 log(7/1) = 0.3458

H(erkek) = 3/8 log(8/3)+ 3/8 log(8/3)+ 2/8 log(8/2) = 0.4700

Ağırlıklı toplam = (7/15)x0.3458+(8/15)x0.4700 = 0.4120

Cinsiyet için kazanç = 0.4384 – 0.4120 = 0.0264

BOY	KİLO
151-159.99 arası 1. Grup	[45-55,99] arası 1. Grup
160-164.99 arası 2. Grup	[56-64,99] arası 2. Grup
165-174.99 arası 3. Grup	[65-75,99] arası 3. Grup
175-184.99 arası 4. Grup	[76-85,99] arası 4. Grup
185- üzeri de 5. Grup	[86-100] arası 5. Grup

Tablo 23: Verilerin Gruplara Ayrılması

CİNSİYET	KİLO	BOY	BEDEN
K	1	3	ORTA
K	1	1	KÜÇÜK
K	1	1	ORTA
K	2	3	ORTA
E	2	2	KÜÇÜK
K	2	1	ORTA
E	2	2	KÜÇÜK
E	2	3	ORTA
K	3	3	ORTA
K	3	4	BÜYÜK
E	3	3	ORTA
E	3	3	KÜÇÜK
E	4	4	ORTA
E	4	5	BÜYÜK
E	5	5	BÜYÜK

Aynı işlemler kilo ve boy için yapıldığında;

1. Grup kilo için entropi = 1/3 log(3/1) + 2/3 log(3/2) =0,2764
2. Grup kilo için entropi = 2/5 log(5/2) + 3/5 log(5/3) =0,2923
3. Grup kilo için entropi = 1/4 log(4/1) + 2/4 log(4/2) + 1/4 log(4/1) = 0,4515
4. Grup kilo için entropi = 1/2 log(2/1) + 1/2 log(2/1) = 0,3010
5. Grup kilo için entropi = 1/1 log(1/1) = 0

Ağırlıklı toplam = (3/15x0,2764) + (5/15x0,2923) + (4/15x0,4515) + (2/15x0,3010) +(1/15x0)

= 0,3133

Kilo için kazanç = 0.4384 – 0.3133 = 0.1252

1. Grup boy için entropi = 1/3 log(3/1) + 2/3 log(3/2) = 0,2764
2. Grup boy için entropi = 2/2 log(2/2) = 0
3. Grup boy için entropi = 1/6 log(6/1) + 5/6 log(6/5) = 0.1957

4. Grup boy için entropi = 1/2 log(2/1) + 1/2 log(2/1) = 0,3010
5. Grup boy için entropi = 2/2 log(2/2) = 0

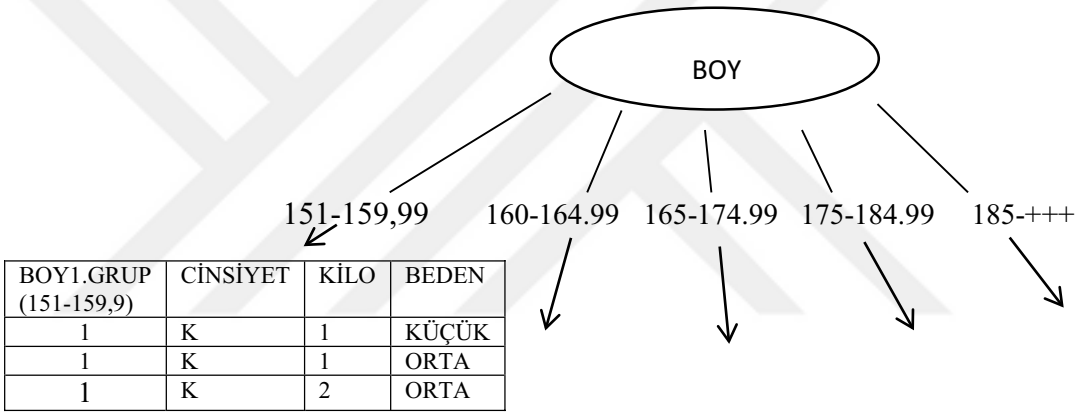
Ağırlıklı toplam = (3/15x0,2764) + (2/15x0) + (6/15x0,1957) + (2/15x0,3010) + (2/15x0)

= 0,1737

Boy için kazanç = 0.4384 – 0.1737 = 0.2648 olacaktır.

Bu değerlerden en büyük olanı boy (Kazanç = 0,2648) değişkeni kök (ilk düğüm) olarak seçilecektir.

Şekil 8: ID3 Örnek Veri Karar Ağacı



BOY1.GRUP (151-159,9)	CİNSİYET	KİLO	BEDEN
1	K	1	KÜÇÜK
1	K	1	ORTA
1	K	2	ORTA

H(Beden) = 1/3 log(3/1) + 2/3 log(3/2) = 0,276

Cinsiyet için kazanç;

H(kadın/beden) = 1/3 log (3/1) + 2/3 log (3/2) = 0,276

Cinsiyet için kazanç = 0,276 – 0,276 = 0

Kilo için;

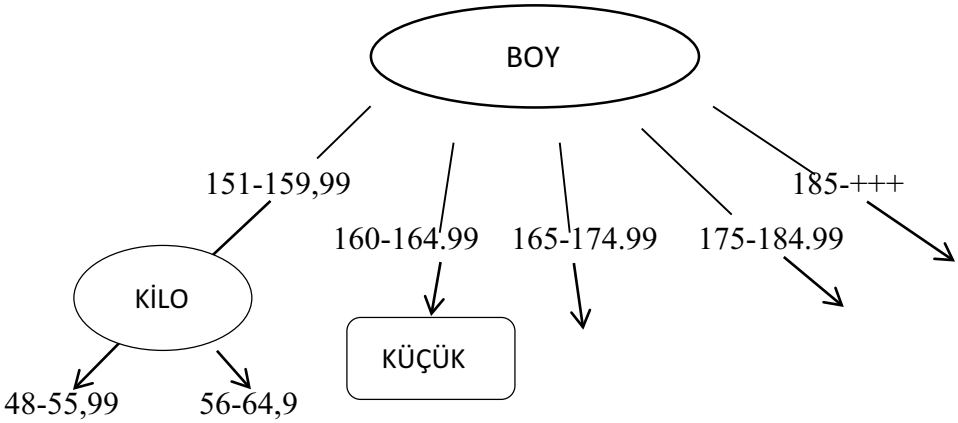
H(kilo/1) = 1/2 log(2/1) + 1/2 log(2/1) = 0,301

H(kilo/2) = 1/1 log(1/1) =0

Ağırlıklı toplam = 2/3x0,301 + 0x1/3 = 0,200

Kazanç (kilo/beden) = 0,276 – 0,200 = 0,076

Şekil 9: ID3 Örnek Veri Karar Ağacı-2



2.2.3.2 C4.5 ve C5.0 Algoritması

C4.5 algoritması ID3 algoritmasına geliştirilmesi ile oluşturulmuş bir algoritmadır. ID3 algoritması ile Karar ağacı oluşturulduğunda, eksik veriler basit bir şekilde yok sayılır. C4.5 algoritması, eksik verileri diğer verilerin özellik değerlerine dayanarak tahmin ederek bir karar ağacı oluşturabilir. Temel fikir, eğitim örneğinde bulunan verilerin özellik değerlerine dayanarak verileri aralıklara bölmektir (Dunham, 2003:100).

C5.0 algoritması ise birçok veri madenciliği paketinde yaygın olarak kullanılan bir C4.5 algoritmasının ticari sürümüdür. Büyük veri kümeleriyle kullanıma yöneliktir. C4.5'e yakın, ancak kural oluşturma farklı ve C4.5'e göre daha geliştirilmiştir. Sonuçlar, C5.0'ın bellek kullanımında çok iyi olduğunu, C4.5'ten daha hızlı çalıştığını ve daha doğru kurallar ürettiğini göstermektedir.

ID3 algoritması verileri çok fazla alt bölüme ayırır. Bu da aşırı öğrenmeye sebep olabilir. Aşırı öğrenmeye önlemek için C4.5 algoritması ile kazanç yerine kazanç oranını kullanmaktadır.

Kazanç oranı (D,S) = Kazanç(D,S) / Bölünme Bilgisi (D,S)

Bölünme bilgisi (D,S) = $H\left(\frac{|D_1|}{|D|}, \ldots, \frac{|D_S|}{|D|}\right)$

Maksimum kazanç oranına sahip olan nitelik, bölme niteliği olarak seçilir.

Yukarıda ID3 için verdiğimiz örnek verileri kullanılarak C4.5 ve C5.0 algoritması ile hesaplama yapılmak istenirse;

Genel durum entropisi yukarıda da hesaplanıldığı gibi 15 adet veri ve bu verilerin 4 tanesi küçük, 8 tanesi orta ve 3 tanesi büyük sınıfında olması üzerinden;

$H(beden) = \sum_{i=1}^{n} p_i \log\left(\frac{1}{p_i}\right)$ =4/15 log(15/4)+8/15 log(15/8)+3/15 log(15/3)

= 0,4384

Cinsiyet içi bölünme bilgisi;

$H(D_{kadın}$;$D_{erkek})$=7/15 log*15/7+8/15* log*15/8 = 0.3001*

Cinsiyet için kazanç *= 0.4384 – 0.4120 = 0.0264*

Cinsiyet için kazanç oranı = 0,0264/0.3001= <u>0,0880</u>

Kilo için bölünme bilgisi;

H($D_{1.grup}$; $D_{2.grup}$; $D_{3.grup}$; $D_{4.grup}$; $D_{5.grup}$) =3/15 log(15/3) + 5/15 log(15/5) + 4/15 log(15/4) + 2/15 log(15/2) + 1/15 log(15/1) = 0,6470

Kilo için kazanç = 0.4384 – 0.3133 = 0.1252

Kilo için kazanç oranı=0,1252/0,6470 = <u>0,1935</u>

Boy için bölünme bilgisi;

H($D_{1.grup}$; $D_{2.grup}$; $D_{3.grup}$; $D_{4.grup}$; $D_{5.grup}$) = 3/15 log(15/3) + 2/15 log(15/2) + 6/15 log(15/6) + 2/15 log(15/2) + 2/15 log(15/2) = 0,6490

Boy için kazanç = 0.4384 – 0.1737 = 0.2648 olacaktır.

Boy için kazanç oranı= 0,2648/0,6490 = <u>0.4080</u>

Elde edilen kazanç oranlarından *maksimum kazanç oranına* sahip olan nitelik kök (boy değişkeni) olarak atanır.

2.2.3.3 CART (Classification and Regression trees) Algoritması

Sınıflandırma ve regresyon ağaçları (CART) ikili karar ağacı üreten bir sınıflandırma algoritması tekniğidir. CART algoritması da ID3 algoritmasaı gibi, entropi en iyi bölme özelliğini ve kriterini seçmek için ölçü olarak kullanılır. Bölme, en iyi bölme noktası olduğu belirlenen nokta etrafında gerçekleştirilir (Dunham, 2003:102).

Algoritma dallara ayırmada hem uygun değişkeni bulur hem de bu değişken ikiden fazla farklı türde değerler taşıyorsa hangi şekilde iki ayrı gruba ayrılacağını da belirler. Bu işlemi aşağıdaki parametreleri kullanarak gerçekleştirir.

CART kullanım Avantajları

- Parametrik değildir. Bağımsız değişken tahminine ihtiyaç yoktur. Bu sebeple CART analizinde çarpık sayısal değişkenler, sınıflayıcı veya sıralayıcı yapıya sahip kategorik değişkenler kullanılabilmektedir.
- CART, dallara ayırma kriterini hesaplarken kayıp eksik verileri önemsemez.
- CART algoritması, yorumlanması çok kolay bir algoritmadır.

CART kullanım Dezavantajları

- CART yazılım paketlerinde standart bir analiz tekniği olmadığından yer almamaktadır.
- İşlem zamanı diğer karar ağacı algoritmalarına göre uzundur.
- iki değerli ağaç tekniğini kullanması dezavantaj olarak düşünülebilir.

Herhangi bir *t* düğümündeki *s* dallara ayrılma kriteri Ψ(*s/t*) olarak gösterilirse;

$\Psi(s/t)= 2P_L P_R \sum_{J=1}^{M} \left|P(C_J/t_L) - P(C_J/t_R)\right|$

t: Dallanmanın yapılacağı düğüm *c:* Kriter

L: Ağacın sol tarafı *R:* ağacın sağ tarafı

$P_{L,}\ P_R$: Öğrenim kümesinde bir kaydın sağda veya solda olma olasılığı

$P_{L=} \dfrac{t_L \text{ daki her bir nitelik değerinin ilgili nitelik sütunundaki tekrar sayısı}}{\text{Eğitim kümesindeki kayıtların sayısı}}$

$P(C_J / t_L)$ *ve* $P(C_J / t_R)$: C_J sınıfındaki bir kaydın solda veya sağda olma olasılığı

$$P(C_J/t_L)= \frac{t_L \text{ daki kayıtların } C_J \text{ sınıfları sayısı}}{t_L \text{ daki her bir nitelik değerinin ilgili nitelik sütunundaki tekrar sayısı}}$$

Hesaplanan Ψ(*s/t*) değerleri içinden en büyük değere sahip nitelik düğüm olarak seçilir

ID3 örneğindeki verileri CART algoritmasıyla karar ağacına dönüştürülmek istenirse(Silahtaroğlu, 2013:83);

$\Psi(s/t)= 2P_L P_R \sum_{J=1}^{M} \left|P(C_J/t_L) - P(C_J/t_R)\right|$

$\Psi(cinsiyet) =2 \left(\frac{7}{15}\right)\left(\frac{8}{15}\right)\left[\left|\frac{1}{15}-\frac{3}{15}\right| + \left|\frac{5}{15}-\frac{3}{15}\right| + \left|\frac{1}{15}-\frac{2}{15}\right|\right] =0.166$

Kilo için;

$\Psi(kilo\text{-}1) = 0$

$$\Psi(kilo\text{-}2) = 2\left(\frac{3}{15}\right)\left(\frac{12}{15}\right)\left[\left|\frac{1}{15}-\frac{3}{15}\right|+\left|\frac{2}{15}-\frac{6}{15}\right|+\left|\frac{0}{15}-\frac{3}{15}\right|\right] = 0.192$$

$$\Psi(kilo\text{-}3) = 2\left(\frac{8}{15}\right)\left(\frac{7}{15}\right)\left[\left|\frac{3}{15}-\frac{1}{15}\right|+\left|\frac{5}{15}-\frac{3}{15}\right|+\left|\frac{0}{15}-\frac{3}{15}\right|\right] = 0.232$$

$$\Psi(kilo\text{-}4) = 2\left(\frac{12}{15}\right)\left(\frac{3}{15}\right)\left[\left|\frac{4}{15}-\frac{0}{15}\right|+\left|\frac{7}{15}-\frac{1}{15}\right|+\left|\frac{1}{15}-\frac{2}{15}\right|\right] = 0.235$$

$$\Psi(kilo\text{-}5) = 2\left(\frac{14}{15}\right)\left(\frac{1}{15}\right)\left[\left|\frac{4}{15}-\frac{0}{15}\right|+\left|\frac{8}{15}-\frac{0}{15}\right|+\left|\frac{2}{15}-\frac{1}{15}\right|\right] = 0.108$$

Boy için;

