

 **e**Kampus
a n a d o l u m
e K a m p ü s
ve
a n a d o l u m o b i l
dilediğin yerden,
dilediğin zaman,
öğrenme fırsatı!



(ekampus.anadolu.edu.tr)



(mobil.anadolu.edu.tr)

ekampus.anadolu.edu.tr

 Takvim	 Duyurular	 Ders Kitabı (PDF)	 Epub	 Html5
 Video	 Canlı Ders	 Sesli Kitap	 Ünite Özeti	 Sesli Özeti
 Sorularla Öğrenelim	 Alıştırma	 Deneme Sınavı	 İnfografik	 Etkileşimli İçerik
 Bilgilendirme Panosu	 Çıkmış Sınav Soruları	 Sınav Giriş Bilgisi	 Sınav Sonuçları	 Öğrenci Toplulukları



aosdestek.anadolu.edu.tr

444 10 26

www.anadolu.edu.tr

T.C. ANADOLU ÜNİVERSİTESİ YAYINI NO: 3399
AÇIKÖĞRETİM FAKÜLTESİ YAYINI NO: 2251

VERİ MADENCİLİĞİ

Yazarlar

Dr.Öğr.Üyesi Şenay LEZKİ (Ünite 1)

Doç.Dr. Harun SÖNMEZ (Ünite 2, 3)

Dr.Öğr.Üyesi Alper BEKKİ (Ünite 4, 5, 8)

Dr.Öğr.Üyesi Levent TERLEMEZ (Ünite 6)

Doç.Dr. Cengiz BAL (Ünite 7)

Editör

Prof.Dr. Fikret ER

Bu kitabın basım, yayım ve satış hakları Anadolu Üniversitesi'ne aittir.
“Uzaktan Öğretim” teknüğine uygun olarak hazırlanan bu kitabı bütün hakları saklıdır.
İlgili kuruluştan izin almadan kitabı tümü ya da bölmeleri mekanik, elektronik, fotokopi, manyetik kayıt
veya başka sekillerde çoğaltılamaz, basılamaz ve dağıtılamaz.

Copyright © 2016 by Anadolu University
All rights reserved

No part of this book may be reproduced or stored in a retrieval system, or transmitted
in any form or by any means mechanical, electronic, photocopy, magnetic tape or otherwise, without
permission in writing from the University.

Öğretim Tasarımcısı
Öğr.Gör. Orkun Şen

Grafik Tasarım Yönetmenleri

*Prof. Teyfik Fikret Uçar
Doç.Dr. Nilgün Salur
Öğr.Gör. Cemalettin Yıldız*

Dil ve Yazım Danışmanı
Öğr.Gör. Sinem Türkyılmaz

Ölçme Değerlendirme Sorumlusu
Öğr.Gör. Emrah Emre Özkeskin

Kapak Düzeni
Prof.Dr. Halit Turgay Ünalan

Grafikerler
*Gülşah Karabulut
Kenan Çetinkaya
Özlem Çayırlı
Burcu Güler
Ayşegül Dibek*

Dizgi ve Yayıma Hazırlama
Kitap Hazırlama Grubu

Veri Madenciliği

E-ISBN
978-975-06-3326-3

Bu kitabın tüm hakları Anadolu Üniversitesi'ne aittir.

ESKİŞEHİR, Ocak 2019
3221-0-0-2202-V01

İçindekiler

Önsöz	viii
-------------	------

Temel Kavramlar.....	2	1. ÜNİTE
GİRİŞ	3	
VERİ MADENCİLİĞİNİN TARİHSEL GELİŞİMİ	3	
VERİ MADENCİLİĞİNE ETKİ EDEN DİSİPLİNLER	5	
VERİ MADENCİLİĞİ KAVRAMI	6	
VERİTABANLARINDA BİLGİ KEŞFİ SÜRECİ	10	
Amacın Tanımlanması	12	
Veriler Üzerinde Ön İşlemlerin Yapılması	12	
Verilerin Toplanması ve Birleştirilmesi	13	
Verilerin Temizlenmesi	13	
Verilerin Yeniden Yapılandırılması	17	
Modelin Kurulması ve Değerlendirilmesi	17	
Modelin Kullanılması ve Yorumlanması	18	
Modelin İzlenmesi	18	
VERİ MADENCİLİĞİNDE KULLANILAN MODELLER	18	
Tahmin Edici Modeller	19	
Tanımlayıcı Modeller	22	
VERİ MADENCİLİĞİNİN DİĞER VERİ ANALİZİ YAKLAŞIMLARI İLE KARŞILAŞTIRILMASI	23	
VERİ MADENCİLİĞİNİN UYGULANDIĞI ALANLAR	24	
Pazarlama Alanındaki Uygulamalar	24	
Finans Alanındaki Uygulamalar	25	
Sağlık Alanındaki Uygulamalar	25	
Endüstri ve Mühendislik Alanındaki Uygulamalar	25	
Eğitim Alanındaki Uygulamalar	25	
Özet	26	
Kendimizi Sınayalım	28	
Kendimizi Sınayalım Yanıt Anahtarı	29	
Sıra Sizde Yanıt Anahtarı	29	
Yararlanılan ve Başvurulabilecek Kaynaklar	30	
Yararlanılan İnternet Kaynakları	31	
R Yazılımı	32	2. ÜNİTE
GİRİŞ	33	
R YAZILIMININ ELDE EDİLMESİ	35	
R YAZILIMIN TEMELLERİ	36	
Temel Komutlar	36	
Vektörler	37	
Matrİsler	40	
Mantık Operatörleri	42	
List Nesneleri	44	
Data Frame	46	
KİŞİSEL FONKSİYON YAZMA	47	
HAZIR VERİ AKTARIMI	49	
EK KÜTÜPHANE YÖNETİMİ	50	

Özet	52
Kendimizi Sınayalım	53
Kendimizi Sınayalım Yanı Anahtarları	54
Sıra Sizde Yanı Anahtarları	54
Yararlanılan ve Başvurulabilecek Kaynaklar	55

3. ÜNİTE**Verinin Hazırlanması 56**

GİRİŞ	57
TEMEL DEĞİŞKEN TIPLERİ	58
İsimsel (Nominal) Değişkenler	58
İkili (Binary) Değişkenler	59
Sıra Gösteren (Ordinal) Değişkenler	59
Tam sayılı (Integer) Değişkenler	59
Aralıklı Ölçümlendirilmiş (Interval-Scaled) Değişkenler	59
Oranlı Ölçümlendirilmiş (Ratio-Scaled) Değişkenler	59
VERİ HAZIRLAMA	60
Veri Temizleme	60
Eksik Veri	61
Gürültülü Veri	62
Tutarsız Veri	62
Veri Birleştirme	63
Veri İndirgeme	63
Veri Küpü Birleştirme	63
Boyut İndirgeme	63
Veri Sıkıştırma	64
Büyük Sayıların İndirgenmesi	64
Veri Dönüştürme	64
Enk-Enb Normalleştirme	65
z-Skor Normalleştirme	67
Ondalık Ölçekleme	68
Özet	70
Kendimizi Sınayalım	71
Kendimizi Sınayalım Yanı Anahtarları	72
Sıra Sizde Yanı Anahtarları	72
Yararlanılan ve Başvurulabilecek Kaynaklar	73

4. ÜNİTE**Benzerlik ve Uzaklık Ölçüleri 74**

GİRİŞ	75
DÖNÜŞÜMLER	76
BASIT NİTELİKLER ARASINDAKİ YAKINLIK	79
BENZERLİK VE UZAKLIK ÖLÇÜLERİ	80
NİCEL DEĞİŞKENLER İÇİN YAKINLIK ÖLÇÜLERİ	81
Öklid ve Karesel Öklid Uzaklığı	81
Öklid ve Karesel Öklid Uzaklığının R Çözümü	83
Karl Pearson Uzaklığı	84
Manhattan (City-Block) Uzaklığı	84
Manhattan (City-Block) Uzaklığının R Çözümü	85
Minkowski Uzaklığı	85
Minkowski Uzaklığının R Çözümü	86

Pearson Korelasyon Katsayısı ve Korelasyon Uzaklığı	86
Pearson Korelasyon Katsayısı ve Korelasyon Uzaklığının R Çözümü	88
Açışal Benzerlik (Cosine Similarity)	89
Açışal Benzerlik (Cosine Similarity) R Çözümü	90
Mahalanobis Uzaklığı	91
Mahalanobis Uzaklığının R Çözümü	91
İKİ SONUÇLU (BINARY) DEĞİŞKENLER İÇİN YAKINLIK ÖLÇÜLERİ	92
Basit Eşleştirme Katsayısı ve Uzaklığı	93
Basit Eşleştirme Katsayısı ve Uzaklığı R Çözümü	94
Binary Öklid ve Binary Karesel Öklid Uzaklığı	95
Binary Öklid ve Binary Karesel Öklid Uzaklığı R Çözümü	96
Jaccard Benzerlik Katsayısı ve Uzaklığı	96
Jaccard Benzerlik Katsayısı ve Uzaklığı R Çözümü	98
Özet	99
Kendimizi Sınayalım	100
Kendimizi Sınayalım Yanı Anahtarları	101
Sıra Sizde Yanı Anahtarları	101
Yararlanılan ve Başvurulabilecek Kaynaklar	101
İlişki Kuralları.....	102
GİRİŞ	103
İLİŞKİ KURALLARI	103
PAZAR SEPETİ ANALİZİ	104
İLGINÇ KURAL BELİRLEME ÖLÇÜTLERİ (RULE INTERESTINGNESS MEASURES)	107
Destek (Support)	107
Destek Eşik Değeri	109
Güven (Confidence)	110
Güven Eşik Değeri	111
Kaldıraç (Lift)	111
İLİŞKİ KURALI BELİRLEME AŞAMALARI	113
Apriori Algoritması	113
İLİŞKİ KURALLARI R ÇÖZÜMÜ	120
Özet	122
Kendimizi Sınayalım	123
Kendimizi Sınayalım Yanı Anahtarları	124
Sıra Sizde Yanı Anahtarları	124
Yararlanılan ve Başvurulabilecek Kaynaklar	125
Karar Ağaçları	126
GİRİŞ	127
KARAR AĞAÇLARI	129
Ayırma Kriterleri	132
Entropi İndeksi ile En İyi Ayırıcı Niteliğin Seçilmesi	132
Gini İndeksi ve Ayırıcı Niteliğin Belirlenmesi	138
Karar Ağacı Oluşturma Algoritmaları	140
Karar Ağacı Budama Süreci ve Karar Ağacının Performansının Test Edilmesi ..	141
SINIFLANDIRMA VE REGRESYON AĞAÇLARININ R ÇÖZÜMÜ	142
R'ye Veri Aktarma	143

5. ÜNİTE**6. ÜNİTE**

csv Dosyası ile R'ye Veri Aktarma	143
Kopyala-Yapıştır Komutu ile R'ye Veri Aktarma	145
Veritabanı Erişimi ile R'ye Veri Aktarma	146
Sınıflandırma ve Regresyon Ağaçlarının rpart Paketi ile Çözümü	148
Özet	156
Kendimizi Sinayalım	158
Yaşamın İçinden	159
Kendimizi Sinayalım Yanıt Anahtarı	160
Sıra Sizde Yanıt Anahtarı	160
Yararlanılan ve Başvurulabilecek Kaynaklar	162

7. ÜNİTE**Kümeleme Analizi 164**

GİRİŞ	165
KÜMELEME ANALİZİ	166
UZAKLIK VE BENZERLİK ÖLÇÜLERİ	167
KÜMELEME YÖNTEMLERİ	168
AŞAMALI KÜMELEME YÖNTEMLERİ	168
Birleştirici Aşamalı Kümeleme Yöntemleri	168
Ayrıcı Aşamalı Kümeleme Yöntemleri	168
Dendrogramlar (Ağaç Diyagramları)	169
BİRLEŞTİRİCİ KÜMELEME YÖNTEMLERİ	171
Tek Bağlantı Kümeleme Yöntemi	171
Tam Bağlantı Kümeleme Yöntemi	172
Ortalama Bağlantı Kümeleme Yöntemi	173
McQuitty Bağlantı Kümeleme Yöntemi	173
Küresel Ortalama Bağlantı Kümeleme Yöntemi	173
Medyan Bağlantı Kümeleme Yöntemi	174
Ward Bağlantı Kümeleme Yöntemi	174
R PROGRAMINDA TEK BAĞLANTI KÜMELEME YÖNTEMİ	
UYGULAMASI	174
AŞAMALI OLMAYAN KÜMELEME YÖNTEMLERİ	178
k-Ortalamalar Yöntemi	179
k-Medyanlar Yöntemi	179
k-Medoidler Yöntemi	180
k-Ortalamalar Yönteminin Uygulanması	180
R PROGRAMINDA K-ORTALAMALAR KÜMELEME YÖNTEMİ	
UYGULAMASI	181
Özet	185
Kendimizi Sinayalım	186
Sıra Sizde Yanıt Anahtarı	187
Kendimizi Sinayalım Yanıt Anahtarı	187
Yararlanılan ve Başvurulabilecek Kaynaklar	188

8. ÜNİTE**Web Madenciliği ve Sosyal Medya Madenciliği..... 190**

GİRİŞ	191
VERİ MADENCİLİĞİ VE WEB MADENCİLİĞİ	192
WEB MADENCİLİĞİ SÜRECİ	194
WEB MADENCİLİĞİ VERİ KAYNAKLARI	196
Web Verisinin Özellikleri	197

WEB MADENCİLİĞİNİN SINIFLANDIRILMASI	197
Web İçerik Madenciliği	198
Web Arama	198
Kısa Metin İşleme	199
Bilgi Keşfi	199
Web Görüş Madenciliği	199
Web Yapı Madenciliği	199
İnternette Arama ve Bağlantı Köprüleri	200
Atıf Analizi	200
Web Topluluğu Keşfi	200
Web Şeması Ölçüm ve Modellemesi	200
Web Sayfalarının Sınıflandırılması	201
Web Kullanım Madenciliği	201
Web Kullanım Madenciliği Aşamaları	201
Web Kullanım Madenciliği Temel Uygulama Alanları	203
Kişiselleştirme (Personalization)	203
Sistem Geliştirme (System Improvement)	203
Web Sitesi Güncelleme (Site Modification)	204
İş Zekası (Business Intelligence)	204
Kullanım Karakteristiği (Usage Characterization)	204
SOSYAL MEDYA MADENCİLİĞİ	205
R ile Twitter Verisinin Analizi	207
Analiz I: Kişisel Twitter Verilerinizin Analizi	207
Analiz II : Kelime Bulutu	209
R ile Facebook Verisinin Analizi	214
Özet	221
Kendimizi Sinayalım	222
Kendimizi Sinayalım Yanıt Anahtarı	223
Sıra Sizde Yanıt Anahtarı	223
Yararlanılan ve Başvurulabilecek Kaynaklar	224

Önsöz

Sevgili öğrenciler,

Son yıllarda teknolojik gelişmelere bağlı olarak ortaya çıkan veri sayısında büyük bir artış meydana gelmiştir. Bugün bir çok magazin, kitap ve televizyon programında yığınlar halinde ortaya çıkan veri kavramından bahsedilmektedir. Bir günde gönderilen e-posta sayısı 100 milyarın üzerindedir. Gün boyu atılan Tweet sayısı beşyüzbin rakamının üzerine çıkmaktadır. Dünya üzerinde 2016 yılı içerisinde HIV/AIDS ile alakalı olarak hayatını kaybeden kişi sayısı 1.600.000'i aşmaktadır. Sosyal medya kavramı artık nerede ise tüm evlerin konuşulan ortak bir konusu haline gelmiştir. Veri miktarının yüzlerle değil de milyonlar ile telfafuz edildiği bu dönemde, elde edilen verinin en verimli biçimde değerlendirilmesi, yorumlanması büyük bir önem arz etmektedir. Elinizde bulunan bu eser verinin derlenmesi, düzenlenmesi ve size anlatmaya çalıştığı öğeleri görebilmenizi sağlayacak teknikleri bir araya getirerek, en sade hali ile sizlere aktarmaktadır.

Veri madenciliğinin tarihi bilgisayarların hayatımıza girmesiyle başlamıştır. 1950'li yıllarda ilk bilgisayarların geliştirilme ve kullanım amacı sayı ve karmaşık hesaplamaları kolaylıkla yapabilmekti. Daha sonra kullanıcıların ihtiyaçları doğrultusunda, bilgisayarlar veri depolama işlemleri için de kullanılmaya başlanmıştır. Verilerin depolanması ihtiyacı ile birlikte, 1960'lı yillardan itibaren teknoloji dünyası veri tabanı kavramı ile tanışmıştır. 1960'ların sonunda ise basit öğrenmeli bilgisayarlar geliştirilmiştir.

Teknolojinin hızla ilerlediği bu dönemde veri analisinin ayrılmaz bir parçası da bilgisayar donanım ve yazılımlarıdır. Bu eserde sizi yazılım maliyeti ile karşı karşıya bırakmayan ve ücretsiz olarak indirilip kullanılabilen R yazılımı kullanılmıştır. Mümkün olduğunda, ele alınan veri madenciliği tekniklerine ilişkin R programlama adımları, gerçek hayat örnekleri ile sizlere aktarılmıştır.

Kitabın oluşturulması sürecinde başta yazarlar olmak üzere geçen herkese sonuz teşekkürlerimi sunarım.

Kitapta yer alan bilgilerin ve uygulama örneklerinin gerçek yaşamınızda siz öğrencilerimize ve konuya ilgilenen herkese faydalı olmasını diliyorum.

Editör
Prof.Dr. Fikret ER

1

Amaçlarımız

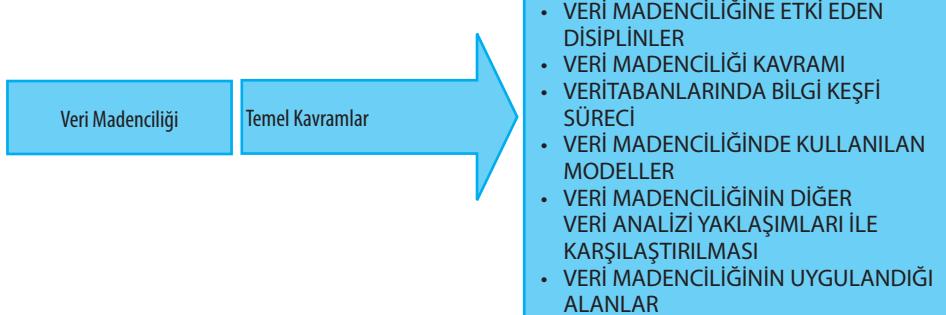
Bu üniteyi tamamladıktan sonra;

- 🕒 Veri madenciliğinin tarihsel gelişimini özetleyebilecek,
- 🕒 Veri madenciliğine etki eden disiplinleri betimleyebilecek,
- 🕒 Veri madenciliği kavramını tanımlayabilecek,
- 🕒 Veritabanlarında bilgi keşfi sürecini açıklayabilecek,
- 🕒 Veri madenciliğinde kullanılan modellere ilişkin özellikleri özetleyebilecek,
- 🕒 Veri madenciliğini diğer veri analizi yaklaşımları ile karşılaştırabilecek,
- 🕒 Veri madenciliğinin uygulandığı alanları örnekleyebilecek bilgi ve becerilere sahip olabileceksiniz.