$\Psi(boy\text{-}1) = 0$

$$\Psi(boy\text{-}2) = 2\left(\frac{3}{15}\right)\left(\frac{12}{15}\right)\left[\left|\frac{1}{15}-\frac{3}{15}\right|+\left|\frac{2}{15}-\frac{6}{15}\right|+\left|\frac{0}{15}-\frac{3}{15}\right|\right] = 0.192$$

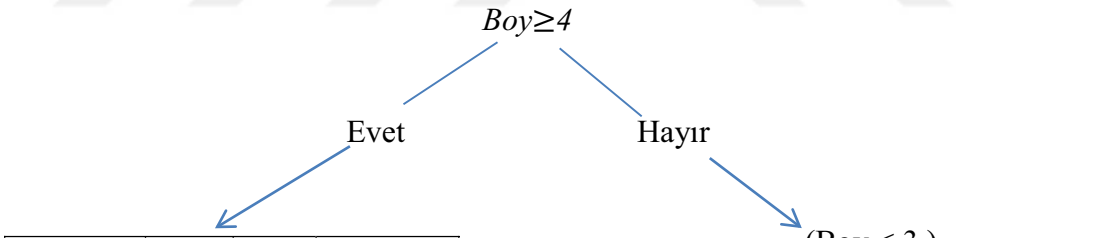
$$\Psi(boy\text{-}3) = 2\left(\frac{5}{15}\right)\left(\frac{10}{15}\right)\left[\left|\frac{1}{15}-\frac{3}{15}\right|+\left|\frac{2}{15}-\frac{6}{15}\right|+\left|\frac{0}{15}-\frac{3}{15}\right|\right] = 0.267$$

$$\Psi(boy\text{-}4) = 2\left(\frac{11}{15}\right)\left(\frac{4}{15}\right)\left[\left|\frac{4}{15}-\frac{0}{15}\right|+\left|\frac{7}{15}-\frac{1}{15}\right|+\left|\frac{0}{15}-\frac{3}{15}\right|\right] = \mathbf{0.339\ (kök\ düğüm)}$$

$$\Psi(boy\text{-}5) = 2\left(\frac{13}{15}\right)\left(\frac{2}{15}\right)\left[\left|\frac{4}{15}-\frac{0}{15}\right|+\left|\frac{8}{15}-\frac{0}{15}\right|+\left|\frac{1}{15}-\frac{2}{15}\right|\right] = 0.200$$

Görüldüğü gibi yine boy değişkeni kök düğüm olarak çıkmıştır. Fakat bu kez soru *boy≥4* mü şeklinde olması gerekir;

Şekil 10: CART Algoritması Karar Ağacı



Cinsiyet	Kilo	Boy	Beden
K	3	4	BÜYÜK
E	4	4	ORTA
E	4	5	BÜYÜK
E	5	5	BÜYÜK

(Boy < 3)

Cinsiyet	Kilo	Boy	Beden
K	1	3	ORTA
K	1	1	KÜCÜK
K	1	1	ORTA
K	2	3	ORTA
E	2	2	KÜCÜK
K	2	1	ORTA
E	2	2	ORTA
E	2	3	ORTA
K	3	3	ORTA
E	3	3	ORTA
E	3	3	KÜCÜK

2.2.3.4 SPRINT (Scalable Parallelizable Induction of Decision Trees Algoritması

SPRINT (Scalable Parallelizable Induction of Decision Trees) algoritması genişlik ilkesiyle çalışmaktadır. SPRINT' algoritması, çok geniş bir eğitim setinden karar ağaçları oluşturabilen, kategorik ve sürekli verileri de işleyebilen bir algoritmadır.

SPRINT ilk olarak her bir değişken için ayrı bir değişken listesi hazırlar. Her tabloda kullanılacak değişkenler için sınıf ve sıra numaraları bulunur. Bu durumda veri tabanındaki değişken sayısı kadar tablo oluşur. Sürekli değer taşıyan tablolar sürekli değer değişkenine göre sıraya dizilirken kategorik veriler sıra numaralarına göre dizilecektir. Sınıf listesinin büyüklüğü eğitim setindeki dizilerin numarası ile orantılı büyür. SPRINT, tüm bellek kısıtlamalarını ortadan kaldırmasına rağmen, eğitim setiyle orantılı bir karma ağacının kullanılmasını gerektirmektedir. Ağaçlar git gide büyüyerek düğümlerin yeni dallar oluşturması ile birlikte her düğümün değişken grupları da bölünerek yeni dallara ayrılır. SPRINT ile en iyi ayrımı bulmak için bir gini indeksi kullanılır (Silahtaroğlu, 2013:86).

Herhangi bir k kümesinin *gini (K)* indeksi aşağıdaki gibi hesaplanır;

$$gini(K) = 1 - \sum p_j^2$$

Burada p_j, K kümesinde *j* sınıfının sıklığıdır. Eğer K kümesi K_1 ve K_2 gibi alt kümelere bölünmüşse, $gini_{bölünmüş}(K)$değeri;

$$gini_{bölünmüş}(K) = \sum_{i=1}^{t} \frac{n_i}{n} gini(K_i)$$

şeklinde hesaplanabilir.

ID3 örneğindeki veriler ile SPRINT algoritması hesaplanmak istenirse(Silahtaroğlu, 2013:88);

Cinsiyet için *gini* değeri hesaplanırsa;

$gini(Kadın)$=1- $[(1/7)^2+(5/7)^2+(1/7)^2]$ =0.4490

$gini(Erkek)$=1- $[(3/8)^2+(3/8)^2+(2/8)^2]$ =0.6563

$gini_{bölünmüş}(Cinsiyet)$=(7/15)x0.4490+(8/15)x0.6563= 0.5596

Kilo için *gini* değeri;

$gini(1)$=1- $[(1/3)^2+(2/3)^2+(0/3)^2]$= 0.4444

$gini(2)$=1- $[(2/5)^2+(3/5)^2+(0/5)^2]$=0.4800

$gini(3)$=1- $[(1/4)^2+(2/4)^2+(1/4)^2]$=0.6250

$gini(4)$=1- $[(0/2)^2+(1/2)^2+(1/2)^2]$=0.5000

$gini(5)$=1- $[(0/1)^2+(0/1)^2+(1/1)^2]$=0.0000

$gini_{bölünmüş}(kilo)$=(3/15)x0.4444+(5/15)x0.4800+(4/15)x0.6250+(2/15)0.5000

= 0.4822

Boy için;

$gini(1)$= 1-$[(1/3)^2 +(2/3)^2 + (0/3)^2]$ = 0.4444

$gini(2)$= 1-$[(2/2)^2 + (0/2)^2 + (0/2)^2]$ = 0.0000

$gini(3)$= 1-$[(1/6)^2 + (5/6)^2 + (0/6)^2]$ = 0.2449

$gini(4)$= 1-$[(0/2)^2 + (1/2)^2 + (1/2)^2]$ = 0.5000

$gini(5)$= 1-$[(0/2)^2 + (0/2)^2 + (2/2)^2]$ = 0.0000

$gini_{bölünmüş}(boy)$=(3/15)x0,4444+ (2/15)x0 + (7/15)x0,2449 + (2/15)x0,5 + (1/15)x0

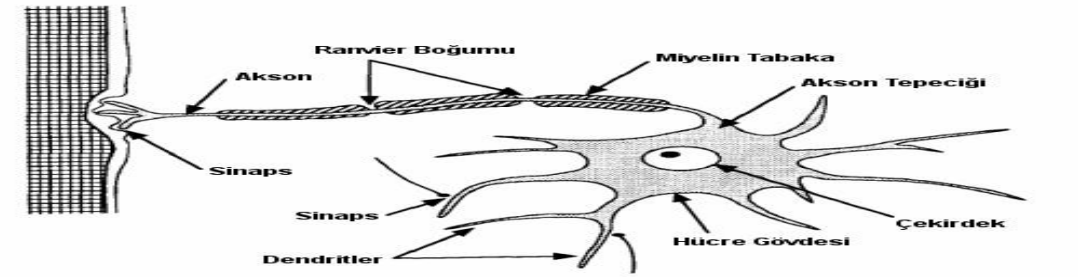
= 0,2698

En küçük $gini_{bölünmüş}$ değerine sahip olan *boy* değişkeni kök düğüm olarak seçilir.

2.2.4 Yapay Sinir Ağları

Yapay sinir ağları (YSA) insan vücudundaki sinir siteminin çalışmasından esinlenerek geliştirilmiş sistemidir. Yapay sinir ağları, genel olarak katmanlar şeklinde düzenlenmektedir. YSA'ları, bağlı nöronlar, bağlantılar arası mesafenin ölçülmesi ve ateşleme fonksiyonu özellikleri en belirgin özellikleri arasındadır. Şekil 11'de biyolojik sinir ağının yapısı görülmektedir.

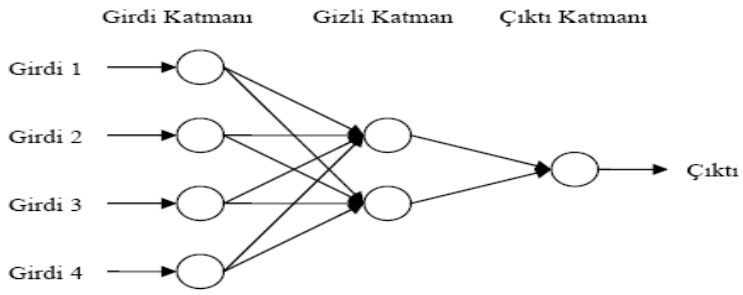
Şekil 11: Biyolojik Sinir Ağı Örneği



Yapay sinir ağlarını meydana getiren her nöronun bir iç hali vardır. Bu iç hale aktivasyon veya aktivasyon seviyesi denir. Çok katmanlı beslemeli bir sinir ağı üzerinde öğrenme gerçekleştirir. Yinelemeli olarak, sınıflardaki etiketlerin

öngörülmesi için bir takım ağırlıklar öğrenilir. Çok katmanlı ileri beslemeli bir sinir ağı, bir giriş katmanından, bir veya daha fazla gizli katmandan ve bir çıkış katmanından oluşur. Nöronlar bir seferde tek bir işaret gönderebilirler. Sinyaller gönderilen fonksiyon için giriş fonksiyonu olur. Şekilde üç katmanlı sinir ağı örneği gösterilmektedir. Bu katmalar girdi, gizli ve çıktı katmanlarıdır.

Şekil 12: Yapay Sinir Ağı Katmanları



Eğitim gruplarında ölçülen her özellik için girdi değerlerini normalleştirmek, öğrenme aşamasını hızlandırmaya yardımcı olacaktır. Genellikle, giriş değerleri 0 ile 1 arasında olacak şekilde normalleştirilir. İki katmalı yapay sinir ağlarında gizli katman bulunmaz. Son katman çıktı katmanıdır. Gizli katmanın aktif hale gelebilmesi için fonksiyonun değerinin belirli bir eşiğin üzerinde olması gerekmektedir.

Yapay sinir ağları sınıflandırma için kullanıldıkları zaman en basit ifadesiyle algoritma bir öğrenme sürecini sürdürür. Bu öğrenme sürecinde çıktı katmanına ulaşabilmek için *w* ağırlıklarını hesaplamaya çalışır. Öğrenme için ayrılmış veri kümesi üzerinde bu ağırlıklarını hesapladıktan sonra, eldeki verilerin diğer kısmını kullanarak öğrenmenin ne kadar gerçekleştiğini bulmak için ağırlıklarını test eder. Eğer test sonucunda bulunan ağırlıkların etkinliği doğrulanırsa, algoritma öğrenme işlemini tamamlamış olur. Aksi durumda *w* ağırlıkları üzerinde düzeltme ya da yeniden değer hesaplama işlemleri yapılır. Öğrenme işlemi tamamlandıktan sonra verilen herhangi bir yeni verinin, eldeki ağırlıklar yardımıyla bağlı olduğu sınıf hesaplanabilir. Yapay sinir ağlarının öğrenmesi uzun zaman alır; ancak bir kez öğrendikten sonra çok duyarlı bir şekilde sınıflandırma yapılabilir. Yapay sinir ağlarının en olumsuz tarafı ise bu ağırlıkların neden ilgili değerleri aldıklarının bilinmemesidir

Yapay sinir ağları tıpkı insanların yaptığı gibi deneyimlerden bilgi çıkarma işlemini yapmaktadır. Çalışma tasarımı deneme yanılma işlemidir ve sonuçta ortaya

çıkan eğitimli ağın doğruluğu etkilenebilir. Ağırlıkların başlangıç değerleri ayrıca ortaya çıkan doğruluğu da etkileyebilir. Bir ağ eğitildiğinde ve doğruluğu kabul edilebilir olarak kabul edilmediğinde, eğitim sürecini farklı bir ağ topolojisi veya farklı bir başlangıç ağırlık seti ile tekrarlamak yaygındır(Han Ve Kamber, 2001:329)

YSA Avantajları

- Düşük bir hata oranı olduğu durumlarda bile uygun bir eğitim verileri alındığında yüksek bir doğruluk derecesi vardır.
- Yapay sinir ağları gürültülü verilerle de iyi sonuçlar verir.
- Sayısal verilerle olduğu gibi kategorik verilerle de çalışılabilmektedir.
- Doğru sınıflandırma oranı genelde yüksektir.
- Farklı alanlaraa iyi uyum gösterir.

YSA Dezavantajları

- Yapay sinir ağlarının anlaşılması zor. Teknik bilgisi olmayan kullanıcılar, YSA'larını nasıl çalıştığını anlamada zorluk yaşayabilir.
- Girdi niteliği değerleri sayısal olmalıdır. Bu verilerin 0 ile 1 arasında olması gerekmektedir.
- Yapay sinir ağlarından kural oluşturmak kolay değildir.
- Öğrenme aşaması uzun sürebilir ve aşırı öğrenme ile sonuçlanabilir.
- YSA'larının kullanımı oldukça maliyetli olabilir.

2.3 KÜMELEME ANALİZİ

Kümeleme analizi, sınıflandırma algoritmalarında yapıldığı gibi var olan verileri gruplara ayırma işlemidir. Sınıflandırma işleminde sınıflar önceden belirli iken kümelemede sınıflar önceden belirli değildir. Verilerin hangi kümelere, hatta kaç değişik kümeye ayrılacağı eldeki verilerin birbirlerine olan benzerliğine göre belirlenir. Kümeleme yöntemlerinin çoğu veriler arasındaki uzaklıkları kullanır.

Kümeleme yapılırken amaç, verilerden yararlanarak niteliklerin birbirlerine olan benzerlik ve farklılıklarına göre gruplara bölünmesidir. Kümeleme analizinde aynı grup elemanlarının olabildiğince birbirine benzer yani homojen, farklı grup elemanlarının birbirinden farklı yani heterojen olması istenmektedir (Şekeroğlu, 2010:62).

Kümeleme analizi, biyoloji, tıp, antropoloji, pazarlama, ekonomi ve telekomünikasyon gibi birçok alanda kullanılmaktadır. Bilgisayar bilimlerinde; desen tanımlama, resim işleme, uzaysal harita verilerinin analizinde kullanılmaktadır. Ayrıca internet üzerinde web sayfalarının aranması, DNA analizi, coğrafi bilişim sistemi gibi alanlarda da kullanılmaktadır.

Kümeleme yöntemleri için birçok farkı algoritma türetilmiştir. Algoritmanın seçimi, analiz edilecek verinin içeriğine göre saptanmaktadır. Kümeleme yöntemleri genel olarak aşağıdaki gibi sınıflandırılmaktadır.

Toplaşım kümeleme algoritmaları ve bölünür kümeleme algoritmaları hiyerarşik yöntemler olarak sınıflandırılabilir.

K-medoids yöntemler, k-means yöntemler ve yoğunluğa dayalı algoritmalar bölümlemeli yöntemler olmak üzere sınıflandırılmaktadırlar.

Genetik algoritmalar, kısıtlara dayanan yöntemler ve makine öğrenmesi Alanında Kullanılan Yöntemler Grid Temelli (ızgara taabanlı) yöntemler olmak üzere sınıflanmaktadır.

2.3.1 Hiyerarşik Yöntemler

Hiyerarşik kümeleme yöntemleri, kümelerin ilk başta toplu bir şekilde tek bir küme olarak ele alınması ve daha sonra kademeli olarak alt kümelere bölünmesi veya ayrı ayrı ele alınan kümelerin kademeli olarak tek küme olarak birleştirilmesi esasına dayanmaktadır (Özkan, 2008:136). Toplaşım kümeleme algoritmaları ve bölünür kümeleme algoritmaları olmak üzere iki ana başlıkta toplanabilir.

Toplaşım kümeleme algoritmalar, her bir öğenin ayrı kümesinin olmasıyla başlar ve tüm öğeler bir kümeye ait oluncaya kadar kümeleri yinelemeli olarak birleştirir. Bölünür kümeleme ise, tüm öğeler başlangıçta bir kümeye yerleştirilir ve kümeler tüm kümeler kendi kümelerine gelinceye kadar art arda ikiye bölünür.

Bölünür kümeleme algoritmaları ise kümeleri birkaç adımın aksine bir adımda oluşturur. Tüm verileri tek bir küme olarak kabul eder ve kullanıcının belirlediği k adet küme oluşturarak veriler arasındaki farklılıklara göre verileri bu k adet kümeye dağıtılır.

Hiyerarşik kümeleme aşağıdaki özelliklere sahiptir:

- Veri tabanını bir kaç kümeye bölünebilir.
- Bu ayrıştırma dendogram (hiyerarşik kümeleme analizinden çıkan tablolara ve ağaçlara verilen isim) adında bir ağaç tarafından yapılır.
- Bu ağaç, gövdeden yapraklara ya da yapraklardan gövdeye doğru oluşturulabilir.

Aşağıdan-yukarıya yaklaşım (toplaşım (agglomerative)) hiyerarşik kümeleme şu şekildedir:

- Her bir nesne için farklı bir grup oluşturarak başla,
- Bazı kurallara göre grupları birleştir: örn.; merkezler arasındaki uzaklık,
- Bir sonlandırma durumuna ulaşılıncaya kadar devam et.

Yukarıdan aşağıya yaklaşımı (bölünür (divisive)):

- Aynı kümedeki bütün nesnelerle başla,
- Bir kümeyi daha küçük kümelere böl,
- Bir sonlandırma durumuna ulaşılıncaya kadar devam et.

2.3.1.1 SLINK Algoritması ve Tek Bağlantı Tekniği

Slink algoritması tek bağlantı ya da en yakın komşu tekniğini kullanmaktadır Bu teknikte, kümeler arasındaki mesafe ölçülürken, iki küme içinde birbirine en yakın iki elemanın uzaklığı ya da başka bir deyişle iki kümeyi en yakın kılan elemanların mesafesi kümeler arası mesafe olarak kabul edilir (Silahtaroğlu, 2013:164).

İlk başta var olan verilerin mesafe/benzerlik matrisi oluşturulur; oluşturulan bu matrisi ağaca dönüştürür. SLINK algoritması k-en yakın komşu algoritmasının bir türevidir. Bu teknik literatürde en yakın komşu kümesi olarak adlandırılmaktadır (Dunham, 2003:134).

Aşağıdaki tabloda bir bankanın kredi kartları bölümünden müşterilerin aylık toplam alışveriş ortalamalarıyla kredi kartı üst limitleri görülmektedir. SLINK algoritması kullanılarak bu müşterileri kümelere ayırmak istenmektedir

Tablo 24: Kredi Kartı Verileri

Müşteri	Aylık Toplam Alışveriş (TL)	Kredi Kartı Üst Limiti (TL)
A	500	5000
B	750	7500

C	850	2250
D	900	3500

Kaynak: Silahtaroğlu, 2013:167

Bu amaçla öncelikle tüm müşterilerin birbirleriyle olan mesafelerini gösteren mesafe matrisi çıkartılacaktır. Euclid mesafeleri;

$d(A,B)=\sqrt{\sum(A_i - B_i)^2}=\sqrt{(500-750)^2+(5000-7500)^2}=2512$

$d(A,C)=\sqrt{\sum(A_i - C_i)^2}=\sqrt{(500-850)^2+(5000-2250)^2}=2772$

$d(A,D)=\sqrt{\sum(A_i - D_i)^2}=\sqrt{(500-900)^2+(5000-3500)^2}=1552$

$d(B,C)=\sqrt{\sum(B_i - C_i)^2}=\sqrt{(750-850)^2+(7500-2250)^2}=5221$

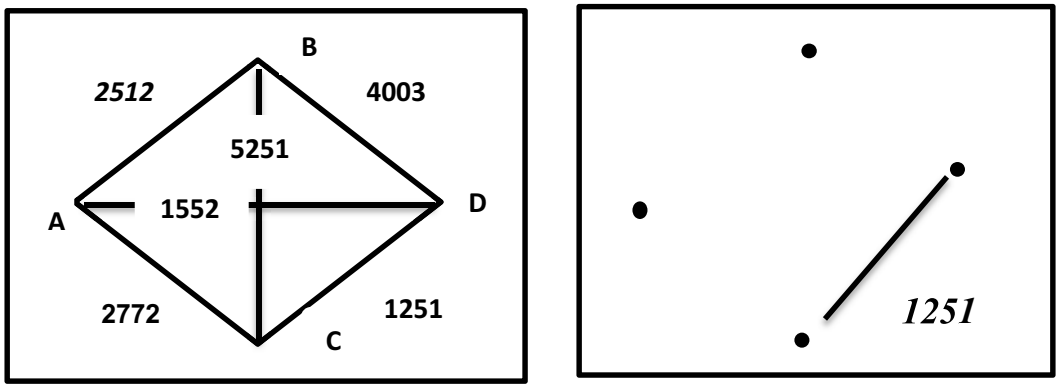
$d(B,D)=\sqrt{\sum(B_i - D_i)^2}=\sqrt{(750-900)^2+(7500-3500)^2}=4003$

$d(C,D)=\sqrt{\sum(C_i - D_i)^2}=\sqrt{(850-900)^2+(2250-3500)^2}=1251$

Tablo 25 : Mesafe tablosu

	A	*B*	*C*	*D*
A	*0*	*2512*	*2772*	*1552*
B	*2512*	*0*	*5251*	*4003*
C	*2772*	*5251*	*0*	*1251*
D	*1552*	*4003*	*1251*	*0*

Şekil 13: Örnek Mesafe Şekli



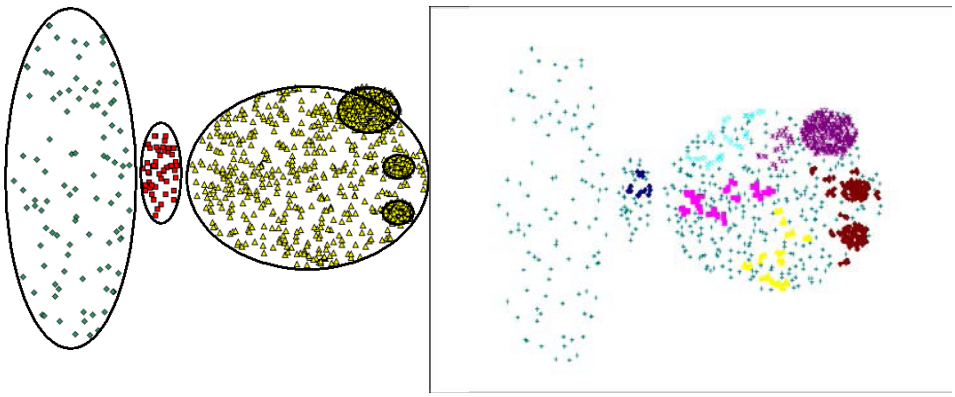
Eşik değeri 1500 olarak alınırsa; eşik değerinden büyük olan bağları koparılır ve toplam üç ayrı kümenin oluştuğu görülür. Bu kümeler {A},{B} ve {C,D} kümeleridir.

2.3.1.2 CURE algoritması

CURE (Clustering Using Representattives), yani temsilciler kullanarak kümeleme algoritmasında amaç aykırı veriyi kontrol altında tutmaktır. Hem hiyerarşik bir bileşene hem de bölümleme bileşenine sahiptir. CURE algoritması başlangıçta her girdiyi sanki ayrı bir kümeymiş gibi ele alır. İki kümenin birbirine benzerliği ya da farklılığı her kümenin kendi içinde olan benzerliği ya da farklılığı ile karşılaştırılır Her adımda bu küme temsilcilerin birbirlerine olan yakınlıklarına göre birleştirilir ya da ayrı küme olarak tutulur. İlk olarak, her kümeden sabit bir sayıdaki c adet temsilci seçilir. Bu temsilci noktalar daha sonra bir α daralma katsayısı ile kümenin merkezine doğru kaydırılır. Bu α katsayısı ile çarpılması aykırı verilerin etkisini azaltmak içindir. Temsilci noktalar, kaydırma işleminden sonra artık o kümenin temsilcileri olarak kabul edilirler. Bu aşamadan sonra iki küme arasındaki uzaklık, her biri bir kümeye ait olan en yakın temsilci çifti arasındaki uzaklıktır. α değeri 0-1 arasında yer alır. Küçük α değeri temsilci noktaların çok az yer değiştirmesine sağlar, büyük α değeri ise küme merkezine daha çok yaklaştıracağı için birleşik kümeler oluşturacaktır. $\alpha=1$ durumunda ise CURE algoritması merkezi temelli algoritmalara yaklaşacaktır. Kullanılan daraltma katsayısı aynı zamanda, oluşan kümelerin şeklini belirlemede de kullanılabilir.

CURE hiyerarşik kümeleme algoritmalarındaki zayıflıkları ve aykırılıklara karşı hassasiyetten kurtulmak için geliştirilmiştir. CURE algoritmasındaki her adımda, en yakın temsilci noktaya sahip kümeler birleştirilmek üzere seçilir. Aralarındaki mesafe, iki kümenin temsilci kümelerindeki herhangi bir nokta çifti arasındaki minimum mesafe olarak tanımlanır.

Şekil 14 : CURE Algoritmasının Şekil İle Gösterimi



Gerçek noktalar **CURE**

Büyük veri tabanları için CURE'ün temel adımları:

1. Örnek bir veri tabanı alınır.
2. Örneği n / p boyutunda p bölümlerine ayırlır. Bu, algoritmayı hızlandırmak için yapılır çünkü kümeleme ilk önce her bölüm üzerinde gerçekleştirilir.
3. Hiyerarşik algoritmayı kullanarak her bölümdeki noktalar kısmen kümelenir.
4. Aykırı değerler çıkarılır.
5. Her küme için x adet temsilci nokta seçilir, Veri tabanındaki bir öğe, kendisine en yakın temsili noktası olan kümeye yerleştirilir
6. İki küme arasındaki uzaklık, temsilci noktalar arasındaki Öklit uzaklığı kullanılarak bulunur
7. En yakın küme çifti birleştirilir. Birleşen yeni küme için yeni temsilci noktalarını belirlenir. Bu noktalar daraltma katsayısı ile çarpılarak merkeze doğru yaklaştırılır,
8. Küme sayısı, kümeleme algoritmasında giriş parametresi olarak verilen k değerine ulaşana kadar 5, 6 ve 7. adımlar tekrarlanır.

CURE algoritmasının başarılı sonuçlara ulaşabilmesi, parametrelerinin doğru seçilmesine bağlıdır. Büyük veri kümeleri ile elde edilen sonuçlar, CURE'ün iyi performans gösterdiğini göstermektedir.

2.3.2 Bölümlemeli Yöntemler

Bölümlemeli yöntemlerde, kullanıcı tarafından belirlenen *k* adet küme sayısına göre n adet gözlem *(k<n)* kümelere ayrılmaktadır. Hiyerarşik yöntemlerde olanlardan farklı olarak, belirtilen grup sayısı için maksimum iç uyumu elde etmemize izin veren gruplandırmaya ulaşma gibi belirlenmiş optimizasyon kriterlerini karşılayan tek bir bölüm verir.(Giudici, 2004:83).

Veriler k gruba ayrılırken her grup en az bir nesne içermeli ve her nesne tam olarak bir gruba ait olmalıdır. İyi bir bölümlemenin genel kriteri, aynı kümedeki

nesnelerin yakın veya birbiriyle ilişkili olmaları, farklı kümelerin nesneleri birbirinden uzak ya da çok farklı olmalarıdır. Genel olarak, Bölümlemeli algoritmalar aşağıdaki gibi özetlenebilir:

1. k seçilir ve ilk küme belirlenir.
2. Her gözlemi başlangıç grubundan başka bir gruba değişimi yapılarak değerlendirilir. Amaç, grupların iç uyumunu en üst düzeye çıkarmaktır. Değişim esnasında belirlenen objektif fonksiyondaki değişim hesaplanır ve eğer uygunsa tekrar değişim yapılmaz.
3. Durdurma kuralı yerine getirilinceye kadar 2. adımı tekrarlayın.

Bölümlemeli algoritmalar, büyük veri kümelrinde hızlı çalışır, uzaklık/mesafe ölçümüne ihtiyaç duymaz ve gözlemleri örtüşmeyen gruplara ayırmanın birçok olası yolu olabilir,

2.3.2.1 K-Ortalama (K-means) Algoritması

K-ortalamalar algoritması istenen kümeye ulaşılana kadar öğelerin kümeler arasında taşındığı yinelemeli kümeleme algoritmasıdır. Küme merkezleri oluşturulurken her bir iterasyonda oluşan kümeler için değişkenlerin ortalamaları alınır. Oluşturulacak k adet küme sayısı başlangıçta kullanıcı tarafından belirlenebilir. K için tipik bir değer 2 ila 10'dur. K-ortalama algoritması kategorik veriler üzerinde çalışmaz ve aykırı değerlerle ilgilenmez. Burada ortalama ile kasıt küme merkezleri olarak düşünülmelidir. K-ortalama algoritması sıklıkla iyi sonuçlar vermesine rağmen, zaman verimli değildir ve iyi ölçeklenemez.

Algoritmanın uygulama aşamaları aşağıda sırlanmaktadır;

1. Kullanıcı tarafından belirlenen k adet kümenin $m_1,m_2……..m_k$ ortalamaları belirlenir.

2. Tüm veriler en yakın olduğu seçilen m_i kümesine atanır.

3. kümelere ait $m_1,m_2……..m_k$ ortalama değerleri yenioluşan kümeye göre yeniden hepalanır.

4. küme üyelerinde değişiklik yoksa algoritma durur. Değişiklik varsa en başa döner.