Anahtar Kavramlar

- Veri Madenciliği
- Veritabanlarında Bilgi Keşfi
- Veritabanı
- Veri Ambarı
- OLAP
- Örüntü
- Kayıp Veri
- Gürültülü Veri

İçindekiler



Temel Kavramlar

GİRİŞ

İletişim ve bilişim teknolojilerinde yaşanan gelişmeler dünyada her şeyin hızla değişmesine neden olmaktadır. İster kâr amaçlı işletmeler, ister diğer kurum ve kuruluşlar açısından olsun, değişimlere ayak uydurabilmek başarı için önemli bir gereklilikdir. İşetmeler açısından ele alındığında bu değişimler; ekonomik koşullarda, iş yapma biçimlerinde, müşteri bekłentilerinde, müşteri eğilimlerinde, rakiplerin stratejilerinde vb. ortaya çıkmaktadır. İşetmelerin bu değişimlere ayak uydurabilmesi, rakipleriyle yarışabilmesi ve varlıklarını başarılı bir biçimde sürdürbilmesi için, işletmelerde karar verici konumunda olan yöneticilerin, doğru kararlar vererek doğru stratejiler belirlemeleri gerekmektedir. Bu da ancak zamanında elde edilebilen doğru bilgilerin kullanımıyla mümkün olacaktır. Bu nedenle işletmelerin iş süreçlerinden ve işletme dışından elde ettikleri verileri karar verme sürede anlamlı bilgilere dönüştürebilmeleri önemlidir.

Günümüzde bilişim teknolojisinde gelinen noktada çok büyük miktarda verinin kolaylıkla elde edilmesi ve kaydedilerek saklanması olanaklı hale gelmiştir. Bununla birlikte veriler tek başlarına bir anlam ifade etmemeye belirli bir amaca yönelik olarak işlendiklerinde anlamlı bilgilere dönüşürler. Verilerin kolaylıkla elde edilip saklanabilmelerine karşın, bu verilerden anlamlı bilgilere ulaşabilmek aynı derecede kolay değildir. Anlamlı bilgilere ulaşabilmek amacıyla geçmişten beri kullanılan farklı yöntemler bulunmaktadır. Bununla birlikte verilerin analiz edilmesinde kullanılan geleneksel yöntemler veri miktarında meydana gelen büyük artış karşısında yetersiz kalmaya başlamıştır. Veri madenciliğinin ortaya çıkışının büyük miktarda veriyi analiz edebilme ve işleyebilme ihtiyacından kaynaklanmıştır.

Veri madenciliğinin amacı, çok büyük miktarda ve karmaşık durumdaki veriler içinden geleneksel yöntemlerle elde edilemeyecek bilgilere ulaşma ve bu bilgileri rakiplere fark yaratacak kararlarda kullanabilmeye olanak sağlamaktır. Buradan anlaşılırabileceğü üzere veri madenciliği tek başına çözümün kendisi olmayıp çözüme ulaşacak kararın verilmesine destek sağlayacak bilgilerin ortaya çıkarılmasında kullanılan bir araçtır.

VERİ MADENCİLİĞİNİN TARİHSEL GELİŞİMİ

Veri madenciliğinin tarihi bilgisayarların hayatımıza girmesiyle başlamıştır. 1950'li yıllardaki ilk bilgisayarların geliştirilme ve kullanım amacı sayı ve karmaşık hesaplamaları kolaylıkla yapabilmekti. Daha sonra kullanıcıların ihtiyaçları doğrultusunda, bilgisayarlar veri depolama işlemleri için de kullanılmaya başlanmıştır. Verilerin depolanması ihtiyaç ile birlikte, 1960'lı yillardan itibaren teknoloji dünyası *veri tabanı* kavramı ile tanışmıştır. 1960'ların sonunda ise basit öğrenmeli bilgisayarlar geliştirilmiştir. Buna karşın,

Perceptron, insan beyninde yer alan sinir hücrelerinin (nöronların) ilk yapay modeline verilen isim olup algılayıcı, fark edici anlamadır. 1957 yılında Frank Rosenblatt tarafından geliştirilen ve tekrar eden, benzerlik gösteren özelliklerin bilgisayar tarafından algılanabilmesini sağlayan bir algoritmadır.

günümüzdeki sinir ağlarının temeli olarak bilinen **perceptron**'ların yalnızca çok basit olan kuralları öğrenebileceği, bazı basit mantıksal işlemlerde ise yetersiz kaldığı 1969'da Minsky ve Papert tarafından ortaya konulmuştur. Zaman içinde giderek büyüyen veri tabanlarının organizasyonu, düzenlenmesi ve yönetimi de doğal olarak zorlaşmıştır. Bu zorlukların üstesinden gelebilmek amacıyla ise veri modelleme kavramı ortaya atılmıştır. İlk veri modelleri; Hiyerarşik Veri Modeli ve Ağ Veri Modeli olarak adlandırılan basit veri modelleridir. 1970'lerde İlişkisel Veri Tabanı Yönetim Sistemleri uygulamaları kullanılmaya başlanmıştır, bu konuya ilgilenen uzmanlar basit kurallara dayanan uzman sistemler geliştirmiştir ve basit anlamda makine öğrenimini sağlamışlardır. 1980'lerde veri tabanı yönetim sistemleri yaygınlaşmış ve pek çok farklı alanda uygulanır olmuştur. Özellikle işletmeler, müşterileri, rakipleri ve ürünlerine ilişkin verileri düzenli biçimde saklamak amacıyla veri tabanları oluşturmuştur.

Kullanıcı ihtiyaçları doğrultusunda şekillenen veri tabanları ve veri modelleme çeşitlerinin kullanımı hızlı biçimde yaygınlaşıken buna bağlı olarak donanım öğeleri de bu ihtiyaçlara cevap verecek biçimde geliştirilmiştir. Günümüzde bellek kapasitesi GigaByte ve TeraByte'tan sonra Peta, Exa ve Zetta ön ekleriyle ifade edilen boyutlara ulaşmıştır. Bu nın sonucu olarak milyarlarca Byte veri fizikselleşmiş olarak çok küçük boyutlardaki donanım öğelerinde saklanabilir hâle gelmiştir. Bu kadar büyük miktarda verinin saklanması sorun olmasa da verilerin düzenlenmesi, organize edilmesi ve istenilen veriye hızlıca ulaşılabilmesi büyük bir sorun durumuna gelmiştir.

1990'lara gelindiğinde ise artık araştırma konusu; veri miktarının sürekli katlanarak arttığı veri tabanları içinden, faydalı bilgilerin nasıl çıkarılacağı konusudur. Bu amaçla pek çok çalışma ve yayın yapılmıştır. Bu çalışmaların en önemlisi, 1989'da yapılan KDD (Knowledge Discovery in Database) IJCAI-89 Veri Tabanlarında Bilgi Keşfi Çalışma Grubu toplantısıdır. 1991 yılında ise KDD (IJCAI)-89'un sonuç bildirgesi sayılabilenek "Knowledge Discovery in Real Databases: A Report on the IJCAI-89 Workshop" makalesi ile Bilgi Keşfi ve Veri Madenciliği ile ilgili temel tanım ve kavramlar ortaya konmuştur. Bu makaleden sonra süreç daha da hızlanmış ve 1992 yılında veri madenciliği için ilk yazılım geliştirilmiştir. 2000'li yıllarda veri madenciliği sürekli gelişmiş ve hemen hemen tüm alanlara uygulanmaya başlanmıştır. Alınan sonuçların faydalari görüldükçe bu alana ilgi artmıştır.

Günümüze geldiğimizde veri madenciliğinin pek çok alanda yaygın olarak kullanıldığılığını görebiliriz. Karar verme sürecinde ihtiyaç duyulan veri analizini gerçekleştirdiği için, operasyonel kararların ötesinde stratejik karar verme süreçlerinde de oldukça önemli bir yere sahiptir. İşletmeler, günümüzde yoğun olarak kullandıkları Müşteri İlişkileri Yönetimi (CRM) ve Kurumsal Kaynak Planlaması (ERP) gibi uygulamalar ve teknikler aracılığıyla veri madenciliği yapmaktadır.

Veri madenciliğinin tarihsel gelişim süreci, Tablo 1.1'de özetlenmiştir.

Tablo 1.1
Veri Madenciliğinin
Tarihsel Süreci

Kaynak: Savaş,
Topaloğlu ve Yılmaz,
(2012), s.5.

1950'ler	<ul style="list-style-type: none"> İlk bilgisayarlar (sayım ve hesaplama amaçlı)
1960'lar	<ul style="list-style-type: none"> Verilerin depolanması ve veritabanları Perceptronlar
1970'ler	<ul style="list-style-type: none"> İlişkisel Veritabanı Yönetim Sistemleri Basit kurallara dayanan uzman sistemler ve makine öğrenimi
1980'ler	<ul style="list-style-type: none"> Büyük miktarda veri içeren veri tabanları SQL sorgu dili
1990'lar	<ul style="list-style-type: none"> Veritabanlarında Bilgi Keşfi Çalışma Grubu ve Sonuç Bildirgesi Veri madenciliği için ilk yazılım
2000'ler	<ul style="list-style-type: none"> Tüm alanlar için veri madenciliği uygulamaları

Göründüğü gibi bugün veri madenciliği olarak ifade ettiğimiz kavrama ilişkin çalışmalar aslında ilk olarak, 1960'lı yıllarda bilgisayar sistemlerinin, verilerin analizi ve problemlerin çözümü amacıyla kullanılmaya başlanmasıyla birlikte ortaya çıkmıştır. Buna göre bilgisayarlarda depolanan veriler üzerinde, yeterli uzunlukta bir tarama yapıldığında, istenilen verilere erişmenin olanaklı olacağı gerçeği kabul edilmiştir. Bu işleme ilk zamanlarda veri taraması, veri yakalaması gibi adlandırmalar yapılmıştır. Veri madenciliği adlandırması ise yukarıda da belirtildiği gibi 1990'lı yıllara gelindiğinde, bilgisayar mühendisleri tarafından kullanılmaya başlanmıştır.

VERİ MADENCİLİĞİNE ETKİ EDEN DİSİPLİNLER

Veri madenciliği işlemleri ile amaçlanan; veri analizinde geleneksel istatistiksel yöntemler yerine bu yöntemlerin yetersizliklerini giderecek yeni yaklaşımları, bilgisayar algoritmalarının kullanımı ile uygulamak ve istenilen analizi hızlı ve sağlıklı bir biçimde gerçekleştirmektir. Bu yeni yaklaşımların temelinde istatistik, makine öğrenimi, veritabanı sistemleri önemli bir yer tutmaktadır.

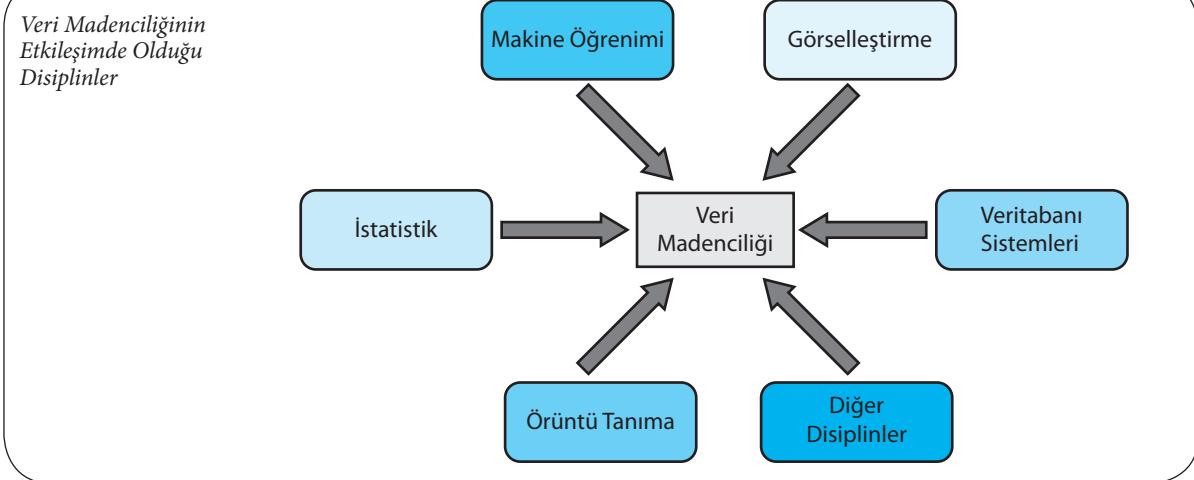
Istatistik, verilerin analizi ve değerlendirilmesi konusunda geçmişten günümüze yoğun bir biçimde kullanılan bir disiplindir. Bilgisayar sistemlerinde hem donanım hem de yazılım alanında sağlanan gelişmeler doğal olarak istatistik alanını da etkilemiştir. İstatistiksel çalışmaların bilgisayar desteğiyle daha güçlü biçimde yapılması, daha önce gerçekleştirilememiş çok mümkün olmayan istatistiksel araştırmaları ve analizleri yapılabilebilir hale getirmiştir. Bu anlamda 1990'lardan sonra, ilgilenilen verinin yiğinlar içinden çekilip çıkarılması ve analizinin yapılarak kullanımına hazır hale getirilmesi sürecinde istatistik, veri madenciliği ile ortak bir platformda ve sıkı bir çalışma birlikteliği içinde olmuştur.

Veri madenciliği çalışmalarında etkili olan ve yapay zekâ çalışmalarının da temelini oluşturan *makine öğrenimi*, kısaca bilgisayarların bazı işlemlerden çıkarsamalar yaparak yeni işlemler üretmesi olarak tanımlanabilir. Makine öğrenimi, insan öğrenmesinde söz konusu olan özelliklerin algoritmalar yardımıyla bilgisayarlara da uygulanabileceği ve bilgisayarların da insanlar gibi öğrenebileceği düşüncesini temel alan bir disiplindir. İnsanlar nasıl öğrenirler? İnsanlar çocukluk dönemlerinden itibaren öğrenmeye başlarlar. Bu, etraflarında gördükleri tüm nesneleri gözleme ve bu gözlemler aynı türde nesneler üzerinde tekrarlandıkça nesneleri kavramlara dönüştürme biçiminde gerçekleşir. Aynı türde nesnelere ilişkin farklı örnekleri görmeyi, incelemeyi sürdürdükçe nesneye ilişkin kavram netleşir ve benzer örnekleri ilgili nesne sınıfına konumlandırarak bir sınıflama modeli oluşturur. Örneğin, ilk kez kedi gören bir çocuğa gördüğü varlığın bir kedi olduğu söylendiğinde bu bilgiyi alan çocuk, başka bir gün başka bir kedi gördüğünde bir önceki deneyimi hatırlayarak o varlığın kedi olduğunu düşünür. Bu deneyim tekrarlandıkça öğrenme de gerçekleşmiş olur. Artık herhangi bir kedi görüldüğünde, bu kedi öncekilerden farklı özellikler (daha küçük, daha farklı renkte) taşısa da çocuk o varlığın kedi olduğunu bilerek ortak özelliklerini temel olarak kedi tanımlamasını yapabilir. Makine öğrenimi de bilgisayarların kendisine algoritmalar yoluyla verilen kuralları uygulaması ve büyük veri kümeleri içinden örnekler çıkararak verileri bu kurallara göre sınıflamaları, tanımlamaları ve dolayısıyla öğrenmeleri olarak ifade edilebilir. Bu öğrenmeler sonucunda çıkarımlarda bulunarak geçmiş veri örnekleri yardımıyla gelecekte daha iyi sonuçlar üretme konusunda veri madenciliği uygulamasına katkıda bulunurlar.

Veri madenciliğinde söz konusu diğer bir disiplin olan *görselleştirme*; verilerin, tablolar ve grafikler gibi görseller yardımıyla sunulmasını sağlayan teknolojileri ifade eder. Görselleştirme; verilerin daha kolay anlaşılmasına, analiz edilmesine ve geleceğe yönelik tahminlerde bulunulmasına önemli katkı sağlamaktadır. Veri madenciliğinde kullanılan görselleştirme teknikleri ilk zamanlarda sadece iki boyutlu serpilme ve serpilme matris

çizimleri ya da üç boyutlu grafikler biçimindeydi. Ancak zaman içinde, verilerin öznitelik sayılarındaki artış klasik istatistiğin sunduğu iki veya üç boyutlu grafiklerin yetersiz kalması sonucunu da birlikte getirmiştir. Bu durum da çok daha fazla boyutun görselleştirilmesine imkân sağlayan yeni grafik araçlarının geliştirilmesine neden olmuştur. Yer-Konum veri analizi, sinyal işleme, görüntü analizi gibi teknikler görselleştirme amacıyla kullanılan tekniklere verilebilecek örneklerdir.

Şekil 1.1



Veri madenciliğinin olmazsa olmazlarından biri de veritabanlarıdır. Bilindiği gibi işletmelerde ve yapısal diğer tüm kurumlarda günlük işlemler ve bu işlemlere konu olan veriler kaydedilmektedir. Bununla birlikte *veritabanı* kavramı gelişigüzel veri yiğinları olmayıp birbirine ilişkili olan ve amaca uygun biçimde düzenlenmiş, mantıksal ve fiziksel olarak tanımlanmış veriler bütünüdür. Veritabanı yönetim sistemi ise kısaca veritabanı tanımlamak, veritabanı oluşturmak, veritabanında işlem yapmak, veritabanının farklı kullanıcı yetkilerini belirlemek, veritabanının bakımını ve yedeklemesini yapmak için geliştirilmiş programlar bütündür. Son olarak, veritabanı ve veri tabanı yönetim sisteminin birlikte oluşturduğu bütün de veritabanı sistemi olarak ifade edilir.

Örütü tanımı: Olaylar ve nesneler arasında daha önceden tanımlanmış, düzenli ve sistematiğin içinde tekrar eden ilişkileri bir model olarak kabul eden ve bu modelin (örütünün) benzerlerini ya da en benzerini veritabanı içinden arama ve bulmaya yönelik teknolojidir.

Örütü, olaylar ve nesneler arasında düzenli ve sistematik bir biçimde tekrarlanan ilişkilerini ifade etmek için kullanılan bir kavramdır. **Örütü tanımı** teknolojisi ise daha önceden tanımlanmış, bir model olarak düşünülebilen çok boyutlu bir örütünün veritabanındaki benzerlerini ya da en benzerini arama ve bulma amacıyla yönelik yazılmaları ifade eder. Örütünün konusu yazılı bir metin olabileceği gibi parmak izi, ses, yüz tanıma, kan hücrelerinin karşılaştırılması, el yazılarının belirlenmesi gibi alanlar da olabilir. Verilen son örneklerde örütü, el, yüz, resim, çizim ve ses gibi nesnelerin bilgisayar ortamlarında sayısal olarak ifade edilmesi anlamındadır.

VERİ MADENCİLİĞİ KAVRAMI

İşletme politikalarının ve stratejik kararların temel ögesi veri ve veriden elde edilmiş güvenilir, güncel ve doğru bilgidir. Bu bilgiler, işletme kaynaklarının daha etkin kullanımı ve müşterilere daha iyi ürün ve hizmet sunabilme amaçlarına yönelik olarak kullanılır. Veri madenciliği de ihtiyaç duyulan bu bilgilerin üretilmesinde kullanılan bir araçtır. Veri madenciliği kavramını tanımlamadan önce veri, enformasyon ve bilgi kavramlarını hatırlatmak faydalı olacaktır.