D={3,7,25,28,35,12,15,17,32,4} bu 10 ayrı noktayı *k*=2 kümeye ayrılması ve başlangıçta m_1=3 ve m_2 =7 (ilk iki değer) seçildiği varsayımı üzerinden ilerlenirse *7,25,28,35,12,15,*17 ve 32 m_2 =7 değerine daha yakın olduğu için bu değer k_2 kümesine; 3 ve 4 m_1=3 değerine daha yakın olduğu için k_1 kümesine dahil olacaktır. Bu durumda m_1 ve m_2 değerleri tekrar hesaplanarak;

m_1=3,5 m_2 =21,3

tüm noktalar yeni m_1 ve m_2 değerlerine göre yeniden kümeleneceklerdir. Bu durumda;

k_1*={3,7, 12, 4}* k_2*={25,28,35,15,17,32}*

k_1 ve k_2 nin elemanları değiştiği için algoritma yürütülmeye devam edecektir. Yeni m_1 ve m_2 değerleri;

m_1=6,5 m_2 =25,3

olduğundan;

k_1*={3,7,4,12,15}* k_2*={25,28,35,17,32}*

yeni ortalamalar;

m_1=8,2 m_2 =27,4

yeni kümeler;

k_1*={3,7,4,12,15,17}* k_2*={25,28,35,32}*

k_1 ve k_2 nin elemanları değiştiği için algoritma yürütülmeye devam edecektir.

m_1=9,6 m_2 =30

bu ortalamalara göre kümeler değişmediğinden algoritma sonlanacak kümeler ise;

k_1*={3,7,4,12,15,17}* k_2*={25,28,35,32}*

şeklinde bulunmuş olacaktır.

Görüldüğü gibi *k*-Ortalama algoritmasının sonlanma kriteri iki defa üst üste aynı kümelerin bulunmasıdır.

2.3.2.2.K-medoids Algoritması (PAM Algoritması)

K-medoidler algoritması olarak da adlandırılan PAM (partitioning around medoids) medoidler etrafında bölümleme algoritması, bir medoidler kümesini gösterir.

Medoidler kümelerin temsilci, noktalarıdır. PAM algoritmasında her seferinde temsilci nokta olan medoid değiştirilerek kümeleme işlemi tamamlanır. Medoid küme merkezine en yakın nokta olarak belirlenebilir. K adet rastgele kümeye ayrılan verilerin her küme için k adet temsilci nokta belirlenebilir. Diğer noktalar bu temsilci noktalara benzerliklerine göre en çok benzedikleri temsilci etrafında toplanmaktadır. Kümenin başarısı temsilci ile diğer noktalar arası mesafeye göre değerlendirilebilir.

Algoritman uygulanırken

1.K adet kümenin k adet temsilcisi belirlenir.

2. seçilmiş bir temsilci ile seçilmemiş bir verinin yer değiştirmesinin kümenin kalitesi üzerinde yaratacağı iyileştirme hesaplanır.

3. bu hesaplanan iyileştirmeye göre temsilcinin değiştirilip değiştirilmiyceğine algoritma karar verir.

4.hiçbir değişiklik olmayana kadar işlem tekrar edilir.

2.3.3 Yoğunluğa Dayalı Algoritmalar

Yoğunluğa dayalı algoritmalar, diğer bir ifade ile çoğu bölümleme yöntemi, veriler arasındaki mesafeye bağlı olarak verileri kümeler. Bu tür yöntemler yalnızca küresel biçimli kümeleri bulabilir ve isteğe bağlı şekil kümelerini keşfetmede zorlukla karşılaşabilir. Genel bir ifade ile komşu yoğunluğun (nesne veya veri noktalarının sayısı) bir miktar eşiği aştığı sürece verilen kümeyi büyütmeye devam etmektir; yani, belirli bir küme içindeki her veri noktası için, belirli bir yarıçapın sınırının en az minimum nokta içermesi gerekir. Kümeler belirlenmesi sırasındaki hesaplamaları etkileyebileceği için aykırı değerler çıkarılmalıdır.

Bir aradayken bir yoğunluk meydana getiren noktalar farklı kümeler olarak ayrılır. Bu yoğunluğun belirlenmesi aşamasında bazı noktalar kümelerin sınırları olarak alınırken bazı noktalarsa kümelerin içini oluşturabilir. Genel olarak yapılan işlem birlikteyken yoğunluk oluşturan verileri küme olarak değerlendirilebilmesidir.

Yoğunluğa dayalı algoritmalara örnek olarak DBSCAN, OPTICS ve DENCLUE algoritmaları verilebilir.

2.3.4 Grid Temelli (Izgara tabanlı) Algoritmalar

Izgara tabanlı yöntemler, nesne alanını bir ızgara yapısı oluşturan sınırlı sayıda hücreye dönüştürür. Kümeleme işlemlerinin tümü ızgara yapısı üzerinde gerçekleştirilir. Bu yaklaşımın temel avantajı, tipik olarak veri nesnelerinin sayısından bağımsız olan ve yalnızca nicelenmiş alandaki her boyuttaki hücrelerin sayısına bağlı olan hızlı işlem süresidir.

2.3.4.1 STING Algoritması

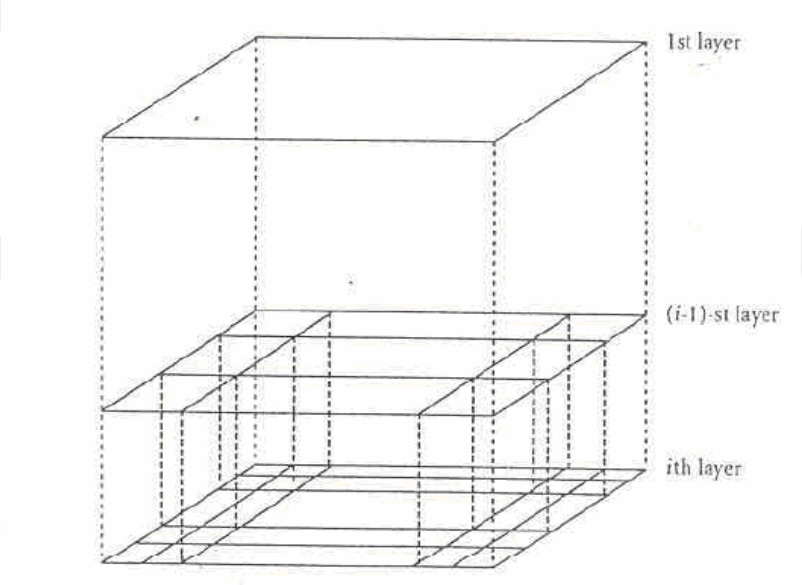
STING (Statistical Information Grid- İstatiksel bilgi Grid) algoritmasında üzerinde çalışılan alan dikdörtgen hücrelere bölünmekte ve aşama sırasına göre bir yapı oluşturulmaktadır. Üst seviyede bulunan bütün bölümler bir sonraki aşamada parçalanmış bir durumda bulunmaktadır. Her bir grid bölümündeki özellikler ile ilgili istatistiksel bilgilerin kaydı tutulmaktadır.

STING, uzamsal alanın dikdörtgen hücrelere bölündüğü, ızgara tabanlı, çoklu çözünürlüklü bir kümeleme tekniğidir. Genellikle farklı çözünürlük seviyelerine karşılık gelen bu tür dikdörtgen hücrelerin birkaç seviyesi vardır ve bu hücreler hiyerarşik bir yapı oluşturur: yüksek seviyedeki her bir hücre, bir sonraki daha düşük seviyede birkaç hücre oluşturmak için bölümlenir. Her bir grid hücresindeki özelliklerle ilgili istatistiksel bilgiler (ortalama, maksimum ve minimum değerler gibi) önceden hesaplanır ve saklanır. Daha yüksek seviyeli hücrelerin istatistiksel parametreleri, daha düşük seviyeli hücrelerin parametrelerinden kolayca hesaplanabilir. Bu parametreler: bağımsız parametreler, sayı; bağımlı parametreler, ortalama, stdev (standart sapma), min (minimum), maks (maksimum); ve hücredeki öznitelik değerinin normal, tek biçimli, üstel ya da hiç (örneğin dağıtım bilinmiyorsa) dağıtım türü şeklindedir. Veriler veritabanına yüklendiğinde, parametreler sayılır, ortalama, stdev, min. ve alt seviye hücrelerin maksimum değerleri doğrudan veriden hesaplanır. Dağılım tipi önceden bilinirse veya χ^2 testi gibi hipotez testleriyle elde edilirse, dağıtım değeri kullanıcı tarafından atanabilir. Daha yüksek seviyeli bir hücrenin dağılım tipi, eşik filtreleme işlemi ile birlikte karşılık gelen düşük seviyeli hücrelerin dağıtım tiplerinin çoğunluğuna dayanarak hesaplanabilir. Alt seviyeli

hücrelerin dağılımları birbirleriyle aynı fikirde değiller ve eşik testinden geçemezlerse, yüksek seviyeli hücrenin dağıtım tipi yok olarak ayarlanır(Han Ve Kamber 2001:425).

İstatistiksel parametreler yukarıdan aşağıya, ızgara bazlı bir yöntemle kullanılabilir. İlk önce, hiyerarşik yapıdaki bir katmanın, sorgu cevaplama işleminin başlayacağı yer belirlenir. Bu katman tipik olarak az sayıda hücre içerir. Geçerli katmandaki her hücre için, hücrenin verilen sorguya ilgi düzeyini yansıtan güven aralığını (veya tahmini olasılık aralığını) hesaplarız. İlgisiz hücreler dikkate alınmaz. Bir sonraki alt seviyenin işlenmesi kalan ilgili hücreleri inceler. Bu işlem, alt tabakaya ulaşılana kadar tekrar edilir. Sting kümelemenin hiyerarşik yapısı Şekil 15'de yer almaktadır.

Şekil 15: STING Kümeleme Hiyerarşik Yapısı.



Kaynak : Han ve Kamber 2001:426

STING kümelemesinin kalitesi, ızgara yapısının en düşük seviyesinin ayrıntı derecesine bağlıdır. Taneliğin çok iyi olması durumunda, işlem maliyeti önemli ölçüde artacaktır; Bununla birlikte, ızgara yapısının alt seviyesi çok kaba ise küme analizinin kalitesini düşürebilir. Küme sınırlarının tümü yatay veya dikeydir ve hiçbir çapraz sınır algılanmaz. Bu, tekniğin hızlı işlem süresine rağmen kümelerin kalitesini ve doğruluğunu düşürebilir.

2.3.4.2 Dalga Kümeleme

Dalga kümeleme (Wave cluster) algoritması grid temelli bir algoritma olup büyük veri tabanlarının kümelenmesi için kullanılan oldukça hassas kümeleme yapan bir algoritmadır. Dalga kümeleme yapılırken aşağıdaki adımlar kullanılmaktadır (Silahtaroğlu, 20013:203).

1. Örnek değişkenler sayısallaştırılır
2. Dalga dönüşüm işlemi örnek uzaya uygulanır.
3.Dönüştürülmüş örnek uzayın alt katmanındaki birbirine bağlı kümeler bulunur
4. Birimler isimlendirilir.
5. Adımda değişkenlerin kümelere ataması yapılır.

Kümeleri belirlemek için filtreler kullanılmakta ve eş zamanlı olarak zayıf bilgileri kendi sınırları içinde bastırmaktadır. Ana veri grubu içerisindeki kümeler ortaya belirgin bir şekilde çıkmakta ve kümenin etrafı temizlenmektedir.

2.3.5 Genetik Algoritmalar

Genetik algoritmalar (GA'lar), doğal seçilimin işleyiş süreçlerini hesaplamalı olarak taklit etmeye çalışır ve bunları işletme ve araştırma problemlerini çözmek için uygular. Genetik algoritmalar dünyasında, çeşitli potansiyel çözümlerin uygunluğu karşılaştırılır ve en uygun potansiyel çözümler daha da optimum çözümler üretmek için geliştirilir.

Genetik algoritmalar alanı genomik terminolojiden esinlenerek geliştirilmiştir. Vücudumuzdaki her hücre, bizi oluşturan bir plan görevi gören DNA dizileriyle aynı kromozom kümesini içerir. Daha sonra her bir kromozom, göz rengi gibi belirli bir özelliği kodlamak için tasarlanmış DNA blokları olan genlere ayrılabilir. Her gen, kromozom üzerindeki belirli bir lokasyonda bulunur. Genetik algoritmalar alanında, bir kromozom, sorun için aday çözümlerden birini ifade eder, bir gen de, aday çözümün tek bir biti veya basamağıdır. Genetik algoritmalar tarafından seçim, çaprazlama ve mutasyon kullanılır (Larose, 2006:241).

Kümelemenin genetik algoritmalarla nasıl gerçekleştirileceğini belirlemek için önce her bir kümenin nasıl temsil edileceğini belirlemeliyiz. Basit bir yaklaşım, olası her küme için bir özellik haritası gösterimini kullanmak olacaktır. Genetik algoritmaları aracılığı ile merkezleri hesaplanan kümeler k adet kümeye ayırılır. Burada amaç kümelerin merkezini minimum yapmaktır. Kümelerin merkezi ise kümeyi oluşturan verilerin birbirleriyle olan Öklid mesafesi toplamına eşittir. Genetik kümeleme algoritmaları bölgesel bir potansiyel çözüm arayışı yerine küresel bir arama gerçekleştirir. Genetik algoritmalar için orijinal çerçeve, ikili kodlanmış veriler için geliştirilmiştir, çünkü çaprazlama ve mutasyon işlemleri bu verilerle doğal ve iyi çalışabilir.

2.4 Birliktelik Kuralları Ve İlişki Analizi

İlişki analizi veri tabanındaki bir dizi bilginin birlikte gerçekleşmesi durumunu açıklayan işlemlerdir. Bir bilgi varken, bu bilgi ile ilişkili başka bir bilginin var olma olasılığı nedir?

Birliktelik kuralları belirli olasılıklar ile ortaya konulur. Özellikle perakende sektöründe herhangi bir ürün satılırken, satılan bu ürünün yanında başka bir ürünün de satılması bu ürünler arası ilişkiyi bu analiz ile irdelenebilir. Satılan ürüler arasındaki satın alınma eğilimini ve müşteri alışkanlıklarını ortaya koyan uygulamalara "*pazar sepeti analizi"* denir.

Alış-veriş merkezlerinde ürün yerleştirmelerinde, mağaza raf tasarımında, mağaza ürün eğilimin belirlenmesi ve katalog tasarımı gibi birçok konuda pazar sepeti analizinden faydalanılabilir.

Birliktelik kurallarının çıkartılabilmesi için çeşitli algoritmalar geliştirilmiştir. AIS algoritması (Agrawal,1993) 1993'te ortaya atılmış, daha sonra SETM (Houtsma ve Swami,1993) algoritması ve 1994 ortaya atılan ve bugünde çok kullanılan APRIORI algoritması geliştirilmiştir.

2.4.1 Destek Ve Güven Ölçütleri

Pazar sepeti analizi ürünler arası ilişkileri gösterebilmek için destek ve güven ölçütünden yararlanılmaktadır.

Destek sayısı A ve B gibi ürünün birlikte alındığı alışveriş sayısını ifade etmektedir ve $sayı(A,B)$ şeklinde gösterilir. N tüm alışverişlerin sayısını göstermektedir.

Destek ölçütünde kural, bir ilişkinin tüm alışverişler içerisindeki tekrarlama oranını ifade eder.

$$destek\ (A \rightarrow B) = \frac{sayı(A,B)}{N}$$

Güven ölçütünde kural, A ürününü alan kişilerin B ürününü de alma olasılığını göstermektedir.

$$güven\ (A \rightarrow B) = \frac{sayı(A,B)}{sayı(A)}$$

Eşik değer, destek ve güven ölçütlerini karşılaştırmak maksadıyla kullanılmaktadır. Hesaplanan destek ve güven ölçütlerinin destek (eşik) ve güven (eşik) değerlerinden büyük olması beklenmektedir. Hesaplanan destek ve güven ölçütleri ne kadar büyükse birliktelik kuralları o kadar güçlüdür kanısına varılır.

2.4.2 Apriori Algoritması

Apriori algoritması, en yaygın olarak kullanılan algoritmalardan biridir. Apriori algoritması, "Büyük bir veri tabanının herhangi bir alt kümesi de büyük olmalıdır." fikrinden yola çıkmıştır.

Büyük nesne kümelerini bulmak genellikle oldukça kolay olabilir. Fakat çok maliyetlidir. En basit yaklaşım, herhangi bir işlemde görünen tüm veri setlerini saymak olacaktır. Büyük boyutlu veri setlerini keşfetmeye yönelik algoritmalar veriler üzerinde çoklu taramalar yapar. İlk taramada, her bir nesnenin destek seviyesi hesaplanır ve kullanıcı tarafından belirlenen minimum destek seviyesi ile karşılaştırılır. Bu duruma göre her bir nesnenin geniş olup olmadığı belirlenir. Takip eden her taramada, önceki taramada büyük bulunan bir çekirdek nesne kümesi ile başlanır. Bu çekirdek kümesini, aday ürün setleri adı verilen ve potansiyel olarak büyük yeni ürün setleri üretmek için kullanılır ve bu aday ürün setleri için gerçek veri

desteğini hesaplanır. Taramanın sonunda, aday öğelerden hangisinin gerçekte büyük olduğunu belirlenir ve bir sonraki geçiş için çekirdek haline gelir. Bu işlem, yeni büyük nesne kümesi bulmayana kadar devam eder (Agrawal ve Srikant, 1994:488)

Aşağıda Tablo 26'da verilen verilere minimum destek %30 (ve güven %60) olacak şekilde apriori algoritması uygulanmak istenirse;

Tablo 26: Apriori Algoritması Örnek Veri Tabloları

ID	Sepet
1	Ekmek, Peynir, Makarna, Zeytin
2	Ekmek, Peynir, Zeytin
3	Yumurta, Zeytin
4	Yumurta, Çay
5	Ekmek, Peynir
6	Ekmek, Peynir, Yumurta

Birinci Tarama

Ürün	Miktar	Destek
Ekmek	4	%67
Peynir	4	%67
Makarna	1	%17
Zeytin	3	%50
Yumurta	3	%50
Çay	1	%17

İkinci Tarama

Ürün	Miktar	Destek
Ekmek, Peynir	4	%67
Ekmek, zeytin	2	%33
Ekmek, Yumurta	1	%17
Peynir, Zeytin	2	%33
Peynir, Yumurta	1	%17
Zeytin, Yumurta	1	%17

Üçüncü tarama

Ürün	Miktar	Destek
Ekmek, Peynir, Zeytin	2	%33

En geniş setler ve yüzdelikleri:
Ekmek alanlar (4 kayıt), Peynir ve Zeytin alır (2 kayıt) [Destek %33, Güven %50]
Peynir alanlar (4 kayıt), Ekmek ve Zeytin alır (2 kayıt) [Destek %33, Güven %50]
Zeytin alanlar (3 kayıt), Ekmek ve peynir alır (2 kayıt) [Destek %33, Güven %67]
Ekmek ve Peynir alanlar (4 kayıt), Zeytin alır (2 kayıt) [Destek %33, Güven %50]
Ekmek ve Zeytin alanlar (2 kayıt), Peynir alır (2 kayıt) [Destek %33, Güven %100]

Görüldüğü gibi 3 ve 5 numaralı (en geniş) nesne kümeleri başlangıçta verilen %30 destek ve %60 güven seviyeleri ölçütünü karşılamaktadır. Elimizdeki veri

tabanına göre Ekmek-Peynir-Zeytin en geniş nesne kümesidir. Güven seviyesi bakış açısına göre değişmektedir.

ÜÇÜNCÜ BÖLÜM
UYGULAMA

Bu uygulamada, Türkiye'deki 21 kitap sitesinden alınan en çok satan kitap listelerindeki veriler, veri madenciliği teknikleri uygulanmak üzere çalışma kapsamında ele alınmıştır. Veriler analiz edilirken IBM SPSS Modeller 18.1 programı kullanılarak analiz edilmiştir. Veri madenciliği süreciyle ilgili en yaygın kullanılan CRISP-DM modelinin aşamaları olan;

1. Aşama: Araştırmanın Tanımlanması (Business understanding)
2. Aşama: Veriyi tanımlama (Data understanding)
3. Aşama: Veri hazırlama (Data preparation)
4. Aşama: Modelleme (Modelling)
5. Aşama: Değerlendirme (Evaluation)
6. Aşama: Uygulama ve Sonuçları Yayma (Deployment)

takip edilmiştir.

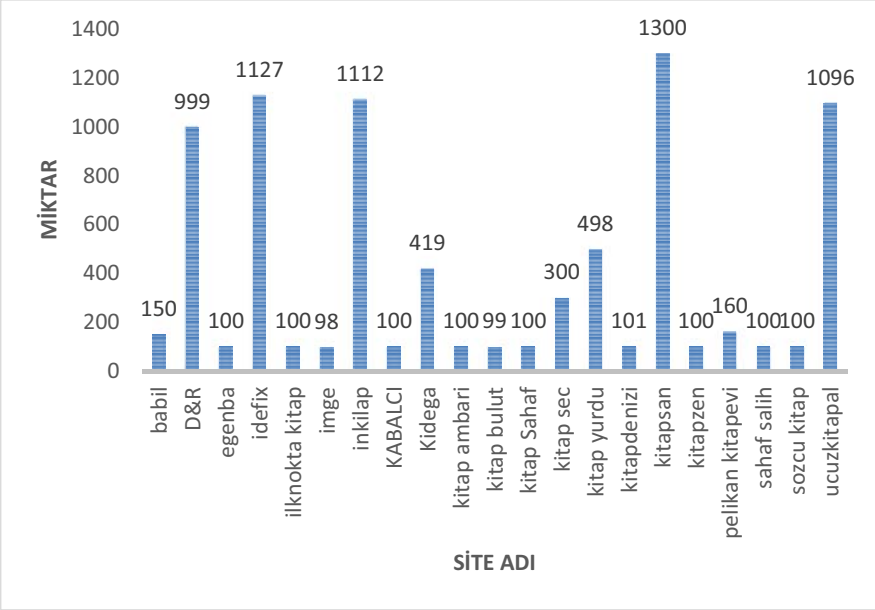
3.1 ARAŞTIRMANIN TANIMLANMASI

Son yıllarda hızla gelişen teknolojiyle birlikte bilgiye ulaşmak her ne kadar kolay olsa da güvenilir ve doğru bilgiye ulaşmak gittikçe zorlaşmaktadır. Her geçen gün farklılaşan ve artan bilgi kaynaklarına rağmen kitapların halen en etkili ve doğru bilgiye ulaşmak için en iyi yol olduğunu söyleyebiliriz (Acıyan, 2008). Kitapların en doğru bilgi kaynağı olması fikrinden yola çıkarak Türkiye'de kitap satışı yapan internet sitelerinden alınan en çok satan kitap listelerindeki kitapları türlerine göre sınıflandırma yaparken hangi faktörlerin etkili olduğu belirlenmeye çalışılmıştır. Kitap türleri 2 farklı başlığa indirgenmiştir. Bunlar eğitim/sınav/kişisel gelişim/tarih/araştırma/bilim kitapları ve roman/edebiyat/çocuk kitapları şeklindedir.

3.2 VERİYİ TANIMLAMA

Uygulamada faydalanılan veriler, Türkiye'deki 21 internet sitesinin 2019 yılı Ocak ayına ait en çok satan kitap listelerinden alınan toplam 8258 veriden oluşmaktadır. Yararlanılan site isimleri ve sitelerden alınan en çok satan kitap liste sayıları aşağıda Şekil 16'de verilmiştir.

Şekil 16: Kitap Liste Sayıları



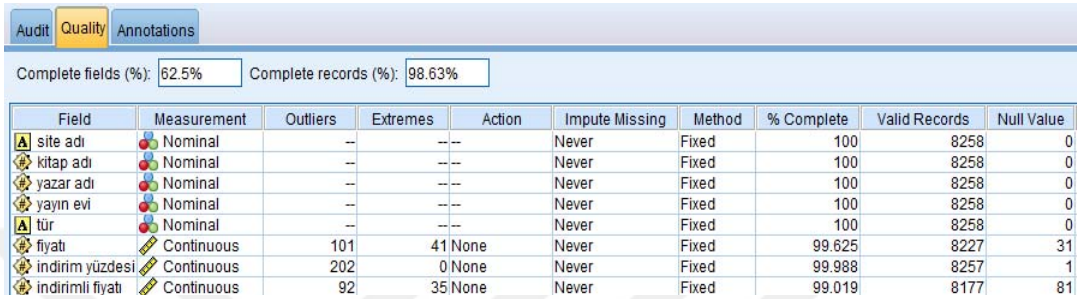
3.3 VERIYI HAZIRLAMA

Veri madenciliğinin en uzun aşamalarından biri elde edilen verilerin birleştirilmesi, düzene koyulması, temizlenmesi, eksik ve aşırı uçta gözlemlerin belirlenip gerekli işlemler yaparak veri kümesinin analiz edilebilir duruma getirme sürecini kapsayan veri hazırlama aşamasıdır. Bu sürecin uzun zaman alması, kullanılan veri kümelerinin farklı fiziksel ortamlardan alınması, çok büyük boyutlarda olması ve işlenmemiş ham halde bulunan verilerde görünen anlamsızlık gibi nedenlere dayanmaktadır.

Verilere farklı internet sitelerinden elde edildiği için farklı sistemlerde ve farklı tip formatlarda tutulmuştur. Bu sebeple, verileri analiz edilebilir hale getirmek için, farklı sistemlerde tutulan veri tabloları birleştirilerek tek bir Excel tablosuna aktarılmıştır.

Bu aşamada karşılaşılan, gürültülü, tutarsız, hatalı, eksik veya yinelenen değerler, analiz için işe yaramayan gözlemler gibi problemler giderilmiştir. Tekrar eden verinin bulunduğu kayıtlar veri kümesinden çıkarılmıştır. Sürekli değişkenler olan fiyat, indirim yüzdesi ve indirimli fiyat verilerinin eksik verilere (null value) sahip olduğu gözlemlenmiştir

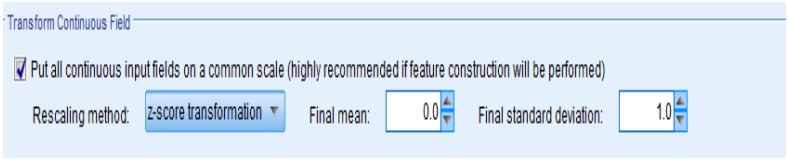
Şekil 17: Veri Düzenleme

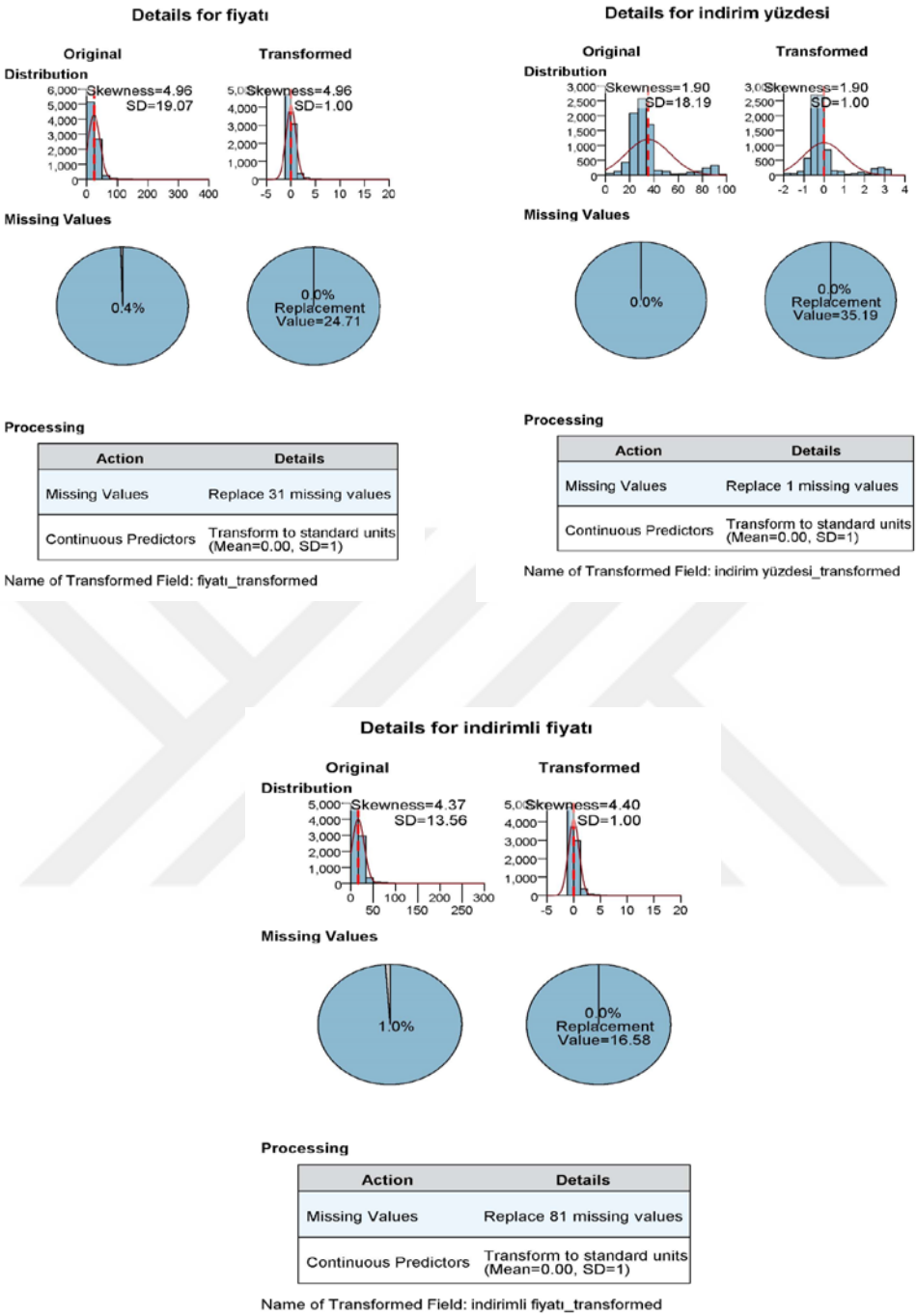


Field	Measurement	Outliers	Extremes	Action	Impute Missing	Method	% Complete	Valid Records	Null Value
site adı	Nominal	--	--	--	Never	Fixed	100	8258	0
kitap adı	Nominal	--	--	--	Never	Fixed	100	8258	0
yazar adı	Nominal	--	--	--	Never	Fixed	100	8258	0
yayın evi	Nominal	--	--	--	Never	Fixed	100	8258	0
tür	Nominal	--	--	--	Never	Fixed	100	8258	0
fiyatı	Continuous	101	41	None	Never	Fixed	99.625	8227	31
indirim yüzdesi	Continuous	202	0	None	Never	Fixed	99.988	8257	1
indirimli fiyatı	Continuous	92	35	None	Never	Fixed	99.019	8177	81

Bu eksik veriler SPSS Modeller 18.1 programı aracılığı ile veri hazırlama (data preparation) seçeneği kullanılarak verilerin ortalama değerleriyle doldurulup gerekli dönüşümler sağlanmıştır. Ek olarak sürekli değişkenlerimizde z skor standardizasyon işlemi uygulanmış ve ortalama 0 ve standart sapma 1 e dönüştürülmüştür.