Veri, ham gözlemler, işlenmemiş gerçekler ya da izlenimlerdir. Bu gözlemler, gerçekler ya da izlenimler harf, rakam ya da çeşitli sembol ve işaretler yardımıyla temsil edilir. Bir-

birleriyle ilişkilendirilip yorumlanmadıkları sürece tek başlarına bir anlam ifade etmezler ve bu hâllerde karar verme konusunda da karar vericilere bir katkı sağlayamazlar.

Enformasyon, verinin bir anlam oluşturacak şekilde düzenlenmiş hâlidir. Diğer bir ifadeyle toplanan verilerin birbiriyle çok yönlü olarak ilişkilendirilmesi ve değer yaratıacak bir kaynak niteliğine dönüştürülmesi enformasyonu üretir. Veriden farklı olarak enformasyon tek başına anlamlıdır. Bununla birlikte bu anlam yalnızca konuya ilgili kişi tarafından anlaşılır ve bu kişiye olayları ve nesneleri yorumlamada bir bakış açısı sağlar.

Bilgi ise en yalın tanımıyla verinin işlenmiş ve dönüştürülmüş halidir. Söz konusu işleme ve dönüştürme süreci, veri üzerinde kaydetme, sınıflama, sıralama, hesaplama, özetleme, çoğaltma, analiz ve raporlama işlemlerinin uygulanması ile gerçekleştirilir. Bu işleme sonucunda veri, karar verme sürecine destek olacak şekilde anlam kazanarak bilgiye dönüşmüştür. Dolayısıyla karar vermede etkili olan asıl unsur veriden ziyade bilgidir.

Bilgisayar sistemlerinin hayatımıza bu derece girmesinden önce de veriler kaydedilmekte ve saklanmaktadır. Ancak günümüzden farkı, kayıt ortamı olarak kâğıt, defter, dosya gibi araçların kullanılmasıydı. Teknoloji ve bilgisayar sektöründeki gelişmeler sayesinde dijital kayıt ortamlarına geçilmiş ve buna paralel olarak da daha çok veri kaydedilip saklanabilir duruma gelmiştir. Dijital kayıt ortamlarında yer alan veri hacmindeki büyük artış ise beraberinde yeni sorunları doğarmıştır. Bunlardan biri; çok büyük miktarda veri içinden ihtiyaç duyulan veriye hızlı ve kolay biçimde erişebilmek amacıyla, *verilerin yönetimi sorunu* bir diğer ise verileri işleyip anlamlı bilgilere dönüştürmek amacıyla *yeni yöntem ve yaklaşımlar belirleme zorunluluğudur*. Söz konusu bu sorumlara çözüm bulma isteği bilgisayar teknolojilerindeki gelişmelerin de temel nedenlerinden biri olmuş ve ilkinde çözüm amacıyla veritabanı sistemleri, ikincisine çözüm amacıyla da veri madenciliği yöntemleri geliştirilmiştir. Dikkat edilirse ihtiyaçlar teknolojik gelişmeleri, teknolojik gelişmeler ise yeni ihtiyaçları tetiklemekte ve bu bir döngü biçiminde devam etmektedir.

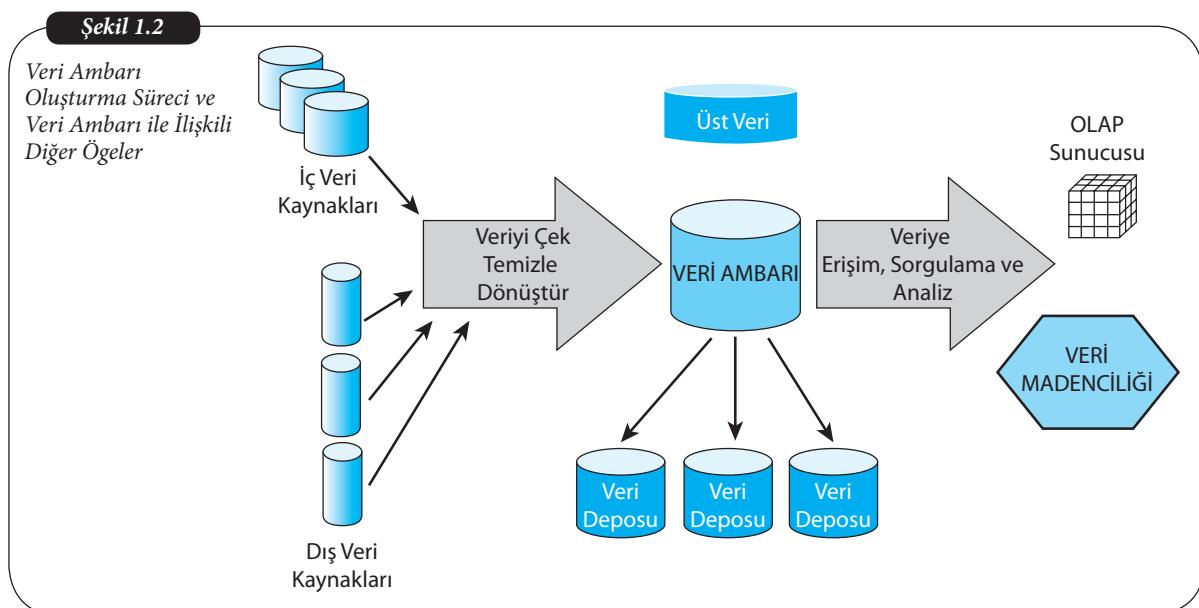
Bilindiği gibi ekonomik yönden değer taşıyan maddelerin (altın, gümüş, elmas, bor, kömür vb.) bulunduğu yerlere maden, bu maddelerin çıkarılıp işlenmesi ile ilgili olarak gerçekleştirilen faaliyetlere de madencilik denir. Bu maddeler bulundukları yerden çıkarılıp işlenmedikleri sürece bir değer taşımazlar. Benzer durum veritabanlarında yoğunlar biçiminde bulunan veriler için de geçerlidir. Veritabanlarında kayıtlı olan veriler de madenlerden çıkarılıp işlenmeye bekleyen değerli maddelere benzetilebilir. Bu nedenle büyük miktarda yoğun veri içinden bilgiye ulaşmak amacıyla kullanılan teknikler bütünü de veri madenciliği adı altında ele alınmaktadır.

Veri madenciliği çalışmaları yapmak için var olması gereken iki temel öğe *veri* ve *veritabanı*dır. Bununla birlikte burada sözü edilen veritabanı, işletmelerin günlük kayıtlarının yer aldığı ve *işlemsel veritabanı* olarak adlandırılan veri tabanları değildir. Daha doğru bir ifadeyle *işlemsel veritabanları* veri madenciliği uygulamalarında doğrudan kullanılmaz. Bu veritabanlarında yer alan veriler birtakım işlemlerden geçirilerek veri madenciliği için kullanılabilir, hazır hâle getirilir. İşte işletmelere ait veritabanlarının, belirli bir amaca göre konu odaklı olarak düzenlenmiş, veri madenciliğinde doğrudan kullanılabilir duruma getirilmiş hâli **veri ambarı** olarak tanımlanır. İşletme çalışanları günlük işlemlerini sürdürmek, farklı yönetim düzeyindeki yöneticiler ise operasyonel, taktik ya da stratejik kararlar verebilmek için farklı veri kaynaklarına ihtiyaç duyarlar. İşletmelerdeki veri kaynakları iç kaynaklar ve dış kaynaklar biçiminde sınıflandırılabilir. İç veri kaynakları; işletmenin kendi iş süreçlerinden elde ettiği verilerin yer aldığı kaynaklardır. Örneğin, üretim bölümü, satın alma ve pazarlama bölümleri kendi günlük işlemlerine ilişkin kayıtları tutarak birer iç veri kaynağı oluştururlar. Bunun dışında işletmelerde günlük faaliyetleri ya da her düzeyde alınacak kararları doğrudan etkileyebilecek dış kaynaklı verilere de ihtiyaç duyulur. Örneğin işletmelerin içinde bulundukları sektördeki veriler, istatistik

Veri ambarı işletmelerde iç veri kaynakları ile dış veri kaynaklarının birleştirilmesi ve düzenlenmesi ile oluşturulmuş, üzerinde veri madenciliği işlemlerinin gerçekleştirileceği veriyi sağlayan daha geniş ve özel veritabanlarına verilen isimdir.

kurumlarının yayınladığı raporlar, yasal düzenlemeler, döviz kuru vb. gibi sermaye piyasasına ilişkin veriler dış kaynaklı verilere verilebilecek örnekler arasındadır. Veri ambarları, söz konusu bu iç ve dış kaynaklı verilerin biraraya getirilmesi ile oluşturulan özel veritabanlarıdır. Bununla birlikte verilerin birleştirilmesi gelişigüzel bir işlem değildir. Veriler farklı kaynaklardan elde edildiği için veriler arasındaki uyumsuzlıkların, tutarsızlıkların giderilmesi ve verilerin amaca uygun, kullanılabilecek biçimde dönüştürülmesi gereklidir (söz konusu bu işlemler izleyen kesiminde *Veriler Üzerinde Ön İşlemlerin Yapılması* başlığı altında anlatılmıştır).

Veri ambarı oluşturulmasına ilişkin süreç ve ilgili olduğu diğer öğeler Şekil 1.2'de verilmiştir.



İngilizce karşılığı meta data olan *üst veri*, veri ambarında yer alan veriler hakkında tanımlamalar olup veri ambarına ilişkin veri kataloğu olarak düşünülebilir.

Farklı kaynaklarda veri martı kavramı ile ifade edilen *veri deposu* (data mart) kavramı ise veri ambarının bir alt kümesi olup işletmenin yalnızca belirli bir bölümünü ya da belirli bir iş sürecini, daha özel bir fonksiyon alanını ilgilendiren parçasıdır. Veri ambarı tüm işletmeyi ilgilendirirken veri deposu tek bir konuya ya da özel bir amaca yönelik verileri içerir.

İşletmeler günlük faaliyetlerine ilişkin basit sorgulamaları ve analizleri işlemsel veritabanları üzerinde kolaylıkla gerçekleştirebilirler. Buna karşın, çok yönlü veri analizi ve sorgulama yapmak istediklerinde normal veri analizi ve sorgulamadan farklı bir sistem kullanırlar. Çevrimiçi Analitik İşleme olarak adlandırılan bu sisteme kısaca **OLAP** (On-Line Analytical Processing) denir. OLAP uygulamaları veri ambarından çekilen veriler üzerinde gerçekleştirilir. OLAP sorgulamaları işlemsel veri tabanlarında gerçekleştirilen basit analiz ve sorgulamalardan farklı olarak, veriyi çok boyutlu biçimde analiz eder ve analiz sonucunda yöneticilere stratejik kararlarında destek olacak yararlı bilgiler sunar. İşletmelerin geleceklerine yönelik önemli kararlarında, karşı karşıya kaldıkları problemler basit yapıda olmayıp çok yönlü, karmaşık, analistik sorgulamalar gerektirecek yapıda ortaya çıkarlar. Bu tür problemlerin çözümü için günlük veri analizi ve sorgulamalar doğal olarak yetersiz kalacaktır.

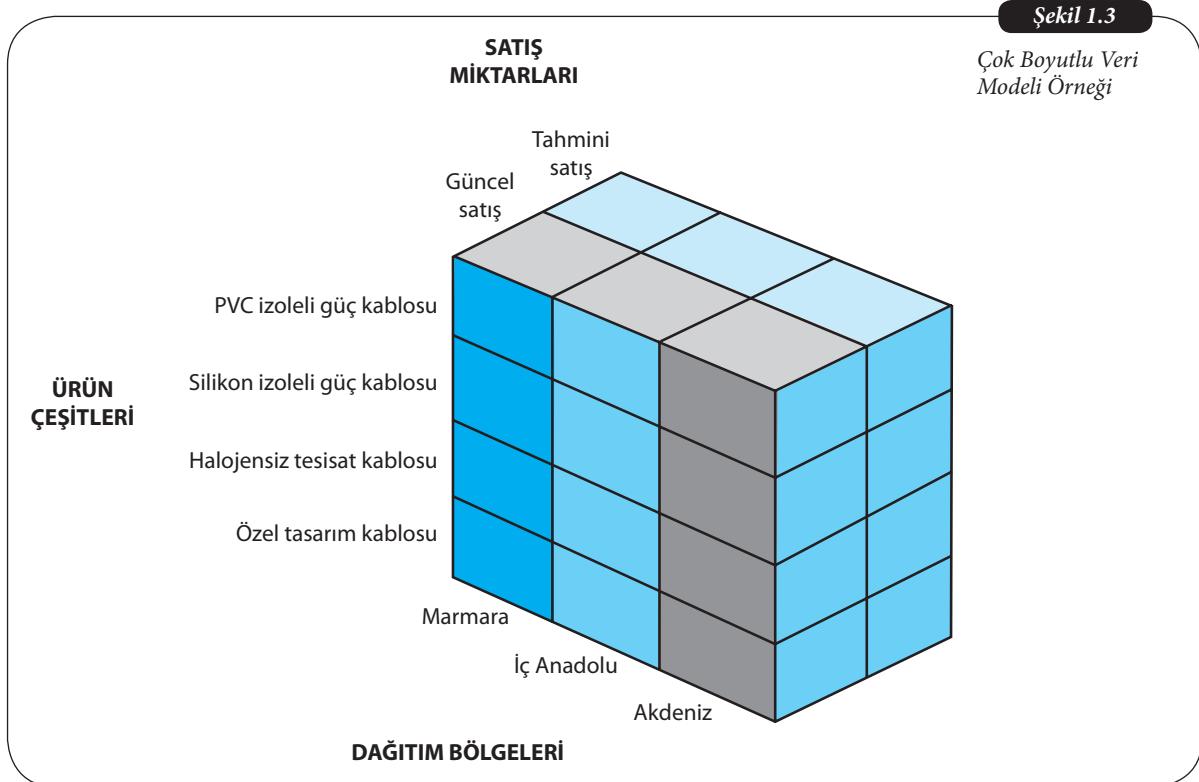
OLAP (Online Analytical Processing - Çevrimiçi Analitik İşleme) veri ambarında yer alan veriler üzerinde çok boyutlu, çok yönlü analiz ve sorgulama yapılmasını sağlayan sistemlerdir.

OLAP işlemini gerçekleştirmek üzere veri ambarı ile etkileşim içinde olan OLAP sunucuları, karmaşık analitik sorguların kısa sürede gerçekleştirilmesine imkân veren çok boyutlu veri modelini kullanırlar.

Örneğin farklı kablo üretimi yapan ve bunları farklı bölgelere dağıtan bir işletmenin ürettiği kablo türleri; PVC izoleli güç kablosu, silikon izoleli güç kablosu, halojensiz tesisat kablosu ve özel tasarım kablosu biçiminde olsun. Önceki ay “halojensiz tesisat kablosundan toplam ne kadar satıldığı” sorusuna cevap almak için, işlemsel veritabanlarında basit sorgulama yapmak yeterli olacaktır. Buna karşın her bir satış bölgesinde ne kadar halojensiz tesisat kablosu satıldığını öğrenmek ve hedeflenen satış miktarı ile gerçekleşen satış miktarı sonuçlarını karşılaştırmak daha karmaşık bir sorgu yapısı olacaktır. İkinci sorgu türüne cevap almak amacıyla OLAP sorgulama işleminin yapılması gerekecektir. Bu örnekte; ürünün ne olduğu, fiyatı, maliyeti, satış bölgesi, satış zamanı vb. verinin farklı boyutlarını temsil eder. Bu nedenle işletme yöneticisi Ağustos ayında İç Anadolu bölgesinde ne kadar halojensiz tesisat kablosu satıldığını, bu miktarın bir önceki ay ile ve geçen yılı Ağustos ayı ile ve satış tahminleriyle karşılaştırmasının sonuçlarını öğrenmek için çok boyutlu veri analizine imkân sağlayan OLAP sistemini kullanabilecektir. OLAP, verdiğimiz örnekten çok daha karmaşık veri analizi ve sorgulamalarına da çok hızlı bir sürede cevap verebilme olanağı sağlamaktadır. Şekil 1.3 yukarıda verdiğiz örneğe ilişkin olarak ürün çeşitlerini, dağıtım bölgelerini, güncel satış miktarlarını ve tahmini satış miktarlarını temsil etmek için oluşturulan çok boyutlu veri modeline örnektir.

Şekil 1.3

Çok Boyutlu Veri Modeli Örneği



Buraya kadar anlatılanlardan da görüldüğü üzere, veri analizi ve sorgulama konusunda en basit işlem; işletmelerin işlemsel veritabanlarında yaptıkları basit sorgulamalardır. Daha karmaşık ikinci düzey sorgulamalar OLAP işlemleri ile gerçekleştirilir. OLAP ile elde edilemeyecek karmaşılıktaki sorgulamalar ve sonuçlara ulaşmak ise veri madenciliğinin konusudur. İşlemsel veritabanlarından yapılan sorgulamalar da OLAP ile yapılan

sorgulamalar da var olan, bilinen, tahmin edilebilecek sonuçları ortaya koymaktadır. Veri madenciliği ise daha önceden bilinmeyen, tahmin bile edilemeyen bilgileri açığa çıkarmayı amaçlar.

Veri madenciliği kavramı için çeşitli tanımlar yapılmıştır. Bu tanımlardan bir kısmı aşağıda verildiği gibidir:

Veri madenciliği, büyük miktardaki veri yiğinları üzerinde analiz yaparak veriler arasında var olan ve geleceğin tahmin edilmesine yardımcı olacak anlamlı ve yararlı ilişki ve kuralların bilgisayar yazılımları aracılığıyla aranması faaliyetleridir. Bu anlamda veri madenciliği, çok büyük miktardaki veriler arasındaki bağlantıları inceleyerek aralarındaki ilişkisi ortaya çikaran ve veritabanları içinde açıkça fark edilemeyen, gizli kalmış yararlı bilgilerin açığa çıkarılmasını sağlayan veri analizi tekniğidir.

Veri madenciliği, çeşitli analiz araçlarını kullanarak veriler arasındaki örüntü (desen) ve ilişkileri keşfeder, bunları doğru tahminler yapmak için kullanan bir süreçtir. Veri madenciliğinin amacı, geçmiş faaliyetleri analiz ederek bu analizleri geleceğe yönelik tahminlerde temel almak ve karar vermeye destek olacak modeller oluşturmada kullanmaktadır. Buna göre veri madenciliği, büyük miktarda veri içinden, gizli kalmış, değeri olan, kullanılabilir bilgileri açığa çıkarmak ve bu bilgileri özellikle stratejik kararlarda destek sağlayacak biçimde elde etmek amacıyla kullanılmaktadır.

Veri madenciliği, veri analizi için, gelişmiş ve karmaşık araçlar kullanarak yoğun veri kümeleri içinden daha önceden bilinmeyen olgu ve olayları keşfetmek ve veriler arasında mantıklı ilişkileri ve kalıpları ortaya çıkarmak amacıyla yapılan çalışmalardır. Burada vurgulanması gereken önemli nokta, veri madenciliği ile elde edilecek bilginin daha önceden bilinmeyen yeni keşfedilen olmasıdır. Önceden bilinmeyen bilgi, önceden tahmin bile edilemeyen bilgi anlamındadır. Bu anlamda veri madenciliği, tahmin edilen ya da farklı teknikler yardımıyla daha önceden ulaşılmış sonuçların doğruluğunu ispatlamak amacıyla kullanılan bir araç değildir. Diğer tekniklerden temel farkı, daha önce düşünülmemiş hiç akla gelmemiş sonuçları ortaya çıkarmasıdır.