Şekil 18: z Skor Standardizasyon





Kitap adı, yazar adı ve yayınevi gibi kategorik veriler için kategorilere ait tanımlamalar getirilmiştir. En çok satan kitap verilerinde kitap isimleri 1'den başlayarak 4813'e kadar etiketlenmiştir. Aynı şekilde yazar adı ve yayınevi verileri de analizi edilebilir hale getirmek için numaralarla etiketlendirilmiştir. Veriler kategorilerine ayrıldığında 4813 farklı kitap, 2585 farklı yazar ve 616 farklı yayınevinden oluştuğu görülmüştür. Kitap, yazar ve yayın evi verilerinde çeşitliliğin

fazla olması ve analizi kolaylaştırmak adına SPSS Modeller 18.1 programı kullanılarak gruplama işlemi yapılmıştır. Verilerin gruplanması her ne kadar ham verilerdeki detayları kaçırmamıza sebep olsa da bu yöntemle elde edilen kapsayıcı biçim ve bu biçimin sağladığı ilişkiler önemli fayda sağlamaktadır. Ayrıca bazen bir değişken tarafından temsil edilen bilgilere daha genel bakış açısı ile bakmak ve farklı değerlerin sayısını azaltmak gerekebilir (Wendler ve Gröttrup, 2016:237). Gruplandırma genişliği 16 seçilmiştir. 16 seçilmesinin nedeni daha küçük bir genişlik seçildiğinde veri sayısı fazla olduğundan gruplandırma yapmanın bir faydası olmamakta daha büyük bir genişlikte ise modellerin doğruluk oranları düşmektedir. Gruplandırma sonucunda kitap isimleri 302, yazar isimleri 163, yayın evi 40 grup olarak gruplanmıştır.

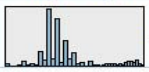
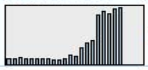
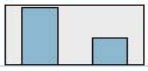
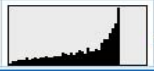
Çocuk, edebiyat, eğitim, sınav, roman, kişisel gelişim, tarih, araştırma, bilim vb. isimler altında sınıflandırılan kitap türleri SPSS Modeller 18.1 programı kullanılarak yeniden sınıflandırılma yapılmıştır. 2 türe düşürülen kitaplardan edebiyat, şiir, roman, çocuk, öykü, fantastik, çizgi roman sanat, biyografi vb. türündekiler 1 etiketiyle, eğitim, sınav, din, psikoloji, siyaset, tarih, bilim, kişisel gelişim vb. türündekiler 2 etiketiyle adlandırılmıştır. 2 türe indirilen kitap türlerinden 1. tür 5635 veriden 2. tür 2623 veriden oluşmaktadır.

Şekil 19 : SPSS Modeller 18.1 Arayüzü

Analiz edilebilir hale getirilen veriden, en çok satan kitap isimleri, yazar adları, yayınevi, kitap türü, indirim yüzdesi ve indirimli fiyatı gibi nicel bilgiler elde edilmiştir.

Şekil 20: Verilerin Genel Durumu

Audit | Quality | Annotations

Field	Sample Graph	Measurement	Min	Max	Mean	Std. Dev	Skewne...	Unique	Valid
indirim yüzdesi_transformed		Continuous	-1.935	3.399	-0.000	1.000	1.898	--	8258
indirimli fiyatı_transformed		Continuous	-1.229	19.621	0.000	1.000	4.396	--	8258
site adı _transformed		Nominal	--	--	--	--	--	21	8258
tür_transformed		Nominal	--	--	--	--	--	2	8258
kitap adı_BIN_transformed		Nominal	0	301	--	--	--	302	8258
yazar adı_BIN_transformed		Nominal	0	162	--	--	--	163	8258
yayın evi_BIN_transformed		Nominal	0	39	--	--	--	40	8258

3.4 MODELLEME VE SONUÇ

Çeşitli işlemlerden geçerek analize hazır hale gelen verileri sınıflandırma yapabilmek için en uygun algoritma ve model belirlenmeye çalışılmıştır. Bunun için öncelikle veriler eğitim (training), test (testing) olmak üzere 2 gruba ayrılmıştır. Genel olarak model eğitim verileri olarak seçilen grup ile eğitilir, daha sonra test verilerinde modelin başarısı test edilir. Bu 2 gruba ayırma işlemi SPSS Modeller 18.1 paket programındaki bölme (partition) düğümü ile yapılmıştır. Bu çalışmada 8258 kitap verileri %80 eğitim ve %20 test verisi olmak üzere ikiye ayrılmıştır.

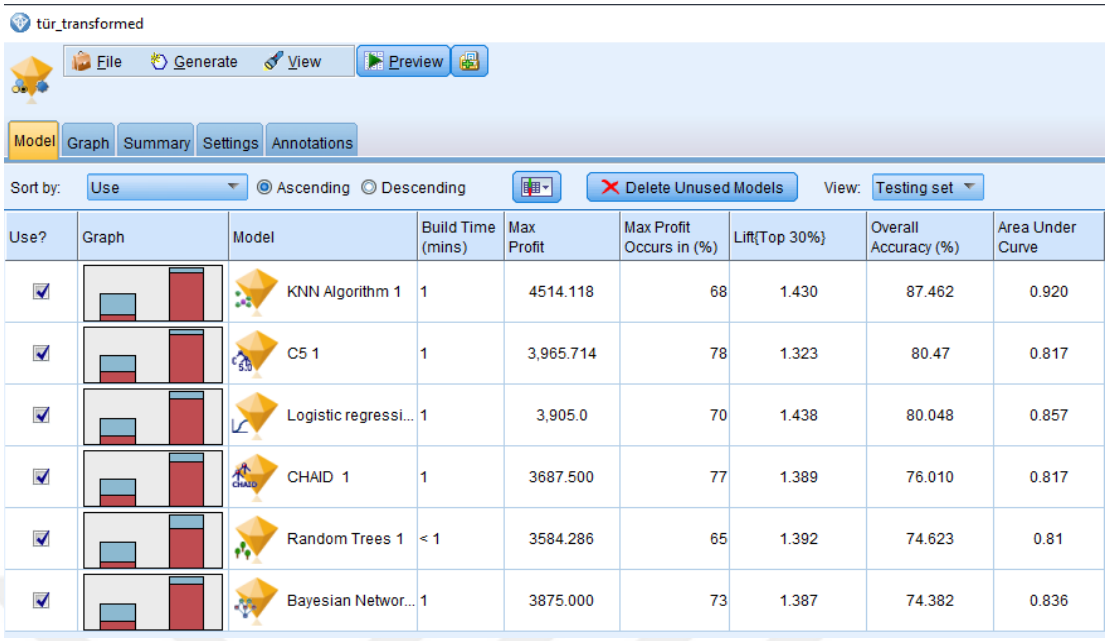
Şekil 21: Test Ve Eğitim Verileri Miktarları

Table | Graph | Annotations

Value	Proportion	%	Count
Training		79.91	6599
Testing		20.09	1659

SPSS Modeller 18.1 paket programı ile verilerin sınıflandırma algoritmaları ile çalıştırılması sonucunda oluşturulan modellerin performans oranları Şekil 22'de gösterilmiştir.

Şekil 22: Model Performans Oranları



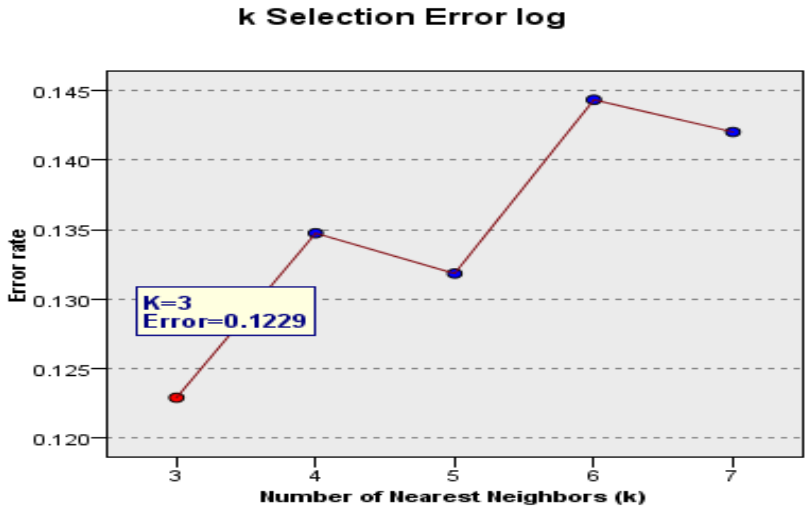
Use?	Graph	Model	Build Time (mins)	Max Profit	Max Profit Occurs in (%)	Lift{Top 30%}	Overall Accuracy (%)	Area Under Curve
☑		KNN Algorithm 1	1	4514.118	68	1.430	87.462	0.920
☑		C5 1	1	3,965.714	78	1.323	80.47	0.817
☑		Logistic regressi...	1	3,905.0	70	1.438	80.048	0.857
☑		CHAID 1	1	3687.500	77	1.389	76.010	0.817
☑		Random Trees 1	< 1	3584.286	65	1.392	74.623	0.81
☑		Bayesian Networ...	1	3875.000	73	1.387	74.382	0.836

Bu çalışmada %87.462 doğruluk oranı ve 0.92 AUC değeri ile K-Yakın Komşuluk (k-nearest neighbor, k-nn) algoritması en iyi algoritmalardan biri olmuştur. Bu sebeple sınıflandırma algoritmaları içinde en doğru tahminlemeyi yapan k-nn algoritması kullanılarak model kurulmaya çalışılmıştır.

3.4.1 k-nn Algoritmasının Uygulanması

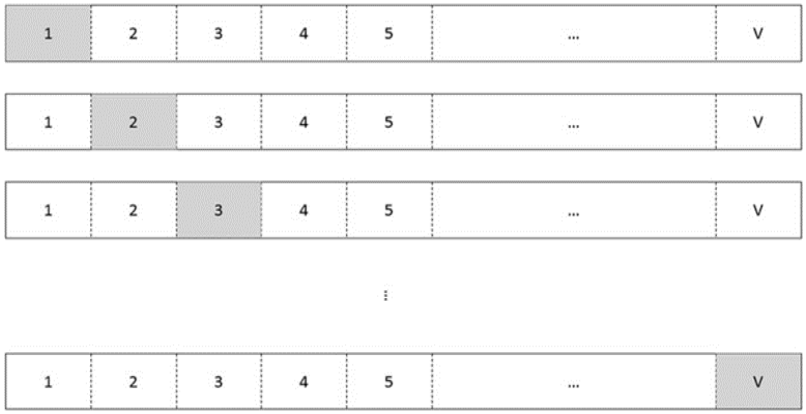
En yakın komşu (k-nn) algoritması parametrik olmayan bir sınıflamadır ve sınıflandırma yöntemleri arasında en basitlerinden biridir. Birbirine benzer veri noktalarının aynı sınıfta olduğu varsayımına dayanmaktadır. k-nn algoritmasında veriler koordinat sistemindeymiş gibi düşünülerek aralarındaki uzaklık hesaplanmakta ve bu uzaklık değeri için çeşitli uzaklık fonksiyonları kullanılabilmektedir. Uygulamamızda Öklid uzaklık fonksiyonu (Euclidean metric) kullanılarak uzaklık değerleri hesaplanmıştır. Başlangıçta kullanıcı tarafından belirlenen k sayısı kadar veri seçilip k tane verinin içinden en fazla hangi sınıfa ait veri tespit edilirse test edilen veri o sınıfa atanacaktır. Uygulamamızda k-nn algoritmasının gerekliliği olan k değeri SPSS Modeller 18.1 programı ile 3 ile 7 arasından otomatik olarak atmış ve en az hatayı veren k değeri seçilmiştir. En az hatayı veren değer 0,1229 hata oranı ile 3 değeri olmuştur. Şekil 23'de k değerleri hata grafiği verilmiştir.

Şekil 23: k Değeri Hata Grafiği



k-nn algoritmasında otomatik k seçimi seçildiğinden en iyi k'yı bulmak için çapraz doğrulama (cross-validation) işlemi yapılır. Çapraz doğrulama işleminde kullanılan yöntem, veri kümesini rastgele olarak eşit boyuttaki V parçaya ayırarak V-kat çapraz doğrulama yapmaktır. Bu V değeri genellikle 10 olarak seçilmektedir. 10 eşit parçaya ayrılan verilerin 9 parçası eğitim, 1 parçası test seti olarak alınır. Her seferinde farklı kombinasyon şeklinde 1 parça test olarak, diğer 9 parça da eğitim seti olarak alınır ve eğitim seti üzerinde eğitilen veriler test seti üzerinde test edilir. Test veri setindeki verileri eğitim veri setindeki veriler ile karşılaştırarak hangi sınıfa ait olduğunu bulur. En düşük hata oranına sahip model daha sonra nihai model olarak seçilir. Farklı kombinasyonların nasıl seçildiği Şekil 24'de gösterilmiştir.