Veri madenciliği, istatistiksel ve matematiksel tekniklerle birlikte örüntü tanım teknolojilerini kullanarak çeşitli depolama ortamlarında kayıtlı bulunan veri yiğinları üzerinde gerçekleştirilen elemeler sonucunda anlamlı yeni korelasyon, örüntü ve eğilimlerin keşfedilmesi sürecidir.

Yığın veri içinden anlamlı ilişkiler çıkarma ve yararlı bilgilere dönüştürme işlemine zaman içerisinde; bilgi çıkarımı, enformasyon keşfi, enformasyon hasadı, veri arkeolojisi, veri örüntü işleme, veri şablon işleme gibi farklı isimler verilmiştir. Aynı anlamda kullanılan *veri madenciliği* terimi ise çoğunlukla bilgisayar mühendisleri, istatistikçiler, veri analistleri ve yönetim bilgi sistemleri toplulukları tarafından tercih edilmektedir.

Burada belirtirmesi gereken diğer bir nokta, *veri madenciliği* kavramı ile *veritabanlarında bilgi keşfi* kavramının zaman zaman aynı anlamda kullanıldığıdır. Ancak bu doğru bir kullanım değildir. Çünkü veri madenciliği, veritabanlarında bilgi keşfi sürecinin yalnızca bir adımıdır.

SIRA SİZDE

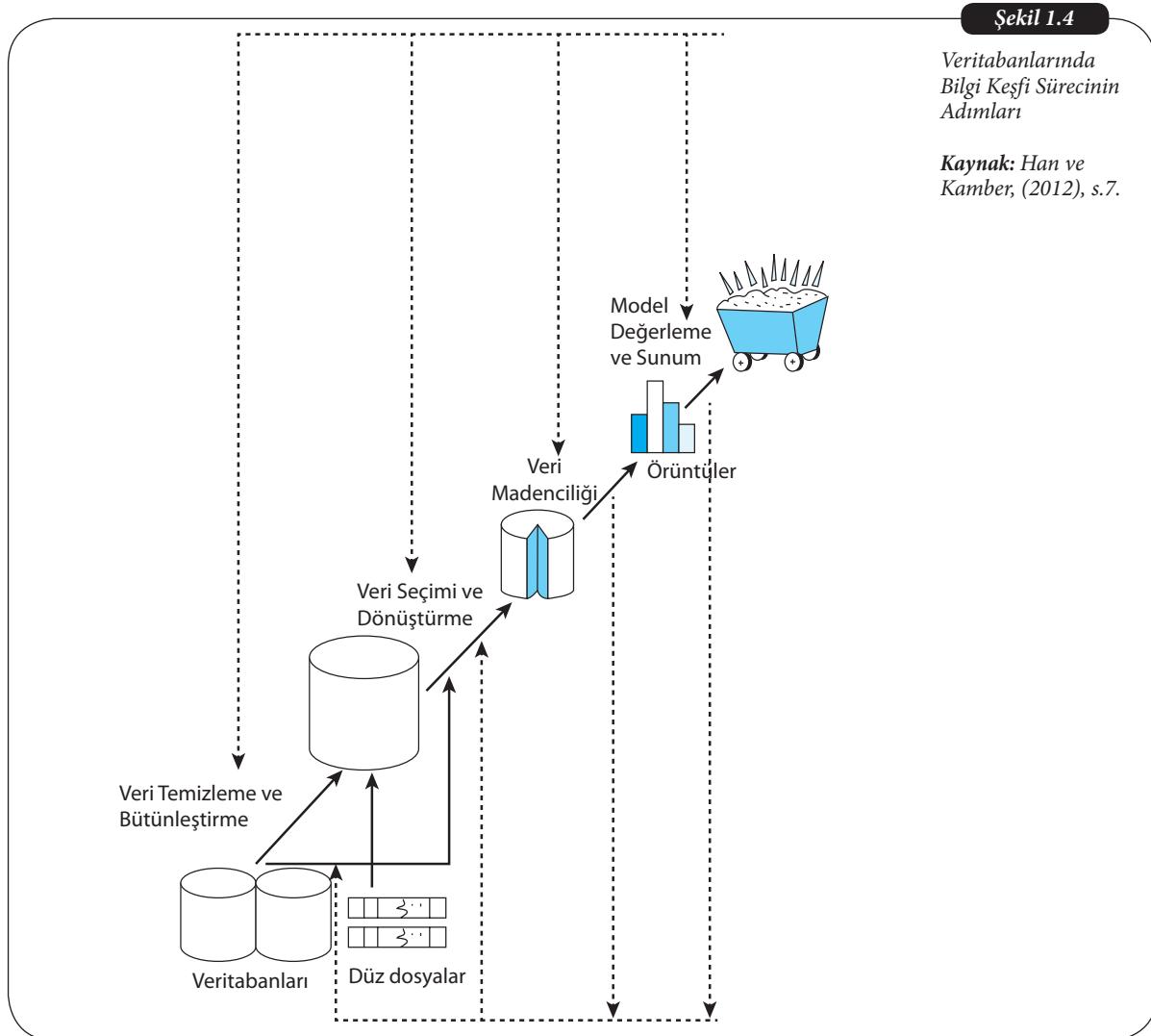


Veri madenciliğinde, üzerinde sorgulama ve analiz yapılan verilerin yer aldığı veritabanı nasıl adlandırılır, özellikleri nelerdir?

VERİTABANLARINDA BİLGİ KEŞFİ SÜRECI

Veritabanlarında bilgi keşfi ifadesi ilk kez 1989 yılında “Veritabanlarında Bilgi Keşfi Çalışma Toplantısı”nda ortaya atılmıştır. Bu toplantıda; *bilginin*, veri keşfi sürecinin sonunda elde edilen ürün olduğu vurgulanmıştır.

Veritabanlarında Bilgi Keşfi, veriden faydalı bilginin keşfedilmesi sürecinin tamamıdır. Veri madenciliği ise bu sürecin bir adım olup veriden örüntülerin belirlenmesi ve aktarımı için özel algoritmaların uygulanması işlemlerine karşılık gelmektedir. Veritabanlarında Bilgi Keşfi sürecinin adımları Şekil 1.4'teki gibi gösterilebilir.



Veritabanlarında Bilgi Keşfi sürecinde, işlemel veritabanlarında depollanmış olan verinin sorgulama ve analiz için uygun hâle getirilmesi işlemleri yürütülür. Veritabanlarında Bilgi Keşfi sürecinde izlenmesi gereken temel aşamalar aşağıdaki gibi sıralanabilir.

1. Amacın Tanımlanması
2. Veriler Üzerinde Ön İşlemlerin Yapılması
3. Modelin Kurulması ve Değerlendirilmesi
4. Modelin Kullanılması ve Yorumlanması
5. Modelin İzlenmesi

Sıralanan bu aşamalara bütünsel olarak bakıldığından, veri madenciliği sürecinde;

- Veri madenciliği öncesindeki işlemler,
- Veri madenciliği işlemleri,
- Veri madenciliği sonrasında işlemler

biriminde bir uygulamanın söz konusu olduğu görülebilir.

Veri madenciliği öncesindeki işlemler; veri tabanlarında bilgi keşfi sürecinin ilk iki aşaması olan, amacın tanımlanması ve veriler üzerinde ön işlemlerin yapılması aşamalarına karşılık gelmektedir.

Veri madenciliği işlemlerinin kendisi, modelin kurulması ve değerlendirilmesi aşamasında gerçekleştirilen faaliyetlerdir.

Veri madenciliği sonrasındaki işlemler ise modelin kullanılması ve yorumlanması ile modelin izlenmesi aşamalarındaki işlemlerdir.

Ünitenin izleyen kesiminde söz konusu bu aşamalara ilişkin bilgiler verilmiştir.

Amacın Tanımlanması

Bu aşamada, işletmenin ya da kurumun veri madenciliğini hangi amaca yönelik olarak gerçekleştirmek istediği belirlenir. Söz konusu amaç bir problemi ortadan kaldırılmaya odaklanmış ve açık bir biçimde ifade edilmiş olmalıdır. Buna ek olarak, elde edilecek sonuçların başarı düzeylerinin nasıl ölçüleceği de tanımlanmalıdır. Bu aşamada ayrıca, süreç sonunda yapılacak değerlendirme ve öngörülerin yanlış olması durumunda katlanılaç maliyetlere ve doğru olması durumunda elde edilecek kazanımlara ilişkin tahminlere de yer verilmelidir.

Veriler Üzerinde Ön İşlemlerin Yapılması

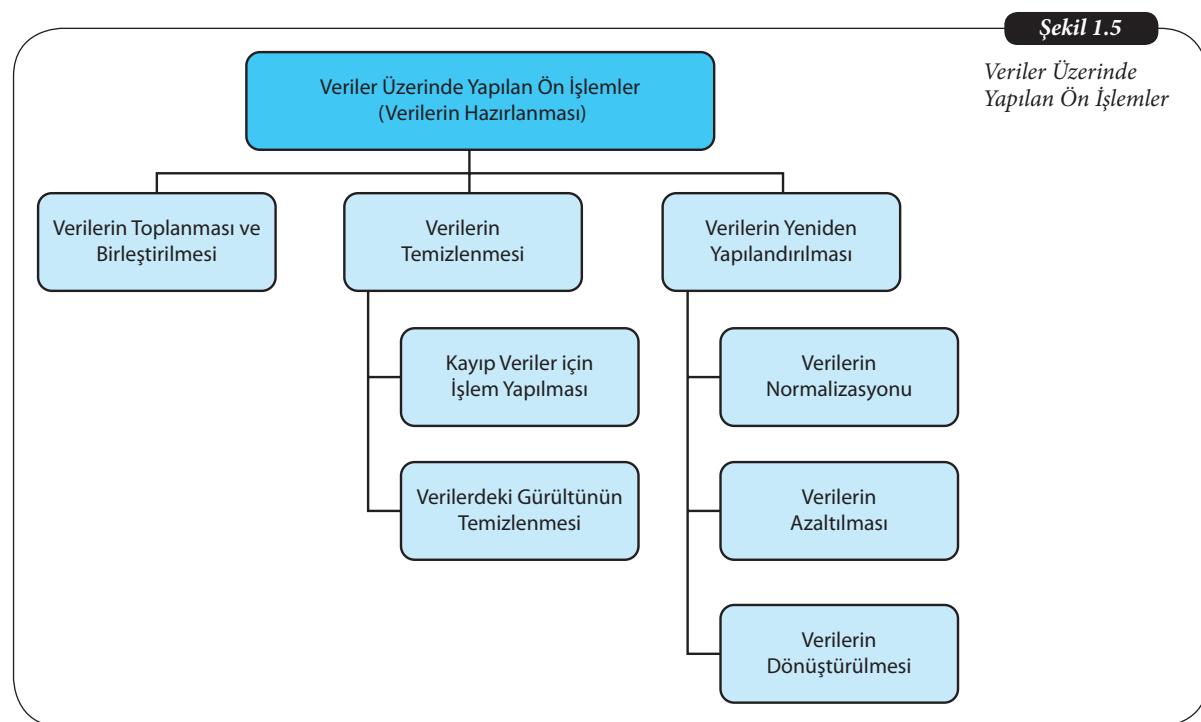
Veriler üzerinde ön işlemler yapılması, verilerin veri madenciliği için hazırlanması anlamadadır. Veri madenciliği ile ulaşılması hedeflenen sonuçların kalitesi veritabanlarında yer alan verinin kalitesi ile yakından ilişkilidir. Bu nedenle veri madenciliği işlemleri öncesinde verilerin analize hazır hale getirilmesi oldukça önemli bir aşamadır. Buna bağlı olarak veriler üzerinde yapılan ön işlemler, veri tabanlarında bilgi keşfi sürecinin en fazla zaman alan aşamasıdır. Bir sonraki aşama olan modelin kurulması aşamasında herhangi bir sorunun ortaya çıkmasının, veri üzerindeki ön işlemlerin ne kadar titizlikle yapıldığına bağlıdır. Ön işlemler aşamasında yeterli özenin gösterilmemesi, model kurma aşamasından ön işlemler aşamasına tekrar geri dönülmesine ve verinin yeniden düzenlenmesine neden olacaktır. Bu durum, ön işlemlerle verinin hazırlanması ve modelin kurulması aşamaları için harcanan enerji ve zamanın, bilgi keşfi sürecinin toplamı içinde büyük bir paya sahip olmasına neden olacaktır.

Veriler üzerindeki ön işlemler genel olarak;

- Verilerin toplanması ve birleştirilmesi,
- Verilerin temizlenmesi,
- Verilerin yeniden yapılandırılması

birimde sınıflandırılabilir.

Verilerin temizlenmesi aşamasında yapılan işlemler kendi içinde; kayıp veriler için yapılan işlemler ve gürültülü veriler için yapılan işlemler olarak ikiye ayrılabilir. Verilerin yeniden yapılandırılması ise kendi içinde; verilerin normalizasyonu, verilerin azaltılması ve verilerin dönüştürülmesi biçiminde sınıflandırılabilir. Bu sınıflandırmalar Şekil 1.5'te bir araya getirilmiştir.



Verilerin Toplanması ve Birleştirilmesi

Verilerin veri madenciliğine hazırlanabilmesi için yapılması gereken ilk şey doğal olarak verilerin belirlenmesidir. Bu yapılrken öncelikle tanımlanan amaca ve probleme uygun verilerin neler olduğu ve bu verilerin hangi kaynaklarda yer aldığı araştırılır. Bu belirleme sonrası veriler bulundukları farklı kaynaklardan toplanır ve birleştirilir. Gerekli verilerin toplanmasında öncelikli olarak kurumun kendi veritabanı ve veri kaynaklarından yararlanılır. Daha önceden de belirtildiği üzere bu tür veriler iç kaynaklı verilerdir. Bunun yanı sıra, istatistiksel bilgiler, finansal raporlar, menkul kıymet değerleri gibi bilgilerin yer aldığı kamuya ait kurumsal veri tabanlarından veya veri pazarlayan farklı kuruluşların veri tabanlarından da yararlanılabilir. Örneklenen veri kaynakları ise dış veri kaynaklarıdır. Verilerin hangi kaynaklardan, hangi koşullar altında ve hangi yöntemlerle toplandığı önemlidir.

Verilerin Temizlenmesi

Veritabanlarından alınan kayıtların bir kısmında, diğer kayıtlarda var olan bazı veriler eksik olabilir. Örneğin, işletme çalışanlarına ait kişisel bilgilerin tutulduğu kayıtlarda çoğunuğun doğum tarihi yer alırken bazı çalışanlara ait kayıtlarda doğum tarihi verisi eksik olabilir. Müşteriler arasında yapılan bir araştırmada yaşını, kilosunu ya da gelirini belirtmek istemeyen müşteriler olabilir. Veritabanı kayıtları içindeki böylesi eksik veriler **kayıp veri** olarak adlandırılır. Bunun yanı sıra kayıtlarda yer alan bir kısım veriler doğru olamayacak kadar uç değerlerde, dolayısıyla yanlış girilmiş olabilir. Örneğin, doğum tarihi 1974 olan bir kişi için bu değer 1074 olarak kaydedilmiş olabilir. 1074 değeri gibi aşırı uç değerler **aykırı değer**, bu şekildeki uç verilerin geneli de **gürültülü veri** olarak nitelendirilir. Bunların dışında kayıtlarda yer alan bazı veriler (örneğin, olmayan bir ürün adı ya da stok numarası gibi) tamamiyla yanlış ya da anlamsız olabilir. Verilerin temizlenmesi aşamasında dikkat edilmesi gereken diğer bir konu veriler arasındaki uyumsuzluktur. Veri madenciliğinde kullanılacak verilerin farklı kaynaklardan toplanması, doğal olarak veri

Kayıp veri, veritabanlarındaki kayıtlarda eksik olan verilerdir. Kayıp veriler çeşitli nedenlerden kaynaklanabilir; veri toplamada yanlış araçların kullanılması, veri girişinde hata yapılması ya da veri toplama aşamasında sorulara eksik cevap verilmesi bunedenlerden bazılıdır.

Veritabanlarında doğru olmayacak kadar uç değerler, aykırı değer ya da sıra dışı değer olarak tanımlanır. Bu şekildeki aykırı değerler ya da farklı sebeplerle yanlış girilmiş değerler genel olarak **gürültülü veri** olarak tanımlanır.

uyumsuzluklarına neden olabilecektir. Bu uyumsuzluklardan başlıcaları, verilerin farklı zaman dilimlerine ait olmaları, farklı ölçü birimleriyle temsil edilmeleri ya da farklı biçimlerde kodlanmalarıdır. Örneğin; bir veri tabanında cinsiyet özelliğinin E/K biçiminde, bir diğer veritabanında ise 0/1 biçiminde kodlanması veriler arasında uyumsuzluğa neden olacaktır. Dolayısıyla, toplanan verilerin birbirleriyle ne ölçüde uyumlu oldukları bu adımda incelenerek değerlendirilmeli, uyumsuz veriler belirlenerek gerekli düzenlemeler yapılmalıdır.

Verilerin temizlenmesi, kayıp ya da eksik değerleri tamamlamak, aykırı değerleri belirleyerek gürültüyü ortadan kaldırmak ve verilerdeki tutarsızlıkları, uyumsuzlukları gidermek için kullanılan birçok yaklaşımı ve tekniği kapsar. İzleyen kesiminde bu yaklaşımlardan bir kısmı hakkında bilgi verilmiştir.

Kayıp verilerin neden olduğu olumsuzlukları ortadan kaldırmak amacıyla kullanılan yaklaşımalar:

- a. *Kayıp veri içeren kaydı veri kümesinden çıkarmak:* Bu yaklaşım, kayıp veri içeren kayıt sayısının toplam kayıt sayısı içinde çok küçük bir orana karşılık gelmesi ve kayıp verilerin sonuçlara önemsenmeyecek bir etki yapması durumunda kullanılabilecek bir yaklaşımdır. Kayıp veri içeren kayıt sayısının çok olması durumunda, sonuçları olumsuz etkileyeceğinden bu yaklaşım önerilmez.
- b. *Kayıp verileri tek tek yazmak:* Veri kümesi küçük, kayıp verilere ulaşmak mümkün, yeterince zaman mevcut ve kayıp verilere mutlaka ihtiyaç duyuluyorsa bu yaklaşım kullanılır. Söz konusu bu durumların dışında bu yaklaşımı kullanmak gereksiz zaman kaybına neden olacaktır.
- c. *Kayıp verilerin hepsi aynı veriyi girmek:* Örneğin, yapılan bir hane halkı araştırmasında bazı bireyler gelirlerini belirtmekten kaçınmış olabilir. Ya da bir işletmenin müşterileri arasında yaptığı bir araştırmada bazı müşteriler doğum tarihi bilgisini yazmamış olabilir. Bu durumda gelir bilgisinin bulunmadığı tüm kayıtlar için, yok anlamında Y harfi, doğum tarihinin eksik olduğu tüm kayıtlar içinse, eksik anlamında E harfi kayıp veri yerine yazılabilir. Kayıp verilerin bu yaklaşımı giderilmesi farklı sonuçlar verebilir. Gelir bilgisinin Y olması ya da doğum tarihinin E olması belirleyici ya da ayırt edici bir özellikmiş gibi görünebilir. Diğer bir ifadeyle bu veriler kullanılan veri madenciliği algoritmasını yanıltabilir. Bunun tersine bu yaklaşım bazı durumlarda veri madenciliğinin gerçek amacına hizmet ederek gizli bilgilerin keşfedilmesini de sağlayabilir. Örneğin, doğum tarihi girmemiş olan müşterilerin, en çok para harcadıkları ürünlerin yaşlanma karşıtı ürünler olduğu sonucu elde edilebilir. Kayıp verilerin aynı veri değeri ile temsil edilmesinde kayıp veriyi temsil etmek üzere sayısal bir değer de atanabilir. Örneğin, 9 rakamı veri girişi yapılmadığı anlamında kullanılabilir. Veri madenciliği algoritması bu değeri göz ardı ederek yok sayabilir ve analiz buna göre gerçekleştirilebilir. Burada dikkat edilmesi gereken bir nokta, ister sayısal ister alfabetik kodlama yapılsın kayıp verileri temsil için seçilen değerin, analizde anlamlı olacak (diğer verileri temsil eden) başka bir değere karşılık gelmediğinden emin olunmasıdır.
- d. *Kayıp veri yerine tüm verilerin ortalama değerinin girilmesi:* Örnek olarak ücret verisi eksik olan kayıtlar için, diğer kayıtlarda yer alan ücret verilerinin ortalaması yazılabılır. Ortalama değeri bulunurken tüm verilerin ortalaması yerine belirlenen bir sınıfın ortalamasının alınması daha uygun olacaktır. Daha açıklayıcı olması açısından Tablo 1.2'de görülen bir muhasebe bürosu çalışanları kira bilgileri örnek verisi üzerinden devam edelim.

Sıra No	Çalışma Süresi	Kira Miktarı	Mahalle
1	5	850	Yenibağlar
2	6	675	Emek
3	3	780	Alanönü
4	9	950	Alanönü
5	14	1250	Batıkent
6	9		Alanönü

Tablo 1.2
Örnek Veri Kümesi

Tablo 1.2'de 6 sıra numarasına sahip Alanönü mahallesinde ikamet eden 9 yıllık çalışanın ödediği kira miktarı bilgisi elde edilememiş olsun. Bu kayıp veri için doğrudan aritmetik ortalama değeri bulunacak olursa verileri girilmiş olan diğer beş kişinin kira miktarlarını ortalaması hesaplanır. Bu değer, veri için 901 olarak hesaplanabilir. Bu yaklaşım yerine verileri kendi arasında sınıflayıp, sınıf ortalaması alma yoluna da gidilebilir. Bu durumda önce uygun sınıfın belirlenmesi gerekecektir. Sınıf olarak Alanönü mahallesinde ikamet eden çalışanlar seçilirse bu grupta yer alan iki çalışanın kira ortalama değeri 865 olacaktır. Hangi değerin kayıp veri için kullanılacağı karar verici tarafından tespit edilecektir.

e. *Kayıtlarda yer alan diğer değişkenler yardımıyla kayıp verilerin tahmin edilmesi:*

Veri kümesinde yer alan ve eksik olmayan kayıtlardaki veriler kullanılarak kayıp veriler tahmin edilebilir. Kayıp verilerin tahmininde, regresyon analizi, zaman serileri analizi, Bayesien sınıflandırma, karar ağaçları, maksimum bekleneni vb. biçimde sıralanan teknik ya da yöntemler kullanılabilir.

Verilerdeki gürültünün temizlenmesi amacıyla kullanılan yaklaşımlar ise aşağıdaki gibidir:

- a. *Bölümleme yöntemiyle gürültünün temizlenmesi:* Bu yöntemde üzerinde analiz yapılacak veriler önce küçükten büyüğe doğru sıralanır. Daha sonra veriler eşit sayıda eleman içeren gruplara bölünür. Her grupta bulunan verilerin ortalama değeri ya da medyan değeri bulunarak grupta yer alan tüm veriler ortalama ya da medyan değeri ile değiştirilerek düzeltme yapılır. Örneğin; 24, 18, 7, 27, 31, 24, 11, 37, 28 biçimindeki bir veri seti üzerinde bölümleme yöntemiyle düzeltme yapmak istenirse, öncelikle verilerin küçükten büyüğe doğru sıralanması gereklidir. Sıralı veri 7, 11, 18, 24, 24, 27, 28, 31, 37 olacaktır.

Veri eşit sayıda birim içerecek biçimde gruplara bölündüğünde izleyen yapı oluşacaktır.

Bölüm 1: 7 11 18

Bölüm 2: 24 24 27

Bölüm 3: 28 31 37

Bu aşamada ilgili bölümlendirmeye göre her grup ortalaması tespit edilerek birimlerin gerçek değerleri yerine kullanılabilir. Bu durumda izleyen veri yapısı ortaya çıkacaktır.

Bölüm 1: 12 12 12

Bölüm 2: 25 25 25

Bölüm 3: 32 32 32

Buradan görüldüğü üzere her grupta yer alan orijinal değerler o grubun ortalama değeri ile değiştirilerek düzeltme sağlanmıştır (Verilerin düzeltilmesi amacıyla ortalama değeri yerine medyan değerinin kullanımını da tercih edilebilirdi).

- b. *Sınır değerleri kullanılarak gürültünün temizlenmesi:* Bu yöntemde de veriler önceki yöntemde olduğu gibi küçükten büyüğe doğru sıralanarak eşit bölmelere ayrılır. Daha sonra, her bölümün en küçük ve en büyük değerli verileri sınır değerleri olmak üzere bölüm içindeki her bir değer üst sınır ya da alt sınır değerlerinden hangisine yakınsa o sınır değeri ile değiştirilir. Bu durumu bir önceki örnekteki verileri kullanarak gösterirsek;

Bölüm 1 :	7	11	18
Bölüm 2 :	24	24	27
Bölüm 3 :	28	31	37

Sınır değerleri ile düzeltme yapıldıktan sonra veriler;

Bölüm 1:	7	7	18
Bölüm 2:	24	24	27
Bölüm 3:	28	28	37

biçiminde olur.

Bu yöntemde kullanılabilen diğer bir yaklaşımın, veri kümesi içindeki en büyük veri ile en küçük veri değerlerinin birbirinden farkının, kümedeki eleman sayısına bölünmesiyle elde edilen değerin o küme elemanlarına atanmasıdır. Buna göre örneğimizde;

$$\begin{aligned} \text{Bölüm 1: } & 7 \quad 11 \quad 18 = (18-7)/3 = 3,67 \\ \text{Bölüm 2: } & 24 \quad 24 \quad 27 = (27-24)/3 = 1 \\ \text{Bölüm 3: } & 28 \quad 31 \quad 37 = (37-28)/3 = 3 \end{aligned}$$

olacaktır. Bunun sonucunda her bölümdeki değer hesaplama sonucu bulunan değerler değiştirilerek;

Bölüm 1:	3,67	3,67	3,67
Bölüm 2:	1	1	1
Bölüm 3:	3	3	3

biçiminde düzeltilmiş olur.

- c. *Kümeleme yöntemiyle düzeltme yapılması ve gürültünün temizlenmesi:* Bu yaklaşım aykırı değerlerin ortaya çıkarılması ve düzeltmesinde kullanılır. Buna göre, veri setinde yer alan veriler birbirlerine olan benzerlik ve yakınlıklarına göre kümeleme yapılır. Bu kümeleme işlemi sırasında üç değer olarak kabul edilen bazı veriler hiçbir küme içinde yer alamayacaktır. Bu şekilde belirlenen her bir aykırı değere, en yakın olduğu kümenin ortalama değeri veya en küçük ya da en büyük değeri atanarak aykırı veriler temizlenmiş olur.
- d. *Regresyon yöntemiyle düzeltme yapılması ve gürültünün temizlenmesi:* Verilerde gürültünün temizlenmesi amacıyla kullanılabilen diğer bir yöntem, değişken değerlerini bir fonksiyon yardımıyla ilişkilendiren regresyon yönteminin kullanılmasıdır. Doğrusal regresyon iki nitelik ya da iki değişken arasındaki en uygun doğruya bulmayı içerir. Bu nedenle bir nitelik (ya da değer) diğerinin tahmin edilmesinde kullanılabilir. Çoklu doğrusal regresyon doğrusal regresyonun genişletilmiş biçimi olup ikiden fazla nitelik (değişken) söz konusu olduğunda kullanılır ve analiz çok boyutlu düzlemde gerçekleştirilir.

Verilerin Yeniden Yapılandırılması

Veri madenciliği amacıyla kullanılan model, teknik ve algoritmalar belirli yapılardaki veriler üzerinde uygulanabilir. Örneğin, bir kısım algoritmalar yalnızca sayısal değerler üzerinde çalışırken bir kısım algoritmalar da yalnızca kategorik değerler üzerinde çalışır. Bunun dışında bazı algoritmalar ise yalnızca 0 ve 1'lerle temsil edilen veriler üzerinde çalışır. Bu nedenle eldeki verilerin kullanılacak algoritmeye uygun hâle getirilmesi, diğer bir ifadeyle yeniden yapılandırılması gereklidir. Bu amaçla gerçekleştirilen işlemler; verilerin normalizasyonu, verilerin azaltılması ve verilerin dönüştürülmesi başlıklarını altında incelenebilir.

- a. *Verilerin normalizasyonu:* Farklı değerlerdeki verilerin 0,0-1,0 gibi aralıklardaki değerlerle temsil edilmesi işlemine normalizasyon denir. Normalizasyon işlemi için kullanılabilen yöntemlerden bir kısmı; min-maks normalizasyonu, sıfır-ortalamalı normalizasyonu ve ondalıklı normalizasyon biçiminde sıralanabilir.
- b. *Verilerin azaltılması:* Bellek kapasitelerinin artmış olması ve bilgisayar sistemlerinin ucuzlaması sonucunda veri tabanlarında gerekli olsun ya da olmasın çok miktarda veri tutulmaktadır. Bu aşırılık veri ön işlemleri aşamasında veri analizi çalışmalarını zorlaştırmaktadır. Bu nedenle veriler yapılandırılırken gerçekleştirilen bir diğer işlem de verilerin temel özelliklerini kaybetmeden miktar olarak azaltılmasıdır. Verilerin azaltılması, veri kümesi içinde gereksiz olduğu düşünülen verinin kaldırılması biçiminde olabileceği gibi daha çok birden fazla değişkenin birleştirilerek tek bir değişkenle ifade edilmesi biçiminde gerçekleştirilir. Bu işleme veri indirgeme işlemi de denilmektedir. Verilerin azaltılması amacıyla geliştirilen çeşitli yöntemler bulunmaktadır. Bu yöntemlerden bazıları; boyut sayısını azaltma, veri sıkıştırma, temel bileşenler analizi, faktör analizi biçiminde sıralanabilir.
- c. *Verilerin dönüştürülmesi:* Bu aşama, analize konu olan veri kümesinin gerekli veriyi içermesi ancak verinin kullanılan algoritmeye uygun yapıda olmaması durumunda gerçekleştirilir. Verilerin gösterim biçimini kullanılan algoritmanın etkinliği üzerinde çok önemli bir paya sahiptir. Buna göre verilerin dönüştürülmesi, algoritmada doğrudan kullanılabilecek biçimde verinin kendi içinde yeniden düzenlenmesini ifade etmektedir. Örnek olarak günlük işlemlerin kaydedildiği işlemsel veritabanlarındaki verilerin büyük çoğunluğu sayısal veriler olup sürekli değerler alır. Veri madenciliğinde kullanılan bazı algoritmalar her bir veriyi ayrı bir değişken olarak ele aldığından buna göre işlem yapar. Bu durumu açıklamak üzere, bir işletmede çalışanların maaşlarının 1.500 TL ile 5.000 TL arasında değiştiğini varsayıyalım. Bu durumda bazı veri madenciliği algoritmaları, söz konusu alt limit ve üst limit değerlerini ve bu iki değer arasındaki tüm değerleri ayrı değişkenler olarak ele alacaktır. Bunun sonucu olarak veriler üzerindeki işlem süresi artacak, elde edilecek sonuçlar gereğinden ayrıntılı ve uzun olacaktır. Bu nedenle sürekli nitelikteki bu tür verilerin kesikli ve kategorik veri biçimine dönüştürülmesi gereklidir. Bu amaçla, belirlenen farklı değerler arasındaki maaşlar, düşük, orta, yüksek biçiminde kategorize edilebilir.

Modelin Kurulması ve Değerlendirilmesi

Bu aşama, veri madenciliği modelinin kurulduğu ve geçerli bir model olmadığının değerlendirildiği aşamadır. Tanımlanan amaca ulaşmada kullanılacak en uygun modelin belirlenmesi için, çok sayıda modelin denenmesi gerekebilir. Bu nedenle, veri hazırlama ve model kurma aşamaları, en iyi olduğu düşünülen modele ulaşılınca kadar tekrarlanır. Kurulan modelin geçerli bir model olup olmadığı da çeşitli açılardan sınanmalıdır. Yanlış model kurulması ya da modelde kullanılan verilerin tutarsız, eksik ya da sıra dışı değerlerden oluşması modelin geçerliliğini etkileyen önemli nedenlerdir.

Modelin Kullanılması ve Yorumlanması

Bir önceki aşamada kurulan ve geçerliliği sınanarak uygulanmak üzere kabul edilen modelin kullanıldığı aşamadır. Kurulan modellerden elde edilen sonuçlar değerlendirilerek başlangıçta belirlenen amaca ulaşılıp ulaşılmadığı yorumlanmalıdır. Probleme çözüm getirmediği düşünülyorsa süreç yenilenmelidir.

Modelin İzlenmesi

Ne kadar doğru ve iyi bir model kurulmuş olsa da sistem zaman içinde ortaya çıkacak değişimlerden etkilenebilecektir. Bu nedenle model kullanılmaya başlandıktan sonra, sistemin ne kadar iyi çalışığının sürekli olarak izlenmesi ve ölçülmesi bir gereklilikdir. İzleme ve ölçme sonuçlarına göre gerekiyorsa modelde değişiklik ve düzenlemelerin yapılması bu aşamanın konusudur.

SIRA SİZDE

2

Veri tabanlarında bilgi keşfi sürecinde veriler üzerinde yapılan ön işlemler nelerdir?

VERİ MADENCİLİĞİNDE KULLANILAN MODELLER

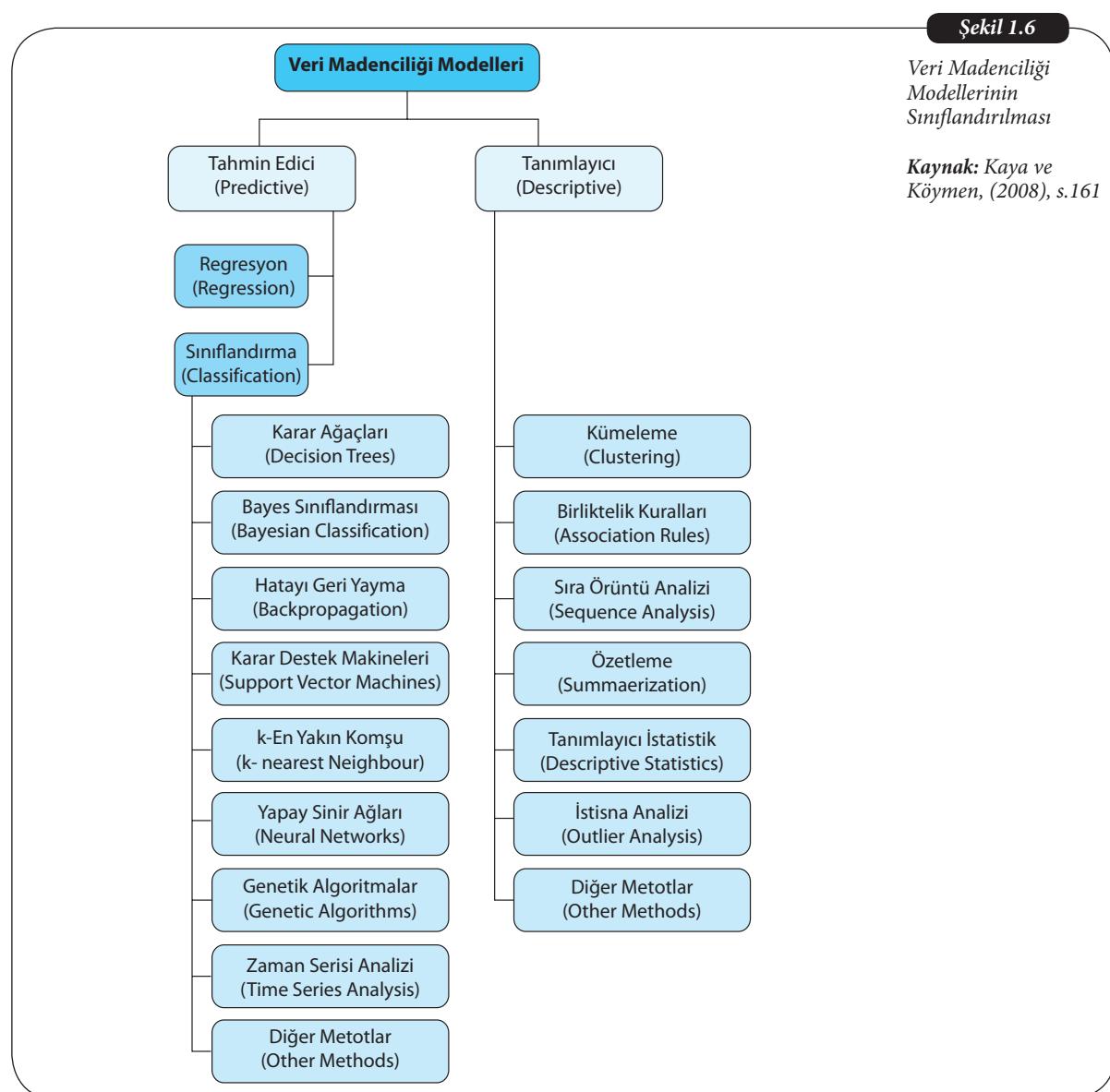
Veri madenciliği büyük hacimli verilerin işlenmesi için geliştirilmiş algoritmalar ile geneliksel veri analiz yöntemlerinin karmaşıklığı olan bir teknolojidir. Veri madenciliği farklı görevleri yerine getirmek amacıyla pek çok farklı algoritmayı kullanır. Aydin (2007) "Algoritmalar veriyi inceler ve incelenen verinin özelliklerine en uygun modeli belirler" ifadesini kullanmaktadır.