Şekil 24: Çapraz Doğrulama



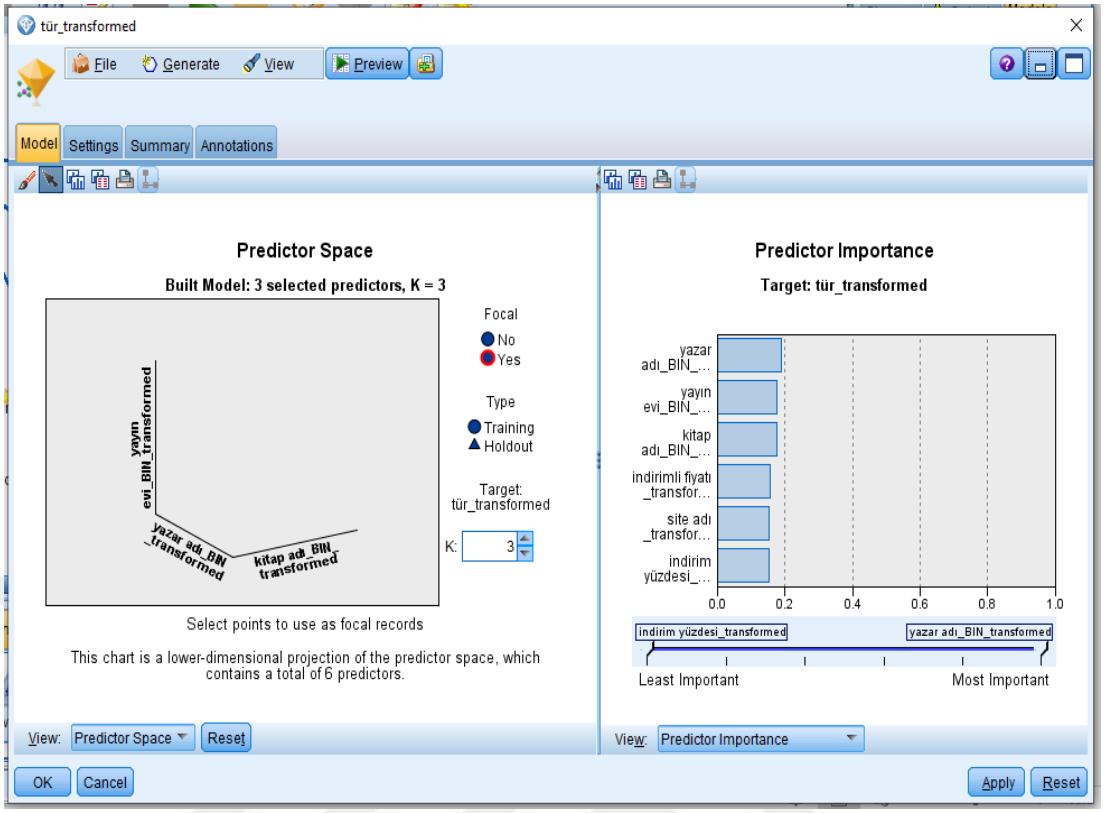
Tablo 27: Sınıflandırma Tablosu

Classification Table

Partition	Observed	Predicted		
		1	2	Percent Correct
Training	1	5210	425	92.5%
	2	617	2006	76.5%
	Overall Percent	70.6%	29.4%	87.4%

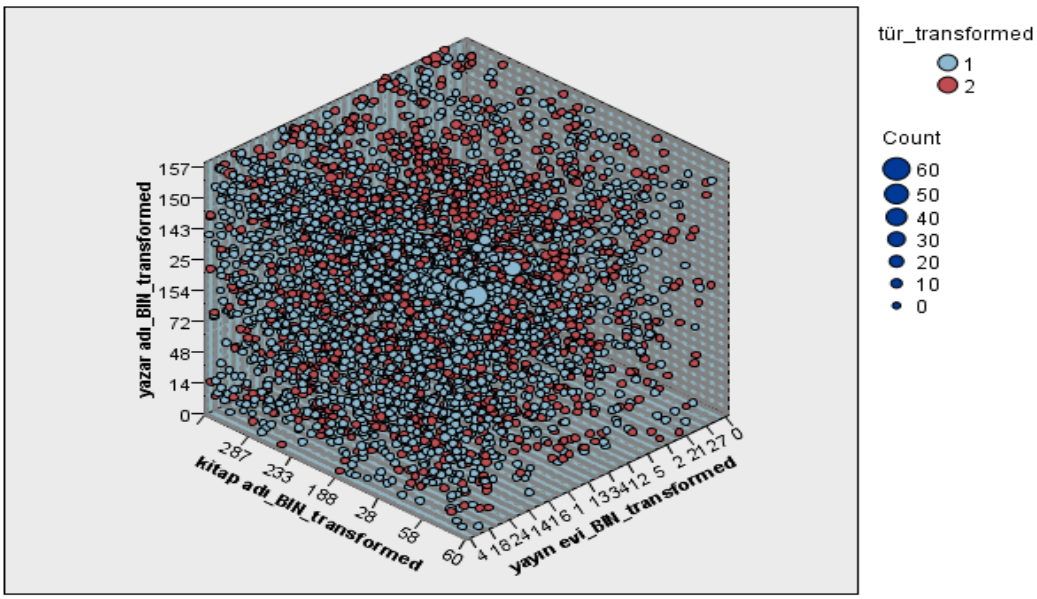
Kitap verilerine ilişkin en yakın 3 komşu ile oluşturulan modelde, tüm değişkenler neredeyse en önemli üç değişken olan "yazar adı", "yayın evi" ve "kitap adı" ile aynı derecede ilgilidir. Şekil 25'de k-nn algoritmasında kullanılan tahminleyicilerin önem sırasını göstermektedir. En önemli değişken yazar adı olarak seçilmiştir.

Şekil 25: Tahminleyicilerin Önem Sırası



Bu 3 önemli değişkene ait değerlerin türlere göre dağılımına ait 3 boyutlu plot grafiği Şekil 26'de gösterilmiştir.

Şekil 26: Türlerin 3 Boyutlu Grafiği



Eğitim ve test olarak 2 ye ayrılan verilerin model performansını değerlendirmek için ROC (Receiver Operating Characteristic) eğirisinden

yararlanılmıştır. ROC eğrisi 0 ile 1 arasında değer alır. Duyarlılık değerleri yani gerçek pozitif değerleri ile 1-özgüllük olan yanlış pozitif değerlerinin çakışma noktalarının birleştirilmesiyle oluşturulan grafikteki eğri ROC olarak adlandırılır. ROC eğrisinin gerçek pozitif oran ve yanlış pozitif oranları arasındaki eşitsizliği vermektedir. Duyarlılık, özgüllük ve 1-özgüllük denklemlerini tekrar gösterilmek gerekirse;

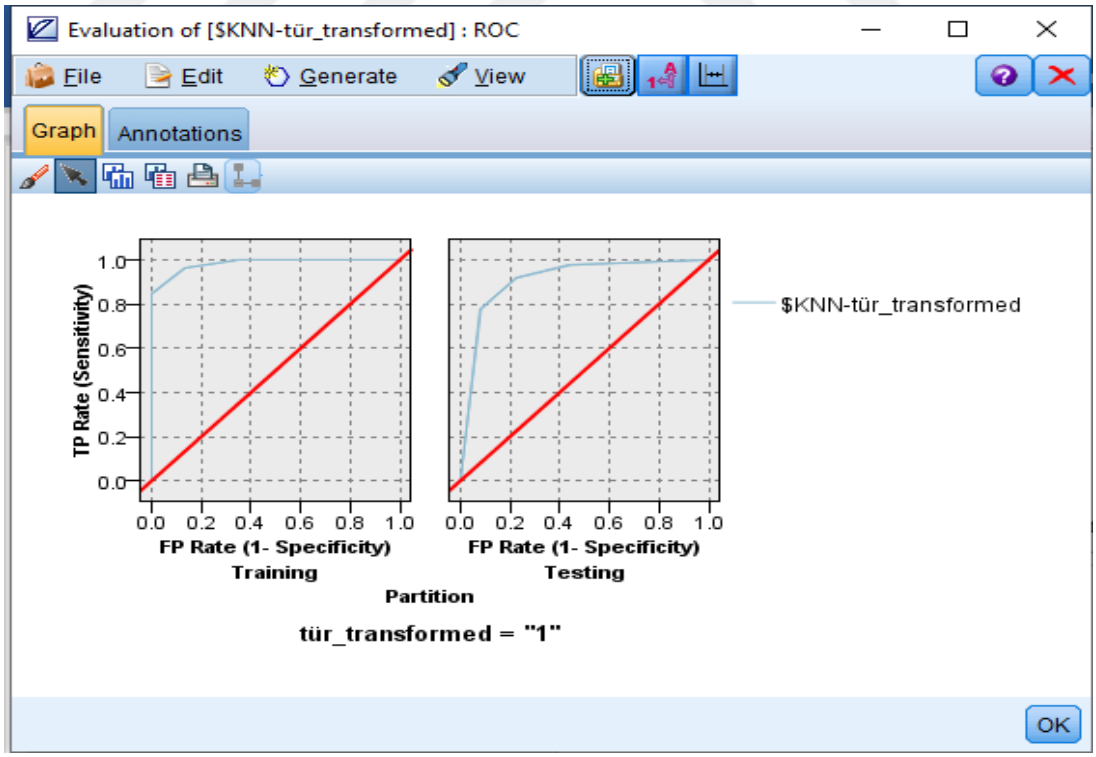
$$Duyarlılık(Sensitivity) = \frac{TP}{TP + FN}$$

$$Özgüllük\ (Specificity) = \frac{TN}{TN + FP}$$

$$FPR = 1 - Özgüllük$$

şeklindedir. ROC eğrisinde köşegenin altında kalan alan 0,5 değerindedir. Doğru sınıflandırılma yapan algoritma modelli 1 değerine en yakın değeri alacaktır.

Şekil 27: ROC Eğrisi



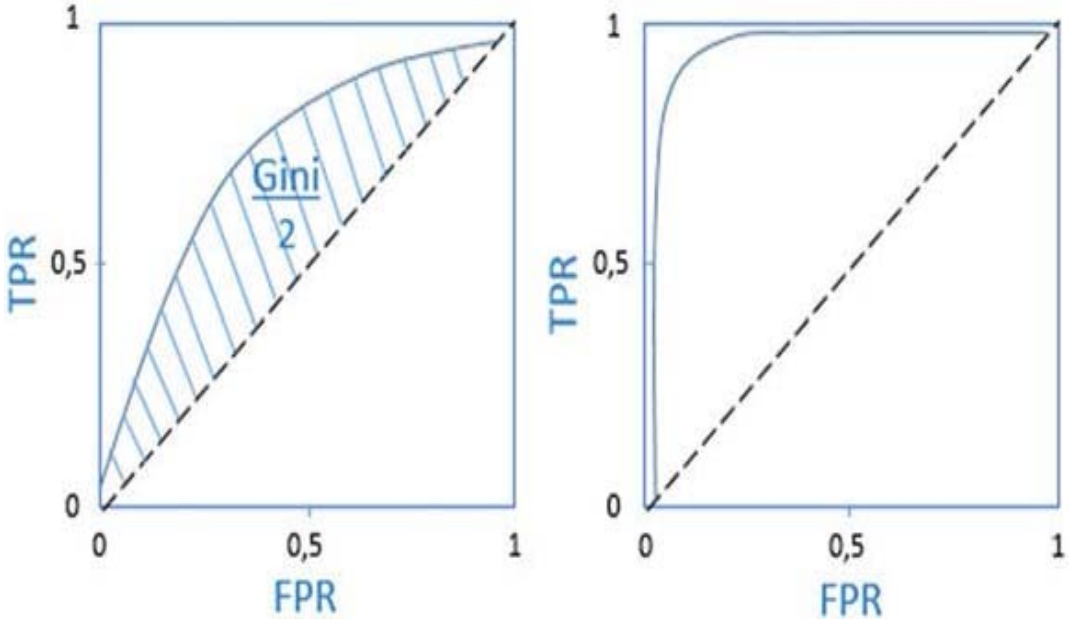
Çalışmamızın neticesinde Şekil 28'de elde edilen ROC grafiklerinden de anlaşılacağı üzere k-nn algoritması sonucunda oluşan modelin eğitim ve test

verilerinin performansları birbirine çok yakındır. ROC eğrisi altında kalan alan AUC (the Area Under this Curve) olarak adlandırılır. AUC değeri ile iki model kıyaslanabilir, yüksek değeri alan daha kesin bir sınıflandırıcı olarak seçilebilir. AUC' un yorumlama aralığı 0.5 ila 1 arasındadır. AUC değerinin 1'e eşit bulunması ise tam sınıflandırmayı göstermektedir. AUC gibi, yüksek bir sayıdaki Gini endeksi de daha iyi bir sınıflandırma modelini gösterir. AUC ile Gini endeksi arasındaki ilişki aşağıdaki denklem ile gösterilmiştir.

$$Gini = 2\,AUC - 1$$

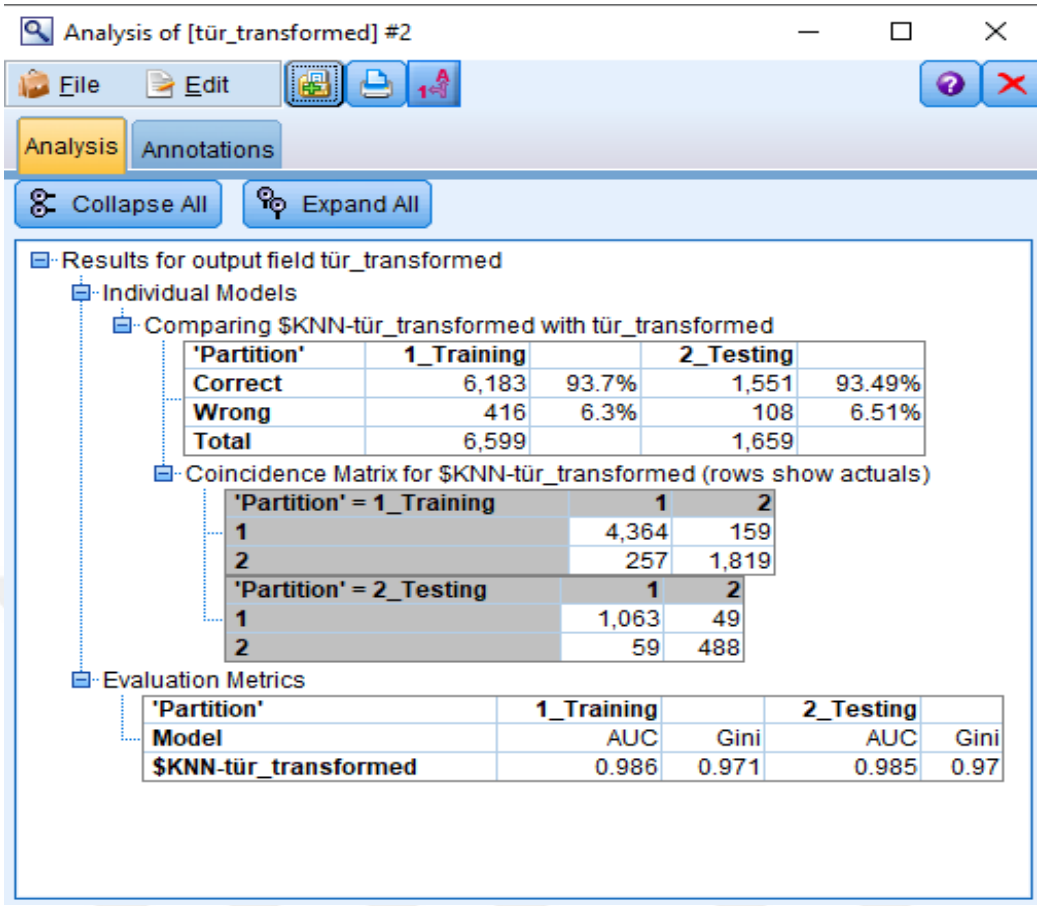
Aşağıda Şekil 28'de ilk grafikte özgün bir ROC eğrisi ve ikinci grafikte ideal bir ROC eğrisi gösterilmiştir

Şekil 28: Örnek ROC Eğrileri



Kaynak : Wendler ve Gröttrup, 2016:725

Şekil 29: Analiz Sonuçları



Analiz sonucunda elde edilen karşılaştırma sonucunda da görüleceği üzere eğitim için alınan 6599 adet verinin 6183 tanesi %93.7 oranı ile doğru sınıflandırılmıştır. Test için ayrılan 1659 adet verinin 1551 tanesi %93.49 oranı ile yine doğru sınıfa atanmıştır Aynı şekilde değerlendirme ölçütlerine bakıldığında eğitim verileri için AUC değeri 0.986 sonucu ile neredeyse 1 değerine yaklaşarak kNN algoritması ile kurulan modelin çok iyi bir performans gösterdiğini nitelemektedir. Test verileri de 0.985 oranı ile hemen hemen aynı orana sahiptir. Benzer bir şekilde Gini endeksi de AUC değeri gibi 1'e yakın bir değer aldığı için iyi bir performans olarak değerlendirilir. Gini endeksi hem eğitim hem de test verilerinin ölçümlerinde 0.97 değerini almıştır. Bu sonuçlardan yola çıkarak k-nn algoritması çalışmamız modellenmesinde en iyi algoritmalardan biri olmuştur.