Veri madenciliğinde kullanılan algoritma, teknik ve modeller sonuçta birer bilgisayar yazılımıdır. Bu yazılımlar matematiksel altyapı ve algoritma adımları olarak birbirlerinden farklı olsalar da ortak olan bazı özellikleri vardır. Bu özelliklerin başında veri madenciliği yazılımlarının *öğrenme* özelliği gelir. Söz konusu bu yazılımlar kendilerine verilen örnek veriler üzerinde inceleme yaparak kullandıkları algoritmalarla bu verilerden bazı sonuçlar ve kurallar çıkarırlar. Yazılımın veriler üzerinde yaptığı bu inceleme işlemine *öğrenme* adı verilir. Daha sonra yazılım bu çıkarımları verilerin kalan kısmına uygulayarak ne kadar öğrendiği konusunda kendini sınar. Bu sinama sonucunda eğer gerekli görürse başlangıçta yaptığı çıkarımlarını yeniler. Yenilenen çıkarımlar (sonuçlar, kurallar) üzerinde yapılan ayrı bir işlemle *doğrulama* gerçekleştirilir. Doğrulama işleminden sonra ise aşırı öğrenme olup olmadığı da kontrol edilir. *Aşırı öğrenme* algoritmanın çıkardığı kuralların sadece üzerinde çalıştığı veriler için geçerli olmasını, dışarıdan başka verilere uygulandığında ise geçersiz olması durumunu ifade eder. Aşırı öğrenme durumunda, mevcut veriler üzerinde uygulandığında doğru sonuç veren çıkarım ve kurallar, dışarıdan gelen yeni veriler üzerinde tam sonuç veremeyecektir (Silahtaroğlu, s.50).

Veri madenciliğinde kullanılan modeller;

- Tahmin edici modeller,
- Tanımlayıcı modeller

olmak üzere temelde iki başlık altında incelenebilir.



Tahmin Edici Modeller

Tahmin edici modeller; eldeki verilerden hareketle bir model geliştirilmesi ve geliştirilen bu model kullanılarak önceden sonuçları bilinmeyen veri kümeleri için sonuçların tahmin edilmesini amaçlar. Kisaca bilinenenden yola çıkarak bilinmeyeni tahmin etme çabasıdır. Tahmin edici modeller özellikle karar verme süreci açısından büyük önem taşır. Örneğin bankalar, müşterilerinin önceki dönemlerde kullanmış oldukları kredilere ilişkin verilerine kendi veritabanlarından ulaşabilirler. Bu verilerden hareketle, müşterilerinin daha sonraki kredi taleplerinde kredi borcunu geri ödeyip ödemeyeceği, ya da ödemelerde düzenli olup olmayacağı konusunda tahminlerde bulunabilirler. Başka bir örnek olarak bir hastanede herhangi bir hastalığa ilişkin verilerin kaydedildiği veritabanı üzerinde tahmin edici modellerin uygulanması verilebilir. Buna göre hastalığa ilişkin geçmiş olaylardan elde edilen tıbbi veriler ve hastanın durumu bir arada değerlendirilerek bir tahmin modeli oluşturulabilir. Bu model kullanılarak, hastaneye yeni gelen bir hastanın hastalığına ilişkin tahmin, testler sonrası oluşan tıbbi veriler kullanılarak yapılabilir.

Tahmin edici modellere ilişkin yazılımlardaki *öğrenme*, daha çok bir insanın öğrenme biçimine benzetilebilir. İnsana ilişkin öğrenme davranışında, insan sürekli çevresini gözleyerek yeni birşeyler öğrenir. Bu açıdan bakıldığından yaşadığımız ve bulduğumuz çevredeki herşey gerçek bir veritabanıdır. Tahmin edici modeller de kendisine verilen veritabanını inceler ve bu veritabanındaki temel unsurları birbirine benzeterek tanımlamaya, onları isimlendirmeye ve sınıflamaya çalışır. Burada öğrenme işlevinin denetimli ve denetimsiz öğrenme olarak ikiye ayrıldığını söylemek gereklidir. **Denetimli öğrenmede**, öğrenici konumunda olan algoritma, nesneler, nesnelerin özellikleri ve yine bu nesnelerin tanımlanmış, daha sonra tahmin edilmesi istenecek olan değişkenleri verilir. Veri madenciliğindeki nesneler veritabanındaki her bir kayittır. İlgili algoritma ya da yazılım veritabanına girilmiş olan nesnelerin özelliklerini değerlendirir. Veri madenciliği algoritmaları, veritabanındaki ya da daha doğru bir ifadeyle veri madenciliği için oluşturulmuş veri ambarındaki nesnelerin özelliklerini nesnelerin isimleriyle ilişkilendirerek bu nesnelerin birbirinden farklı ya da benzer, aynı sınıfın nesnelerini bulur ve öğrenir. Daha sonra, kendisine verilen değişik özelliklerini değerlendirerek bu özelliğe sahip olan nesnenin ismini tahmin eder. Denetimli öğrenmenin tersi durumuna ise **denetimsiz öğrenme** denir. Denetimsiz öğrenmede nesnelerin özellikleri verilirken tahmin için kullanılacak herhangi bir parametre diğer bir ifadeyle nesnelerin isimleri verilmez. (Silahtaroğlu, s.52)

Tahmin edici modeller kendi içinde *regresyon* (*eğri uydurma*) modelleri ve *sınıflandırma* modelleri biçiminde ikiye ayrılır.

Regresyon Modelleri: Bilindiği gibi regresyon, bağımsız değişkenler ile bağımlı değişkenler arasındaki ilişkiyi en iyi tanımlayan fonksiyonu elde etmek için uygulanan istatistiksel tekniktir. Regresyon analizinde model, değişkenler arasındaki ilişkinin net bir biçimde gösterilebildiği bir fonksiyon ile temsil edilir.

Sınıflandırma Modelleri: Sınıflama, veri sınıfı ve kavramlarını tanımlama ve ayırt etmeyi sağlayan bir model kümesini bulma sürecidir. Sınıflandırmada, veriler istatistik ve veya makine öğrenimi yöntemleri kullanılarak önceden belirlenen sınıflara atanır. Sınıflama modelleri, sınıflar önceden incelenen veriler aracılığıyla oluşturulduğundan, denetimli öğrenme modelleridir.

Regresyon ve sınıflandırma modellerinden en yaygın kullanılanlar; *karar ağaçları*, *yapay sinir ağları*, *genetik algoritmalar*, *zaman serisi analizi*, *k-en yakın komşu* ve *Bayes sınıflandırması* biçiminde sıralanabilir. Üniteye izleyen kesimde bu kavramlar kısaca açıklanmıştır.

1. **Karar ağaçları:** Karar ağaçları, sınıflandırma problemlerinde en çok kullanılan algoritmaların biridir. Bunun nedeni, karar ağaçlarının yapılandırılmasının ve anlaşılmasıının diğer algoritmalarla göre daha kolay olması ve veritabanı sistemleri ile daha kolay uyum sağlayabilmesidir. Karar ağaçları biçiminde geliştirilen veri madenciliği modeli, kökleri yukarıda, ters çevrilmiş bir ağaç'a benzetilebilir. Ağaç karar verme noktalarını temsil eden düğümler ve bu düğümleri birbirine bağlayan dallardan oluşur. En üstte yer alan düğüm kök düğüm olarak adlandırılır. Kök düğümde bazı özellikler test edilerek bu testin farklı sonuçlarına göre kök düğümden farklı yönlere dallar oluşturulur. Her bir dal yeni bir karar düğümüne bağlanır ve burada yeni bir takım özellikler test edilerek bu düğümlerden de yeni dallar türetilir. Ağaç yapısının en altında ise artık kendisinden yeni bir dal türemeyecek ve bu nedenle yaprak olarak adlandırılan düğümler bulunur. Buna göre veritabanındaki tüm kayıtlar bir ağaç yapısı biçiminde düzenlenerek ağaçta yer aldıkları dala göre sınıflandırılmış olur.
2. **Yapay sinir ağları:** Yapay sinir ağları karmaşık hesaplamaları gerçekleştiren biyolojik sinir sistemlerini model alır. Bu anlamda biyolojik sinir sistemlerinin simülasyonudur. Biyolojik sinir sistemlerinde öğrenme, sinir hücreleri arasındaki etkileşim

ile gerçekleşir. Biyolojik sinir sistemlerinin öğrenme özelliği, tanımlanan görevden bağımsız olarak esnek yapıda, karmaşık verilerin işlenmesinde hesaplamaya dayalı modellerin oluşturulmasına esin kaynağı olmuştur. Yapay sinir ağları, özellikle bağımlı ve bağımsız değişkenler arasındaki karmaşık ve doğrusal olmayan ilişkileri modelleyebilmesi açısından tercih edilir. Bununla birlikte, bu yöntemle oluşturulan modellerin yorumlanması diğerlerine göre daha zordur.

3. *Genetik algoritmalar*: Genetik algoritmalar karmaşık eniyileme problemlerinin çözümünde kullanılan bir teknolojidir. Dolayısıyla aslında doğrudan bir veri madenciliği modeli değildir. Bununla birlikte veri madenciliğinde de kullanılabilen bir eniyileme yöntemidir. Genetik algoritmalar da yapay sinir ağları gibi biyolojik mekanizmalardan esinlenerek geliştirilmiş algoritmalardır. Genetik algoritmalar doğada gözlenen evrim sürecine benzer bir yapıda ele alınan problemi, sanal olarak evrimden geçirerek çözmektedir. Problemin çözümü için öncelikle, nüfus olarak tanımlanan ve kromozomlar tarafından temsil edilen bir dizi sonuç (bir çözüm kümesi) belirlenir. Bir nüfustan alınan sonuçlar, bir öncekinden daha iyi olması beklenen yeni bir nüfusu oluşturmak için kullanılır. Yeni nüfusların seçiminde her yeni bireyin problem için çözüm olup olmadığına uygunluk fonksiyonları kullanılarak karar verilir. Burada sözü edilen kromozomlar veritabanındaki her bir kayıt ve bu kromozomlar üretilecek yeni sonuçlar hakkında birtakım bilgiler içerirler. Dolayısıyla bu bilgilerin kullanılabilmesi için kromozomların kullanılabilir biçimlere dönüştürülmesi gereklidir; bu işleme kromozomların çözümlenmesi denir.
4. *Zaman serisi analizi*: Zaman serisi analizi, zaman değişkeni ile ilişkilendirilmiş verilerin tahmin edilmesi problemlerinde kullanılır. Zaman serisi analizlerinin kullanıldığı en yaygın alan borsa işlemleridir. Bir hisse senedinin veya borsa endeksinin gelecek değeri tahmini zaman serisi problemlerine örnek oluşturur. Zaman serisi problemlerinin çözümünde istatistiksel ve istatistiksel olmayan birçok veri madenciliği algoritması kullanılmaktadır. Tahmin modellerinin oluşturulmasında geçmiş verilerden yararlanılması nedeniyle bu modeller denetimli öğrenme modellerindendir.
5. *k-en yakın komşu*: k-en yakın komşu algoritması sıklıkla kullanılan bir algoritmadır. Temel olarak algoritma sınıfları belli olan bir örnek kümesindeki gözlem değerlerini inceler. Daha sonra elde edilen bu bilgi sisteme eklenen verinin ait olduğu sınıfın tespitinde kullanılır. Sınıflandırma yapılırken veritabanındaki her bir kaydın diğer kayıtlarla olan uzaklığını hesaplanır. Ancak, bir kayıt için diğer kayıtlardan sadece k adedi göz önüne alınır. Algoritmanın isminden de anlaşılacağı gibi bu k adet kayıt, başka bir ifadeyle veritabanındaki nokta, mesafesi hesaplanan noktaya diğer kayıtlara nazaran en yakın olan kayıtlardır. Bu yöntem coğrafi bilgi sistemlerinde çok kullanılır, belirlenen bir noktaya en yakın şehir, istasyon vb. belirlenmesi aslında k-en yakın komşu algoritmasının temelini oluşturur. Algoritmda k değeri önceden seçilir; değerinin yüksek olması birbirlerine benzemeyen noktaların bir araya toplanmasına, çok küçük seçilmesiyse birbirine benzettiği, yani aynı sınıfın noktaları oldukları hâlde, bazı noktaların ayrı sınıflara konmasına ya da o tür noktalar için ayrı sınıfların açılmasına neden olur (Silahtaroğlu, s. 118). Gözlem değerlerinin arasındaki uzaklıkların hesaplanmasında “Öklid” uzaklık formülü kullanılır.
6. *Bayes sınıflandırması*: Bayes sınıflandırma yöntemi, elde var olan, mevcut sınıflanmış verileri kullanarak yeni bir verinin mevcut sınıflardan herhangi birine girme olasılığını hesaplayan yöntemdir. İstatistikte kullanılan Bayes kuralına dayalı olarak geliştirilmiş algoritma ve sınıflandırma teknikleri bu isimle anılır (Silahtaroğlu, s. 97).

Tanımlayıcı Modeller

Tanımlayıcı modeller verilerdeki örüntü veya ilişkileri tanımlar. Bu modeller tahmin edici modellerin aksine analiz edilen verilerin özelliklerini incelemek için kullanılan modellerdir. Tahmini modellerde kullanılan yazılımlar kendilerine verilen veritabanını bir bütün olarak düşünür ve öğrenme işlemini de bu bütünü temel alarak gerçekleştirir. Buna karşın tanımlayıcı modellerde, veritabanındaki kayıtlar arasında bir bağlantı, ilişki kurulmaya çalışılır. Böylece bir veritabanındaki kayıtlar arasında çok rastlanan kurallar ortaya çıkarılır. Sepet analizi olarak adlandırılan ve İnternet üzerinden alışveriş yapılan sitelerde, alışveriş sepetindeki ürünler arasındaki ilişkiyi ortaya çıkarıp, müşterinin herhangi bir ürünü seçmesinin ardından müşteriye ilgisini çekecek bir başka ürünün önerilmesi, tanımlayıcı modeller kullanılarak yapılan veri madenciliği örneğidir. Bir diğer örnek olarak sigorta policesini yenilememiş müşterilerin benzer özelliklerini belirleyecek bir kümleme çalışması verilebilir. En yaygın kullanılan tanımlayıcı modeller; *kümleme, birelilik kuralları, sıra örüntü analizi ve özetleme* biçiminde sıralanabilir.

1. **Kümleme:** Kümeleme, verileri birbirlerine olan benzerliklerine göre anlamlı ve/veya kullanışlı gruplara ayırmaktır. Eğer amaç anlamlı kümeler oluşturmaksa o zaman kümeler verilerin doğal yapısını yansıtmalıdır. Bazı durumlarda ise kümeleme veri özetleme gibi farklı amaçlar için kullanışlı bir başlangıç noktası oluşturmaktadır. Kümeleme analizi bir hedef değişken içermemişinden, diğer bir ifade ile veriler bağımlı bir değişkene göre değil öznitelik değerlerine göre gruplandırıldılarından, daha önce sözü edilen *sınıflama* analizinden farklı bir yaklaşımdır. Kümeleme analizinde, hedef değişkenin değerini belirlemeye yönelik sınıflama, tahmin etme veya kestirim yapılmaya çalışılmaz. Bunun yerine verinin tamamını böülümlere ayırmak için homojen alt gruplar veya kümeler araştırılır. Bu işlem gerçekleştirilirken kümeler içindeki verilerin benzerliği göz önüne alınır. Oluşturulan kümeler önceden tanımlanmadığından ve verinin özelliklerine göre belirlendiğinden kümelerin anlamı konuya ilgili bir alan uzmanı tarafından yorumlanmalıdır. Verilerin kümeleme analizine göre modellenmesinde matematik, istatistik, makine öğrenimi ve yapay zekâ gibi birçok alandan yararlanılır. Kümeleme sürecinde bağımlı ve bağımsız değişkenler arasında bir bağ kurmak söz konusu olmadığından, kümeleme yaklaşımı makine öğreniminde denetimsiz öğrenme başlığı altında yer alır. Diğer bir ifadeyle kümelemedeki öğrenmenin denetimsiz öğrenme olmasının nedeni önceden belirlenmiş sınıfların olmayacağıdır. Önceden belirli sınıflar olsaydı, bu durumda kullanılan model zaten bir sınıflandırma modeli olacaktır. Önceden sınıflar belirli iken, yani kadın ve erkek diye iki ayrı sınıf varken yapılan (algoritmik) öğrenmeye denetimli öğrenme; herhangi bir sınıf ismi verilmeden yapılan öğrenmeye denetimsiz öğrenme denilir. Örneğin, veritabanındaki kayıtlarda her kadın yanına kadın veya erkek bilgisi yazılıyor olsun, bu durumda veritabanı üzerinde yapılan herhangi bir (kadın veya erkek olduğuna dair) kural çıkarma işlemi denetimli öğrenmedir. Ancak aynı veritabanında, kayıtların yanında kadın mı erkek mi olduğu bilgisi yok iken yapılan kural çıkarma işlemi denetimsiz öğrenmedir. Bu işlem aynı zamanda veritabanını (iki) kümeye ayırma, yani kümeleme işlemidir. Burada kadın/erkek gibi bir etiket ya da sınıf olmayacağı için kümeleme kayıtlar arasındaki benzerlik veya mesafe ölçüfüne göre yapılır. İki verinin benzerliğinden kasıt ise aralarındaki mesafenin ölçülmesi ve değerlendirilmesidir. Bu değerlendirme, veritabanındaki diğer verilere kıyasla iki verinin ne kadar yakın ya da benzer oldukları açısından yapılabileceği gibi önceden belirlenmiş kısıtlar eşik değerleri çerçevesinde de yapılabilir (Silahtaroğlu, s. 59-60).

2. *Birliktelik kuralları:* Birliktelik kuralları veriler arasındaki güçlü birliktelik özelliklerini tanımlayan örüntüleri keşfetmek için kullanılan analiz yöntemidir. Birliktelik kuralı, belirli türdeki veri ilişkilerini tanımladığı için tanımlayıcı modeller içinde yer almaktadır. Herhangi bir ürün alındığında bir başka ürünün de satın alınması bir birliktelik kuralı verir. İş dünyasında birliktelik analizi, pazar sepeti veya benzeşme analizi olarak da adlandırılır ve müşterilerin satın alma alışkanlıklarını analiz ederek, ilgili ürünler arasındaki potansiyel çapraz satış olanaklarını tanımlamak için kullanılır. Örneğin; "Bira satın alan müşteriler %80 olasılıkla cips de satın alırlar" ya da "Düşük yağlı peynir satın alan müşteriler %90 olasılıkla yağız yoğurt da satın alırlar" biçimindeki sonuçlara birliktelik kuralları analizi ile ulaşılabilir. Raf düzenlemeleri bu sonuçlar temel alınarak yapıldığında satış oranları artırılabilir.
3. *Sıra örüntü analizi:* Sıra örüntü analizi birliktelik kurallarına benzer bir yapıda olup aynı zamanda olayların zaman sıralarıyla ilgilendir. Birliktelik kurallarında sözü edilen pazar sepeti analizinde, ürünlerin müşteri tarafından aynı anda alınmasıyla ilgilenilirken sıra örüntüleri analizinde belirli bir zaman aralığında satın alınan ürünler arasındaki ilişkilerle ilgilenilir. "A ameliyatı olan bir hasta, 10 gün içinde %40 olasılıkla B enfeksiyonu olacakaktır", "Menkul Kıymetler Borsası endeksi düşerken A hisse senedinin değeri %20'den daha fazla artacak olursa, üç iş günü içinde B hisse senedinin değeri %60 olasılıkla artacakaktır" ya da "Çekiç satın alan bir müşteri, ilk üç ay içerisinde %15, bu dönemi izleyen üç ay içerisinde %10 olasılıkla çivi satın alacakaktır" biçiminde sıralanabilecek ilişki tanımlamaları, sıra örüntü analizi ile tanımlanabilecek ilişkilere örneklerdir.
4. *Özetleme:* Karakterizasyon veya genelleştirme olarak da adlandırılan özetleme, verileri basit tanımları yapılmış alt gruplar içine yerleştirme işlemidir. Özetteleme veritabanı hakkında betimleyici bilgileri ortaya çıkarır ve verilerden elde edilen ortalamaya veya standart sapma gibi tüm veriyi temsil eden göstergelerin hesaplanması ifade eder. Özettebilgiler, veritabanı fonksiyonları ve tanımlayıcı veri madenciliği teknikleri kullanılarak elde edilebilir.

Veri madenciliğinde kullanılan modeller nasıl sınıflandırılır? Bu sınıflandırma içinde yer alan algoritmalarla örnekler veriniz.



SIRA SİZDE

VERİ MADENCİLİĞİNİN DİĞER VERİ ANALİZİ YAKLAŞIMLARI İLE KARŞILAŞTIRILMASI

Veri madenciliği ile veri analizi amacıyla kullanılan diğer yaklaşımlar farklı açılardan karşılaştırılabilir. Buna göre veri madenciliği ile geleneksel istatistiksel analiz, veri sorgusu, SQL (Yapilandırılmış Sorğu Dili), OLAP (Çevrimiçi Analitik İşleme) gibi diğer yaklaşımalar karşılaştırıldığında izleyen kesiminde verilen farklılıklar olduğu görülmektedir.

Geleneksel istatistiksel analiz ile veri madenciliği arasındaki temel farklar aşağıdaki gibi sıralanabilir:

- İstatistiksel analizde, analize genellikle bir hipotez kurularak başlanırken veri madenciliği ile analizde herhangi bir hipoteze gerek duyulmaz.
- İstatistikçiler hipotezlerini eşleştirmek için kendi eşitliklerini geliştirmek zorunda oldukları hâlde, veri madenciliği algoritmaları eşitlikleri otomatik olarak geliştirir.
- İstatistiksel analizler genellikle sayısal veriler üzerinde gerçekleştirilirken veri madenciliği sayısal verilere ek olarak metin, ses vb. gibi farklı veri türleri üzerinde de işlem yapabilir.
- İstatistiksel analizde, kirli veri analiz sırasında bulunur ve filtre edilirken veri madenciliği temizlenmiş veri üzerinde gerçekleştirilir.

- İstatistiksel analizde bulunan sonuçlar kolaylıkla yorumlanabilirken veri madenciliğinin sonuçlarını değerlendirmek ve yorumlamak aynı derecede kolay olmayıp uzman istatistikçilere gereksinim duyulur.

Veri sorgusu, OLAP ve veri madenciliği, kullanım amacına göre karşılaştırıldığında;

- Veri sorgusu, aranan (ulaşılmak istenen) bilginin ne olduğu bilindiği durumda ve büyük veri tabanı ile çalışılmak istediği durumlarda,
- OLAP, büyük veritabanlarında veriler arasındaki basit ilişkilerin keşfedilmek istediği durumlarda,
- Veri madenciliği, veriler arasında var olan fakat açıkça gözlenemeyen örüntü ve ilişkilerin keşfedilmesi istediği durumlarda

kullanılır.

SQL, OLAP ve veri madenciliği, keşfedilmek istenen bilgi tipine göre karşılaştırıldığına ise,

- Seçilen kayıtlara ait ortalama ve toplam değer gibi özet bilgiler *sığ bilgi* olarak tanımlanır. Bu tür bilgilere ulaşmak için SQL kullanımı yeterlidir.
- Farklı özelliklerin ortaya çıkma sıklığılarındaki bilgi *çok boyutlu bilgi* olarak nitelendirilir. Bu tür bilgiye ulaşma işlemini veri küpü üzerinden OLAP yapabilir.
- Önceden tahmin edilemeyen örüntü ve ilişkiler *gizli bilgi* olarak ifade edilebilir ve bu örüntü ve ilişkiler veri madenciliği için başlangıç olabilir.
- Sadece önsel teknik veya meta-bilginin kullanımıyla keşfedilebilecek gizli örüntüler ve ilişkiler hakkında bilgi ise *derin bilgi* olarak tanımlanabilir ve bunlar da veri madenciliğinin araştırma sınırları içinde yer alır (Koyuncugil ve Özgülbaş, s. 25).

SIRA SİZDE



Geleneksel istatistiksel analiz ile veri madenciliği arasındaki temel farklar nelerdir?

VERİ MADENCİLİĞİNİN UYGULANDIĞI ALANLAR

Kâr amacı güden ya da gütmeyen tüm kuruluşlarda, kurumun yaşamını sürdürmesi öncelikli amaçlardan biridir. Bu amacın başarılabilmesi ise her gün değişen ve yenilenen koşullara uyum sağlayabilme becerisi ile sağlanabilecektir. Bu nedenle yalnızca deneyim ve önsezilere dayanarak kararlar vermek beraberinde yüksek riski de getirecektir. Bu riski azaltmanın yolu ise doğru karar destek sistemlerinden yararlanmaktır. Doğru karar destek sistemlerinin oluşturulması söz konusu olduğunda ise veri madenciliği teknikleri çok önemli araçlar olarak karşımıza çıkmaktadır. Bu araçların kullanımıyla, kurumların ve işletmelerin etkin kararlar almak için ihtiyaç duydukları bilgilere erişmeleri de mümkün olacaktır.

Veri madenciliğinin uygulandığı alanlar kesin çizgilerle sınırlanılamaz. Büyük miktarда verinin üretildiği ve kaydedildiği ve karar verme sürecine ihtiyaç duyulan tüm alanlarda veri madenciliği uygulamaları yapmak mümkündür. Veri madenciliğinin yoğun ve başarılı bir biçimde kullanıldığı başlıca alanlar; pazarlama, finans (bankacılık, sigortacılık, borsa), parekendecilik, sağlık, telekomünikasyon, endüstri ve mühendislik, eğitim, tıp, biyoloji, genetik, kamu, istihbarat ve güvenlik biçiminde sıralanabilir.

Pazarlama Alanındaki Uygulamalar

Veri madenciliğinin en çok kullanıldığı alanların başında pazarlama alanının geldiği söylenebilir. Yapılan çalışmalar incelendiğinde, pazarlama alanında yapılan veri madenciliği uygulama konuları izleyen biçimde sıralanabilir.

- Müşterilerin satın alma örüntülerinin belirlenmesi
- Benzer özellikler gösteren müşterilerin bulunması
- Müşterilerin demografik özellikleri arasındaki bağlantıların belirlenmesi
- Benzer gelir grupları, ilgi alanları, harcama alışkanlıklarının ortaya çıkarılması

- Benzer müşterileri otomatik olarak gruplayarak, pazar dilimlerinin tanımlanması ve bu bilginin pazarlama kampanyalarında kullanılması
- Mevcut müşterilerin elde tutulması, yeni müşterilerin kazanılması
- Satış tahmini yapılması
- Müşteri ilişkileri yönetimi
- İnternet üzerinden satış yapan işletmeler için kullanıcı profillerinin belirlenmesi
- Web sayfalarının kullanıcı bilgilerine göre kişiselleştirilmesi

Finans Alanındaki Uygulamalar

Veri madenciliğinin sıkılıkla kullanıldığı bir diğer alan bankacılık, sigortacılık ve borsa olarak sıralayabileceğimiz finans sektörüdür. Finans sektöründeki veri madenciliği uygulama konuları da izleyen biçimde sıralanabilir.

- Farklı finansal göstergeler arasındaki gizli korelasyonların bulunması
- Müşteri kaybı analizi
- Kredi kartı dolandırıcılıklarının belirlenmesi
- Müşteriler arasındaki benzerliklerin belirlenmesi
- Kredi kartı harcamalarına göre müşteri gruplarının belirlenmesi
- Kredi kartı ve kredi taleplerinin değerlendirilmesi
- Risk analizi ve risk yönetimi
- Hisse senedi fiyatlarının tahmin edilmesi
- Yatırımların modellenmesi
- Sigorta dolandırıcılıklarının belirlenmesi
- Sigorta yapturan müşteriler içinde riskli müşteri grubunun belirlenmesi

Sağlık Alanındaki Uygulamalar

Veri madenciliğinin sağlık alanındaki uygulamaları;

- Yeni ilaçların geliştirilmesi,
- Piyasada var olan ilaçların etkilerinin belirlenmesi,
- Hastalara uygulanan test sonuçlarının tahmin edilmesi,
- Hastalıkların önceden teşhis ve tedavi edilmesi

biçiminde sıralanan konularda yapılmış olup önemli etkileri olan uygulama alanlarından biridir.

Endüstri ve Mühendislik Alanındaki Uygulamalar

Endüstri ve mühendislik alanında veri madenciliğinden;

- Kurum kaynaklarının optimal kullanımı,
 - Üretim süreçlerinin kontrol edilmesi,
 - Kalite kontrol analizlerinin gerçekleştirilmesi,
 - Sistem performanslarına etki eden faktörlerin ve kuralların belirlenmesi
- konularında yararlanılmaktadır.

Eğitim Alanındaki Uygulamalar

Eğitim alanında yapılan veri madenciliği uygulama konuları ise izleyen biçimde sıralanabilir.

- Öğrenci verilerinin analiz edilmesi
- Öğrenci başarı ve başarısızlık nedenlerinin tespit edilmesi
- Öğrenci başarılarının artırılması
- Eğitim-öğretim ortamlarındaki aksaklılıkların tespit edilmesi
- Daha etkili eğitim-öğretim ortamlarının oluşturulması

Veri madenciliğinin uygulandığı alanlara örnekler veriniz.



Özet



Veri madenciliğinin tarihsel gelişimini özetlemek

İlk bilgisayarların geliştirildiği 1950'li yıllarda, bilgisayarlar sayımlar ve karmaşık hesaplamaları kolaylıkla yapabilmek amacıyla kullanılmaktaydı. 1960'lı yillardan itibaren veriler bilgisayarlarda depolanmaya başlanmış ve buna bağlı olarak veri tabanlarının kullanımı ortaya çıkmıştır. 1960'ların sonuna gelindiğinde basit öğrenmeli bilgisayarlar geliştirilmiştir. Zaman içinde giderek büyütlenen veri tabanlarının organizasyonu ve yönetimi sorununun üstesinden gelebilmek amacıyla ise veri modelleme kavramı ortaya atılmıştır. Hiyerarşik Veri Modeli ve Ağ Veri Modeli olarak adlandırılan ilk veri modellerini 1970'lerde İlişkisel Veri Tabanı Yönetimi Sistemlerinin kullanılmaları izlemiştir. 1980'lerde veri tabanı yönetim sistemleri yaygınlaşmış ve pek çok farklı alanda uygulanır olmuştur. Bunu, veri miktarının sürekli katlanarak arttığı veri tabanları içinden, faydalı bilgilerin nasıl çıkarılabileceği konusundaki çalışmalar izlemiştir. 1989'da yapılan KDD (Knowledge Discovery in Database) IJCAI-89 Veri Tabanlarında Bilgi Keşfi Çalışma Grubu toplantısının sonuç bildirgesi sayılan makale 1991'de yayımlanmış ve veri madenciliği ile ilgili temel tanım ve kavramlar ortaya konmuştur. 1992 yılında veri madenciliği için ilk yazılım geliştirilmiş ve sonrasında bu alana ilgi hızla artmıştır. 2000'li yılların başından itibaren veri madenciliği sürekli gelişmiş ve hemen hemen tüm alanlara uygulanmaya başlanmıştır. Günümüzde de veri madenciliği pek çok alanda yaygın olarak kullanılmaktadır.



Veri madenciliğine etki eden disiplinleri betimlemek

Veri madenciliğinin etkileşimde olduğu pek çok farklı disiplin bulunmaktadır. Bunların başlıcaları; istatistik, makine öğrenimi, görselleştirme, veritabanı sistemleri, örtü tanıma biçiminde sıralanabilir. İstatistik, verilerin analizi ve değerlendirilmesi konusunda geçmişten günümüze yoğun bir biçimde kullanılan bir disiplindir. Makine öğrenimi, kısaca bilgisayarların bazı işlemlerden çıkarsamalar yaparak yeni işlemler üretmesi olarak tanımlanabilir. Görselleştirme; verilerin, tablolar ve grafikler gibi görseller yardımıyla sunulmasını sağlayan teknolojileri ifade eder. Veritabanı birbirile ilişkili olan ve amaca uygun biçimde düzenlenmiş, mantıksal ve fiziksel olarak tanımlanmış veriler bütünüdür. Örtü tanıma teknolojisi ise daha önce tanımlanmış, bir model olarak düşünülebilen çok boyutlu bir örtüünün veritabanındaki benzerlerini ya da en benzerini arama ve bulma amacıyla yönelik yazılımları ifade eder.



Veri madenciliği kavramını tanımlamak

Veri madenciliği, büyük miktardaki veri yiğinları üzerinde analiz yaparak, veriler arasında var olan ve geleceğin tahmin edilmesine yardımcı olacak anlamı ve yararlı ilişki ve kuralların bilgisayar yazılımları aracılığıyla aranması faaliyetleridir. Diğer bir tanıma göre veri madenciliği, veri analizi için gelişmiş ve karmaşık araçlar kullanarak yiğin veri kümeleri içinden daha önceden bilinmeyen olgu ve olayları keşfetmek ve veriler arasındaki mantıklı ilişkileri ve kalıpları ortaya çıkarmak amacıyla yapılan çalışmalardır. Burada vurgulanması gereken önemli nokta, veri madenciliği ile elde edilecek bilginin daha önceden bilinmeyen ve yeni keşfedilen olmasıdır. Önceden bilinmeyen bilgi, önceden tahmin bile edilemeyen bilgi anlamındadır.



Veritabanlarında bilgi keşfi sürecini açıklamak

Veritabanlarında Bilgi Keşfi, veriden faydalı bilginin keşfedilmesi sürecinin tamamıdır. Veri madenciliği ise bu sürecin bir adımı olup veriden örtüleri belirlenmesi ve aktarımı için özel algoritmaların uygulanması işlemlerine karşılık gelmektedir. Veritabanlarında Bilgi Keşfi sürecinin adımları aşağıdaki biçiminde sıralanır.

1. Amacın Tanımlanması: İşletmenin ya da kurumun veri madenciliğini hangi amaca yönelik olarak gerçekleştirmek istediğini belirlediği aşamadır.
2. Veriler Üzerinde Ön İşlemlerin Yapılması: Verilerin, veri madenciliği işlemlerine hazırlanması sürecidir. Bu aşamada; verilerin toplanması ve birleştirilmesi, kayıp veriler için işlem uygulanması, verideki gürültünün temizlenmesi, verilerin normalizasyonu, verilerin azaltılması ve verilerin dönüştürülmesi işlemleri gerçekleştirilir.
3. Modelin Kurulması ve Değerlendirilmesi: Veri madenciliği modelinin kurulduğu ve geçerli bir model olup olmadığını değerlendirdiği aşamadır.
4. Modelin Kullanılması ve Yorumlanması: Kurulan ve geçerliliği sınanarak uygulanmak üzere kabul edilen modelin kullanıldığı ve sonuçların yorumlandığı aşamadır.
5. Modelin İzlenmesi: Sistemin ne kadar iyi çalıştığını izlendiği ve gözlenen sonuçlarına göre gerekiyorsa modelde değişiklik ve düzenlemelerin yapıldığı aşamadır.



Veri madenciliğinde kullanılan modellere ilişkin özellikleri özetlemek

Veri madenciliğinde kullanılan modeller;

- Tahmin edici modeller
- Tanımlayıcı modeller

olmak üzere temelde iki başlık altında incelenebilir.

Tahmin edici modeller; eldeki verilerden hareketle bir model geliştirilmesi ve geliştirilen bu model kullanılarak önceden sonuçları bilinmeyen veri kümeleri için sonuçların tahmin edilmesini amaçlar. Tahmin edici modeller kendi içinde *regresyon (eğri uydurma)* modelleri ve *sınıflandırma* modelleri biçiminde ikiye ayrılır. Regresyon modelleri ilgili değişkenler arasındaki ilişkiyi en iyi tanımlayan fonksiyonu elde etmek ve buna göre tahminde bulunmak temeline dayanır. Sınıflama modellerinde veriler, istatistik ve/veya makine öğrenimi yöntemleri kullanılarak önceden belirlenen sınıflara atanır. Regresyon ve sınıflandırma modellerinden en yaygın kullanılanlar; *karar ağaçları, yapay sinir ağları, genetik algoritmalar, zaman serisi analizi, k-en yakın komşu ve Bayes sınıflandırması* biçiminde sıralanabilir.

Tanımlayıcı modeller ise verilerdeki örtüyü veya ilişkileri tanımlar. Bu modeller tahmin edici modellerin aksine analiz edilen verilerin özelliklerini incelemek için kullanılan modellerdir. Tanımlayıcı modeller veritabanındaki kayıtlar arasında bir bağlantı, ilişki kurmaya çalışarak veritabanındaki kayıtlar arasında çok rastlanan kuralları ortaya çıkarır. En yaygın kullanılan tanımlayıcı modeller de *kümeleme, birlikteki kuralları, sıra örtüyü analizi ve özetleme* biçiminde sıralanabilir.



Veri madenciliğini diğer veri analizi yaklaşımları ile karşılaştırılmak

Geleneksel istatistiksel analiz ile veri madenciliği arasındaki temel farklar aşağıdaki gibi sıralanabilir:

- İstatistiksel analizde, analize genellikle bir hipotez kurularak başlanırken veri madenciliği ile analizde herhangi bir hipoteze gerek duyulmaz.
- İstatistikçiler hipotezlerini eşleştirmek için kendi eşitliklerini geliştirmek zorunda oldukları hâlde, veri madenciliği algoritmaları eşitlikleri otomatik olarak geliştirir.
- İstatistiksel analizler sadece sayısal veriler üzerinde gerçekleştirilirken veri madenciliği sayısal verilere ek olarak metin, ses vb. gibi farklı veri türleri üzerinde de işlem yapabilir.
- İstatistiksel analizde, kirli veri analiz sırasında bulunur ve filtre edilirken veri madenciliği temizlenmiş veri üzerinde gerçekleştirilir.

- İstatistiksel analizde bulunan sonuçlar kolaylıkla yorumlanabilirken veri madenciliğinin sonuçlarını değerlendirmek ve yorumlamak aynı derecede kolay olmayıp uzman istatistikçilere gereksinim duyulur.

Veri sorgusu, OLAP ve veri madenciliği, kullanım amacına göre karşılaştırıldığında;

- Veri sorgusu, aranan bilginin ne olduğu bilindiği durumda ve büyük veri tabanı ile çalışılmak istediği durumlarda,
- OLAP, büyük veritabanlarında veriler arasındaki basit ilişkilerin keşfedilmek istediği durumlarda,
- Veri madenciliği, veriler arasında var olan fakat açıkça gözlenemeyen örtüyü ve ilişkilerin keşfedilmesi istediği durumlarda

kullanılır.