SONUÇ

Dijital bir çağa geçiş aşamasında olduğumuz son yıllarda bilgiye ulaşmak çok kolaylaşmıştır. Bilgiye erişimin bu kadar kolay olması beraberinde bilgi kirliliğini de gündeme getirmektedir. Akıllı telefonların ve sosyal medyanın hayatımıza girmesi ile birlikte bilgilere kısa ve kolay yoldan erişildiği düşünülmektedir. Fakat elde edilen pek çok bilgi çoğu zaman yanlış olarak aktarılmaktadır. Bu sebeple en doğru ve güvenilir bilgi kaynağı kitaplardır. Kitapların her geçen gün doğru bilgiye ulaşmadaki önemi artmaktadır.

Satılan her kitap okunacaktır düşüncesiyle, insanların aldığı ve onlara sunulan kitap sayısının artması okuma oranlarına katkı sağlayacaktır. Ayrıca teknoloji çağında tüketim alışkanlıklarının değişmesiyle birlikte bu tüketim alışkanlıkları arasına kitabın eklenebilmesi, hem kitap sektörüne katkı sağlayacak hem de insanların kitap okuma oranını arttıracaktır. Bu bilgiler ışığında uygulamada Türkiye'de kitap satışı yapan internet sitelerinden alınan en çok satan kitap listelerindeki kitapları türlerine göre sınıflandırma yaparken hangi faktörlerin etkili olduğu belirlenmeye çalışılmıştır. Bu sayede, kitapların internet sitelerinde satışa sunum aşamasında, türlerinin sınıflandırılmasına katkı sağlanmaya çalışılmıştır. Veriler, kitap satışı yapan 21 internet sitesinin 2019 yılı Ocak ayına ait en çok satan kitap listelerinden oluşturulmuştur.

Araştırmada veri madenciliği yöntemleri uygulanarak kNN algoritması ile kitap türleri sınıflandırılmaya çalışılmıştır. Öncelikle veriler farklı fiziksel ortamlardan alındığı için ortak bir fiziksel ortama aktarılmıştır. Bu aşamada karşılaşılan uyumsuz, eksik, yinelenen ve işe yaramayan veriler ile ilgili problemler düzeltilmiştir. Kitap adı, yazar adı, site adı ve yayınevi gibi kategorik veriler ve fiyat, indirim yüzdesi, indirimli fiyatı gibi sürekli değişkenler elde edilmiştir. Ayrıca metin olarak elde edilen kitap adı, yazar adı, yayınevi vb. değişkenler sayısal etiketlerle kategorilerine ayrılmıştır. Kitap adı, yazar adı ve yayınevi verilerinde çeşitliliğin fazla olması sebebiyle analizi kolaylaştırmak için gruplama işlemi yapılmıştır. Gruplandırma sonucunda kitap adları 302, yazar adları 163, yayınevi 40 grup olarak gruplanmıştır. Ayrıca yeniden sınıflandırma işlemi yapılarak 2 türe düşürülen kitaplardan edebiyat, şiir, roman, çocuk, öykü, fantastik, çizgi roman, sanat, biyografi vb. türündekiler 1 etiketiyle; eğitim, sınav, din, psikoloji, siyaset, tarih, bilim, kişisel gelişim vb. türündekiler 2

etiketiyle adlandırılmıştır. Tüm bu işlemler SPSS Modeller 18.1 paket program kullanılarak gerçekleştirilmiştir.

Analiz edilebilir duruma getirilen veriler için en doğru sınıflandırma algoritması ve modeli belirlenmeye çalışılmıştır. Bunun için öncelikle 8258 kitap verisi %80 eğitim (training), %20 test (testing) verisi olmak üzere 2 gruba ayrılmıştır.

SPSS Modeller 18.1 paket programı kullanılarak verilerin sınıflandırma algoritmaları ile çalıştırılması sonucunda K-Yakın Komşuluk (k-nearest neighbor, kNN) algoritması %87.462 doğruluk oranı ve 0.92 AUC değeri ile en iyi performans gösteren model olmuştur. Uygulamada noktaların uzaklıklarını hesaplamak için Öklid uzaklık fonksiyonu kullanılmıştır. En düşük hata oranı değeri olan 0.1229 hata oranı ile *k=3* seçilmiştir. En iyi *k*'yı bulmak için 10 kat çapraz doğrulama işlemi uygulanmıştır. Kitap verilerine ilişkin en yakın 3 komşu ile oluşturulan modelde sırasıyla yazar adı, yayınevi ve kitap adı en önemli tahminleyiciler olmuşlardır.

Eğitim ve test olarak ikiye ayrılan verilerin model performansını değerlendirmek için ROC (Receiver Operating Characteristic) eğirisinden yararlanılmıştır. ROC eğrisinde doğru sınıflandırılma yapan algoritma modeli 1 değerine en yakın değeri alan algoritma olacaktır. ROC eğrisi ile değerlendirme yapılırken genelde ROC eğrisi altında kalan alan AUC (the Area Under this Curve) değerinden yararlanılır. Analiz tablosundan da anlaşılacağı üzere knn algoritması eğitim verilerini %93.7 ve test verilerini %93.49 oranı ile doğru sınıflandırmıştır. Ayrıca 0.98 AUC ve 0.97 gini endeksi değerlerinin 1'e çok yakın olması knn algoritmasının en iyi modellerden biri olduğunun göstergesidir.

Sonuç olarak, kitap türleri verilerinin en yakın 3 komşu temel alınarak sınıflandırma yapılması ile en iyi skoru 1. tür kitaplar oluşturmuştur. 1. tür kitaplar içerisinde roman, edebiyat (şiir, anı, biyografi, fantastik, öykü vb.) ve çocuk kitapları yer almaktadır. İnternet sitelerinin bu tür kitapların tanıtımlarını daha çok ön plana çıkarmalarının faydalı olacağı söylenebilir. Elde edilen sonuçlar ile tanıtım stratejileri geliştirilmesi aşamasında, böyle bir modelin uygulanabileceği sonucu çıkarılabilir. Ayrıca kitap satışlarında oluşan anlık değişiklikler için modelin yenilenmesi gerekebilir.

Bu çalışma süresince karşılaşılan problemlerden bundan sonra bu konuda yeni araştırmalar yapacaklara aşağıdaki öneriler getirilebilir;

1) Öncelikle tüketim alışkanları arasına kitabın eklenebilmesi ve kitap satış oranlarının artışına katkı sağlamak amacı ile yapılan bu çalışmada 21 farklı sitenin, listelerinden faydalanılmıştır. Fakat farklı fiziksel ortamlarda ve farklı formatlarda tutulan ve boyutları çok büyük olan verilerin birleştirilmesi aşaması çok zaman almış eksik, hatalı, yinelenen ve analiz için gereksiz olan çok sayıda veri ve sorun ile karşılaşılmıştır. Sistem, mevcut koşullar altında tasarlanmış ve her kitap sitesinin yayınlanan en çok satan kitap listeleri bilgilerinin yer aldığı bir veri tabanından oluşturulmuştur. Sitelerin verilerinin ortak bir veri havuzuna aktarıldığı kapsamlı bir veri tabanının mevcudiyeti halinde, sınıflandırma algoritmasının daha detaylı ve sağlıklı bir şekilde yapıldığı bir model tasarlanılması düşünülebilir.
2) İkinci olarak bu uygulamada müşteri profili (yaş, cinsiyet vb.) hakkında verilere ulaşılamadığı için müşteri profilinden yarar sağlayacak şekilde model oluşturmak mümkün olmamıştır. Ortak bir müşteri havuzu oluşturularak müşteri verileri tek bir veri tabanında birleştirilebilir. Ayrıca bu birleşen müşteri veri tabanından başka bir çalışmada inceleme konusu yapılarak farklı satış stratejileri geliştirilecek modeller kurulabilir.
3) Sistemin tasarımında geçmiş döneme ait verilere ulaşılamadığından daha önceki yıllara göre karşılaştırma yapmak mümkün olmamıştır. Verilerin zamansal olarak tutulabileceği bir veri tabanı sisteminin oluşturulması geçmiş dönem kitap türleri ve şu anki kitap türleri arasında kıyaslama yapılmasına olanak sağlayabilir. Başka bir çalışmada satışların zamansal kıyaslaması üzerine inceleme yapılabilir.
4) Son olarak en çok satan kitap listelerindeki kitaplar sürekli değişim halinde olduğu için analiz her ay tekrarlanmalıdır, sistemlerdeki değişkenlerin değişmesi durumunda sistem tanımlamaları yenilenmelidir. Günümüz koşulları ve sektördeki değişkenliklere ek olarak bilim ve teknolojideki değişiklikler de dikkate alınarak modelin güncellemesi gerekebilir.

KAYNAKLAR

Acıyan, A. A. (2008). *Ortaöğretim Öğrencilerinin Okuma Alışkanlıkları Ve Akademik Başarı Düzeyleri Arasındaki İlişki*. (Yayımlanmamış Yüksek Lisans Tezi). İstanbul: Yeditepe Üniversitesi.

Agrawal R. and Srikant. R. (1994). Fast algorithms for Mining Association Rules. *Proceedings of the 20th International Conference on Very Large Data Bases* (ss. 487-499), Düzenleyen Very Large Data Base Endowment Inc. Santiago, Chile. 12-15 Elyül 1994

Akpınar, H., (2000). Veri Tabanlarında Bilgi Keşfi ve Veri Madenciliği. *İstanbul Üniveristesi İşletme Fakültesi Dergisi*. 29(1): 1-22

Bounsaythipan C., Runsala E.R. (29 Haziran 2001). *Federico Santa Maria Teknik Üniversitesi VTT Information Technology: Overview of Data Mining for Customer Behavior Modeling Version.*
https://www.inf.utfsm.cl/~mcriff/Tesistas/listapapers/customerprofiling.pdf, (22.05.2019)

Cabena, P., Choi H. H., Kim S. I., Otska S., Reinschmidt J., Saarenvirta G., . (1999). *Intelligent Miner for Data Applications Guide*. ABD: IBM Corporation, International Technical Support Organization, Redbook

Cevahir, F. (2011). *Bir Perakende Firmasına Ait Veriler Üzerinden Veri Madenciliği Uygulaması.* (Yayınlanmış Yüksek Lisans Tezi). İstanbul: Mimar Sinan Güzel Sanatlar Üniversitesi Fen Bilimleri Enstitüsü**.**

Ching, W. K., Michael, K. P. (2002). *Advances in Data Mining and Modeling*. Hong Kong: World Scientific.

Chien, C. F., Chen, L. F. (2008). Data Mining to Improve Personnel Selection and Enhance Human Capital: A Case Study in High-Technology Industry. *Expert Systems with Applications.* 34(1): 280-290

Coşkun C., Baykal A. (2011). Veri Madenciliğinde Sınıflandırma Algoritmalarının Bir Örnek Üzerinde Karşılaştırılması. *Akademik Bilişim'11 - XIII. Akademik Bilişim Konferansı Bildirileri* (ss. 51-58), Düzenleyen İnönü Üniversitesi, Malatya. 2-4 Şubat 2011.

Dunham, M.H., (2003). *Data Mining Introductory and Advanced Topics*. New Jersey: Pearson Education Inc..

Demirel B. (2010). *Veri Madenciliğinde CHAID Algoritmasının Sosyal Güvenlik Kurumu Veri Tabanına Uygulanması*. (Yayınlanmış Yüksek Lisans Tezi). Ankara: Gazi Üniversitesi Fen Bilimleri Enstitüsü.

Emel G. G., Taşkın Ç. (2005). Veri Madenciliğinde Karar Ağaçları Ve Bir Satış Analizi Uygulaması. *Eskişehir Osmangazi Üniversitesi Sosyal Bilimler Dergisi*. 6(2): 221-239

Egan, J.P. (1975). *Signal detection theory and ROC analysis. Series in Cognition and Perception.* New York: Academic Press,

Fayyad, U. M., Piatetsky-Shapiro G., Smyth P., (1996). From Data Mining to Knowledge Discovery in Databases. *American Association for Artificial Intelligence Magazine.* 17 (3): 37-54

Fiske J., (2002). *Introduction to Communication Studies*, Londra: Routledge Taylor & Francis Group

Göral, M. A**.,** (2007). *Kredi Kartı Başvuru Aşamasında Sahtecilik Tespiti İçin Bir Veri Madenciliği Modeli*, (Yayınlanmış Yüksek Lisans Tezi). İstanbul: İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü

Giudici, P., (2003). *Applied Data Mining: Statistical Methods for Business and Industry.* İngiltere: John Wiley & Sons Ltd.,

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

Han J., Kamber M., (2001). *Data Mining Concepts and Techniques*, ABD, San Francisco: Morgan Kaufman,

Hand D. J. (1999). Statistics and Data Mining: Intersecting Disciplines *SIGKDD Explorатıons Magazine* 1(1): 16-19

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

IBM Corporation (2011). *IBM SPSS Modeler 14.2 User's Guide.* U.S.A.: IBM Corp.

Khan M., Ding Q.,Perrizo W.(2002). K-Nearest Neighbor Classification on Spatial Data Streams Using P-Trees, *6. Pasifik Asya Knowledge discovery and Data Mining Konferansı* (ss. 517-528), Düzenleyen PAKKDD. TAİPEİ, Taiwan, 6-8 Mayıs 2002.

Koyuncugil A.S., (2006). *Bulanık Veri Madenciliği Ve Sermaye Piyasalarına Uygulanması*. (Yayınlanmış Doktora Tezi). Ankara: Ankara Üniversitesi Fen Bilimleri Enstitüsü.

Larose Daniel T. (2005). *Discovering Knowledge in Data: An Introduction to Data Mining*. ABD: John and Wiley Sons Incorporated

Moss, L.T., Atre, S. (2003). *Business Intelligence Roadmap: The Complete Project Lifecycle for Decision-Support Applications*. ABD: Pearson Addison-Wesley

Oğuzlar A. , (2004). *Veri Madenciliğine Giriş*. Bursa: Ekin Kitabevi

Özkan Y., (2013). *Veri Madenciliği Yöntemleri*. İstanbul: Papatya Yayıncılık

Silahtaroğlu G., (2013). *Veri Madenciliği Kavram ve Algoritmaları*. İstanbul: Papatya Yayıncılık.

Shannon, C.E., (1948). *A Mathematical Theory of Communication*, ABD: The Bell System Technical Journal

Şekeroğlu S.,(2010). *Hizmet Sektöründe Bir Veri Madenciliği Uygulaması*, (Yayınlanmış Yüksek Lisans Tezi). İstanbul: İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü.

Tan P., Steinbach M., Kumar V., (2006). *Introduction to Data Mining.* Boston: Pearson Addison-Wesley.

Thomas, E., H., Galambos, N., (2004)**.** What Satısfıes Students? Mining Student – Opinion Human Sciences Press, Data With Regression and Decision Tree Analysis**.** *Research in Higher Education Human Sciences Press.* 45(3): 251-269

Wendler T., Gröttrup S. , (2016). *Data Mining with SPSS Modeler-Theory, Exercises and Solutions.* Almanya*:* Springer International Publishing,

Yaralıoğlu, K.,(2004)**.** *Uygulamada Karar Destek Yöntemleri,* İzmir: İlkem Ofset