Veri madenciliğinin uygulandığı alanları örneklemek

Veri madenciliğinin yoğun ve başarılı bir biçimde kullanıldığı başlıca alanlar; pazarlama, finans (bankacılık, sigortacılık, borsa), parekendecilik, sağlık, telekomünikasyon, endüstri ve mühendislik, eğitim, tıp, biyoloji, genetik, kamu, istihbarat, güvenlik biçiminde sıralanabilir. Bazı alanlarda yapılan uygulama örnekleri aşağıda verilmiştir:

- Pazarlama alanında: Benzer müşterileri otomatik olarak gruplayarak pazar dilimlerinin tanımlaması ve bu bilginin pazarlama kampanyalarında kullanılması
- Finans alanında: Kredi kartı dolandırıcılıklarının belirlenmesi
- Sağlık alanında: Hastalıkların önceden teşhis ve tedavi edilmesi
- Endüstri ve mühendislik alanında: Sistem performanslarına etki eden faktörlerin ve kuralların belirlenmesi
- Eğitim alanında: Öğrenci başarı ve başarısızlık nedenlerinin tespit edilmesi

Kendimizi Sınayalım

- 1.** İşletmelere ait veritabanlarının, belirli bir amaca göre konu odaklı olarak定制过的, veri madenciliğinde doğrudan kullanılabilir duruma getirilmiş hali aşağıdakilerden hangisidir?
 - a. Veri deposu
 - b. OLAP sunucusu
 - c. Üst veri
 - d. Veri ambarı
 - e. Veri martı

- 2.** Veri analizi için, gelişmiş ve karmaşık araçlar kullanarak yoğun veri kümeleri içinde daha önceden bilinmeyen olgu ve olayları keşfetmek ve veriler arasındaki mantıklı ilişkileri ve kalıpları ortaya çıkarmak amacıyla yapılan çalışmalarla ne ad verilir?
 - a. Makine öğrenimi
 - b. İstatistik
 - c. Örütü tanıma
 - d. Veri işçiliği
 - e. Veri madenciliği

- 3.** Aşağıdakilerden hangisi verilerdeki gürültünün temizlenmesi amacıyla kullanılan yaklaşımlardan biridir?
 - a. Kayıtlarda yer alan diğer değişkenler yardımıyla gürültünün temizlenmesi
 - b. Gürültülü verilerin hepsi için aynı değerin girilmesi
 - c. Verilerin tek tek yazılması ile gürültünün ortadan kaldırılması
 - d. Bölümleme yöntemiyle gürültünün temizlenmesi
 - e. Gürültülü verinin göz ardı edilmesi

- 4.** Verilerin normalizasyonu ve verilerin azaltılması işlemleri hangi başlık altında ele alınır?
 - a. Verilerin yeniden yapılandırılması
 - b. Verilerin temizlenmesi
 - c. Verilerin bireştirilmesi
 - d. Verilerin derlenmesi
 - e. Verilerin sınıflandırılması

- 5.** Veri madenciliğinde, analize konu olan veri kümесinin gerekli veriyi içermesi ancak verinin kullanılan algoritmayla uygun yapıda olmaması durumunda gerçekleştirilen işleme ne ad verilir?
 - a. Verilerin azaltılması
 - b. Verilerin temizlenmesi
 - c. Verilerin dönüştürülmesi
 - d. Verilerin bütünlendirilmesi
 - e. Verilerin ayırtılması

- 6.** Aşağıda verilen ve veri madenciliğinde kullanılan modellerden hangisi sınıflandırma modelleri arasında yer alır?
 - a. Birliktelik kuralları
 - b. Karar ağaçları
 - c. Öztleme
 - d. Kümeleme
 - e. Sıra örütü analizi

- 7.** Doğada gözlenen evrim sürecine benzer bir yapıda ele alınan problemi, sanal olarak evrimden geçirerek çözen veri madenciliği modeli aşağıdakilerden hangisidir?
 - a. Yapay sinir ağları
 - b. Genetik algoritmalar
 - c. Makine öğrenmesi
 - d. k-en yakın komşu
 - e. Bayes sınıflandırması

- 8.** Aşağıdakilerden hangisi verileri birbirlerine olan benzerliklerine göre anlamlı ve/veya kullanışlı gruppala ayıran veri madenciliği modelidir?
 - a. Birliktelik kuralları
 - b. Sıra örütü analizi
 - c. Kümeleme
 - d. Zaman serisi analizi
 - e. k-en yakın komşu

- 9.** Yeni ev alan bir kişinin bir ay içinde koltuk takımı alma olasılığının %80 olduğu biçiminde bir ilişki tanımlayabilen veri madenciliği modeli aşağıdakilerden hangisidir?
 - a. Birliktelik kuralları
 - b. Sıra örütü analizi
 - c. Kümeleme
 - d. Zaman serisi analizi
 - e. k-en yakın komşu

- 10.** Aşağıdakilerden hangisi veri madenciliğinde tanımlayıcı modeller için geçerli olan bir özellikdir?
 - a. Kendilerine verilen veritabanını bir bütün olarak düşünür.
 - b. Önceden sonuçları bilinmeyen veri kümeleri için sonuçları tahmin etmeye çalışır.
 - c. Regresyon modelleri tanımlayıcı modeller içinde değerlendirilir.
 - d. Sınıflandırma modelleri tanımlayıcı modeller içinde değerlendirilir.
 - e. Kurulan model analiz edilen verilerin özelliklerini incelemek için kullanılır.

Kendimizi Sınavalım Yanıt Anahtarları

- | | |
|-------|--|
| 1. d | Yanıtınız yanlış ise “Veri Madenciliği Kavramı” konusunu yeniden gözden geçiriniz. |
| 2. e | Yanıtınız yanlış ise “Veri Madenciliği Kavramı” konusunu yeniden gözden geçiriniz. |
| 3. d | Yanıtınız yanlış ise “Veriler Üzerinde Ön İşlemlerin Yapılması” konusunu yeniden gözden geçiriniz. |
| 4. a | Yanıtınız yanlış ise “Veriler Üzerinde Ön İşlemlerin Yapılması” konusunu yeniden gözden geçiriniz. |
| 5. c | Yanıtınız yanlış ise “Veriler Üzerinde Ön İşlemlerin Yapılması” konusunu yeniden gözden geçiriniz. |
| 6. b | Yanıtınız yanlış ise “Veri Madenciliğinde Kullanılan Modeller” konusunu yeniden gözden geçiriniz. |
| 7. b | Yanıtınız yanlış ise “Tahmin Edici Modeller” konusunu yeniden gözden geçiriniz. |
| 8. c | Yanıtınız yanlış ise “Tanımlayıcı Modeller” konusunu yeniden gözden geçiriniz. |
| 9. b | Yanıtınız yanlış ise “Tanımlayıcı Modeller” konusunu yeniden gözden geçiriniz. |
| 10. e | Yanıtınız yanlış ise “Tanımlayıcı Modeller” konusunu yeniden gözden geçiriniz. |

Sıra Sizde Yanıt Anahtarları

Sıra Sizde 1

Veri madenciliği çalışmalarının yapıldığı veritabanları, işletmelerin günlük kayıtlarının yer aldığı ve *işlemsel veritabanı* olarak adlandırılan veri tabanları değildir. Daha doğru bir ifadeyle işletmeli veritabanları veri madenciliği uygulamalarında doğrudan kullanılmaz. Bu veritabanlarında yer alan veriler birtakım işlemlerden geçirilerek veri madenciliği için kullanılabilir, hazır hâle getirilir. İşte işletmelere ait veritabanlarının, belirli bir amaca göre konu odaklı olarak düzenlenmiş, veri madenciliğinde doğrudan kullanılabilir duruma getirilmiş hali *veri ambarı* olarak tanımlanır. İşletmelerdeki veri kaynakları işletme içindeki kaynaklar ve işletme dışındaki kaynaklar biçiminde sınıflandırılabilir. Veri ambarları, söz konusu bu iç ve dış kaynaklı verilerin biraraya getirilmesi ile oluşturulan özel veritabanlarıdır. Bununla birlikte verilerin birleştirilmesi gelişigüzel bir işlem değildir. Veriler farklı kaynaklardan elde edildiği için veriler arasındaki uyumsuzlukların, tutarsızlıkların giderilmesi ve verilerin amaca uygun, kullanabilecek biçimde dönüştürülmesi gereklidir. Veri ambarında yer alan veriler hakkındaki tanımlamalar üst veri (meta data) olarak adlandırılır ve veri ambarına ilişkin veri kataloğu olarak düşünülebilir. Veri deposu (data mart) kavramı ise veri ambarının bir alt kümesi olup işletmenin yal-

nızca belirli bir bölümünü ya da belirli bir iş sürecini, daha özel bir fonksiyon alanını ilgilendiren parçasıdır. Veri ambarı tüm işletmeyi ilgilendirirken veri deposu tek bir konuya ya da özel bir amaca yönelik verileri içerir.

Sıra Sizde 2

Veri tabanlarından bilgi keşfi sürecinde, veriler üzerinde yapılan ön işlemler genel olarak:

- Verilerin toplanması ve birleştirilmesi,
- Verilerin temizlenmesi,
- Verilerin yeniden yapılandırılması

biriminde sınıflandırılabilir.

Verilerin veri madenciliğine hazırlanabilmesi için yapılması gereken ilk şey doğal olarak verilerin belirlenmesidir. Bu yapılrken öncelikle tanımlanan amaca ve probleme uygun verilerin neler olduğu ve bu verilerin hangi kaynaklarda yer aldığı araştırılır. Bu belirleme sonrası veriler bulundukları farklı kaynaklardan toplanır ve birleştirilir. Veritabanlarından alınan kayıtların bir kısmında, diğer kaytlarda var olan bazı veriler eksik olabilir. Böylece eksik veriler kayıp veri olarak adlandırılır. Bunun yanı sıra bazı kaytlarda yer alan bir kısım veriler doğru olamayacak kadar uç değerlerde, dolayısıyla yanlış girilmiş olabilir. Aşırı uç değerler aykırı değer, bu şekildeki uç verilerin geneli de gürültülü veri olarak nitelendirilir. Verilerin temizlenmesi, kayıp ya da eksik değerleri tamamlamak, aykırı değerleri belirleyerek gürültüyü ortadan kaldırırmak ve verilerdeki tutarsızlıklar, uyumsuzlukları gidermek için kullanılan birçok yaklaşımı ve teknigi kapsar. Eldeki verilerin kullanılacak veri madenciliği algoritmasına uygun hâle getirilmesi, diğer bir ifadeyle yeniden yapılandırılması gereklidir. Bu amaçla gerçekleştirilen işlemler; verilerin normalizasyonu, verilerin azaltılması ve verilerin dönüştürülmesi başlıklar altında incelenebilir.

Sıra Sizde 3

Veri madenciliğinde kullanılan modeller;

- Tahmin edici modeller,
- Tanımlayıcı modeller

olmak üzere temelde iki başlık altında incelenebilir.

Tahmin edici modeller; eldeki verilerden hareketle bir model geliştirilmesi ve geliştirilen bu model kullanılarak önceden sonuçları bilinmeyen veri kümeleri için sonuçların tahmin edilmesini amaçlar. Tahmin edici modeller kendi içinde *regresyon* modelleri ve *sınıflandırma* modelleri biçiminde ikiye ayrılır. Regresyon ve sınıflandırma modellerinden en yaygın kullanılan algoritmalar; *karar ağaçları*, *yapay sinir ağları*, *genetik algoritmalar*, *zaman serisi analizi*, *k-en yakın komşu* ve

Yararlanılan ve Başvurulabilecek Kaynaklar

Bayes sınıflandırması biçiminde sıralanabilir.

Tanımlayıcı modeller verilerdeki örüntü veya ilişkileri tanımlar. En yaygın kullanılan tanımlayıcı modeller ise *kümleme, birlikte kuralları, sıra örüntü analizi* ve *özetleme* biçiminde sıralanabilir.

Sıra Sizde 4

Geleneksel istatistiksel analiz ile veri madenciliği arasındaki temel farklar aşağıdaki gibi sıralanabilir:

- İstatistiksel analizde, analize genellikle bir hipotez kurularak başlanırken veri madenciliği ile analizde herhangi bir hipoteze gerek duyulmaz.
- İstatistikçiler hipotezlerini eşleştirmek için kendi eşitliklerini geliştirmek zorunda oldukları hâlde, veri madenciliği algoritmaları eşitlikleri otomatik olarak geliştirir.
- İstatistiksel analizler sadece sayısal veriler üzerinde gerçekleştirilirken, veri madenciliği sayısal verilere ek olarak metin, ses vb. gibi farklı veri türleri üzerinde de işlem yapabilir.
- İstatistiksel analizde, kirli veri analiz sırasında bulunur ve filtre edilirken, veri madenciliği temizlenmiş veri üzerinde gerçekleştirir.
- İstatistiksel analizde bulunan sonuçlar kolaylıkla yorumlanabilirken, veri madenciliğinin sonuçlarını değerlendirmek ve yorumlamak aynı derecede kolay olmayıp uzman istatistikçilere gereksinim duyulur.

Sıra Sizde 5

Veri madenciliğinin uygulandığı alanlar kesin çizgilerle sınırlanılamaz. Büyük miktarda verinin üretildiği ve kaydedildiği ve karar verme sürecine ihtiyaç duyulan tüm alanlarda veri madenciliği uygulamaları yapmak mümkündür. Aşağıda bazı uygulama örnekleri verilmiştir:

Pazarlama alanında: İnternet üzerinden satış yapan işletmeler için kullanıcı profillerinin belirlenmesi

Finans alanında: Sigorta yapışran müşteriler içinde riskli müşteri grubunun belirlenmesi

Sağlık alanında: İlaç firmalarının yeni ilaç geliştirebilmesi

Endüstri ve mühendislik alanında: Kurum kaynaklarının optimal kullanımı

Eğitim alanında: Daha etkili eğitim-öğretim ortamlarının oluşturulması.

Akpınar, H. (2000). *Veritabanlarında Bilgi Keşfi ve Veri Madenciliği*, İ.Ü. İşletme Fakültesi Dergisi, C.29, Nisan, s. 1-22.

Akpınar, H. (2014). *Data Veri Madenciliği Veri Analizi*, İstanbul: Papatya Yayıncılık Eğitim.

Alagöz, A., Öge, S. ve Ortakarpuz, M. (2014). *Bir Kurumsal Zeka Teknolojisi Olarak Veri Madenciliği ile Muhasebe Bilgi Sistemi İlişkisi*, Selçuk Üniversitesi Sosyal Bilimler Enstitüsü Dergisi, Dr. Mehmet Yıldız Özel Sayısı, s.1-21.

Aydın, S. (2007). *Veri Madenciliği ve Anadolu Üniversitesi Uzaktan Eğitim Sisteminde Bir Uygulama*, (Basilılmış doktora tezi) Eskeşehir: Anadolu Üniversitesi.

Aydıntan, B. (2009). *Örgüt Zekası ve Yönetimi*, Ankara:Gazi Kitabevi.

Barutçugil, İ. (2002). *Bilgi Yönetimi*, İstanbul: Kariyer Yayıncılık İletişim.

Baykal, A. (2006). *Veri Madenciliği Uygulama Alanları*, D.Ü. Ziya Gökalp Eğitim Fakültesi Dergisi, 7, s.95-107.

Çokluk, Ö. Ve Kayrı, M. (2011). *Kayıp Değerlere Yaklaşık Değer Atama Yöntemlerinin Ölçme Araçlarının Geçerlik ve Güvenirliği Üzerindeki Etkisi*, Kuram ve Uygulamada Eğitim Bilimleri, 11(1), s.289-309.

Dondurmacı, G.A. ve Çınar, A. (2014). *Finans Sektöründe Veri Madenciliği Uygulaması*, Akademik Sosyal Araştırmalar Dergisi, Yıl:2, 2(1), s.258-271.

Han, J. ve Kamber, M. (2012). *Data Mining Concepts and Techniques*, Third Edition, U.S.A: Morgan Kaufmann Academic Press Elsevier Inc.

Kaya, H. ve Köyemen, K. (2008). *Veri Madenciliği Kavramı ve Uygulama Alanları*, Doğu Anadolu Bölgesi Araştırmaları; 2008, s.159-164.

Koyuncugil, A.S. ve Özgülbaş, N. (2009), *Veri Madenciliği: Tip ve Sağlık Hizmetlerinde Kullanımı ve Uygulamaları*, Bilişim Teknolojileri Dergisi, 2 (2), s.21-32

Laudon, K.C. Laudon J.P. (2011). *Yönetim Bilişim Sistemleri Dijital İşletmeye Yönetme*, Çeviri Editörü: U.Yozgat, Ankara: Nobel Akademik Yayıncılık.

Öğüt, A. (2003). *Bilgi Çağında Yönetim*, Ankara: Nobel Yayınevi Dağıtım.

Özbay, Ö. (2015). *Veri Madenciliği Kavramı ve Eğitimde Veri Madenciliği Uygulamaları*, Uluslararası Eğitim Bilimleri Dergisi, Yıl:2, (5), s.262-272.

Özkan, M. ve Boran, L. (2014). *Veri Madenciliğinin Finansal Kararlarda Kullanımı*, Çankırı Karatekin Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, 4 (1), s.59-82.

- Savaş, S., Topaloğlu, N. ve Yılmaz, M. (2012). *Veri Madenciliği ve Türkiye'deki Uygulama Örnekleri*, İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi, Yıl:11, (21), s.1-23.
- Silahtaroğlu, G. (2013). *Veri Madenciliği Kavram ve Algoritmaları*, İstanbul: Papatya Yayıncılık Eğitim.
- Tüzüntürk, S. (2010). *Veri Madenciliği ve İstatistik*, Uludağ Üniversitesi, İktisadi ve İdari Bilimler Fakültesi Dergisi, XXIX (1), s.65-90.

Yararlanılan Internet Kaynakları

- http://www.chip.com.tr/haber/yapay-zekanin-gizemleri-sanal-noron-perceptron_36078_5.html (erişim tarihi: 20.05.2016)
- <http://www.gurunlu.com/dokumanlar/dm-yasam-dongusu.pdf> (erişim tarihi: 20.05.2016)
- http://www.tdk.gov.tr/index.php?option=com_gts&arama=gts&gucid=TDK.GTS.56ddc1d2128bf9.23947232 (erişim tarihi: 15.03.2016)
- http://www.sertacogut.com/blog/wp-content/uploads/2009/03/sertac_ogut_-_veri_madenciligi_kavrami_ve_gelisim_sureci.pdf (Öğüt, S., Veri Madenciliği Kavramı ve Gelişim Süreci, erişim tarihi: 15.03.2016)

2

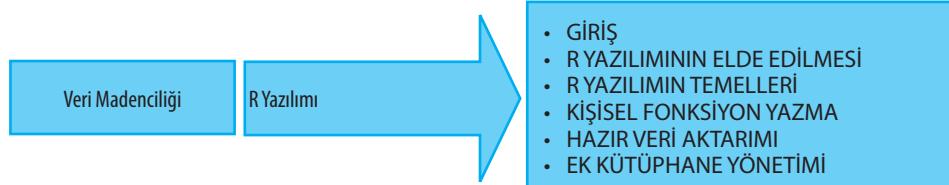
Amaçlarımız

- Bu üniteyi tamamladıktan sonra;
- 🕒 R yazılımında temel komutları kullanabilecek,
 - 🕒 R yazılımında kişisel fonksiyon oluşturabilecek,
 - 🕒 R yazılımında ek kütüphane kullanabilecek bilgi ve becerilerine sahip olabileceksiniz.

Anahtar Kavramlar

- Temel Komutlar
- Matrişler
- Mantık Operatörleri
- List Nesneleri
- Data Frame
- Kişisel Fonksiyon
- Ek Kütüphane

İçindekiler

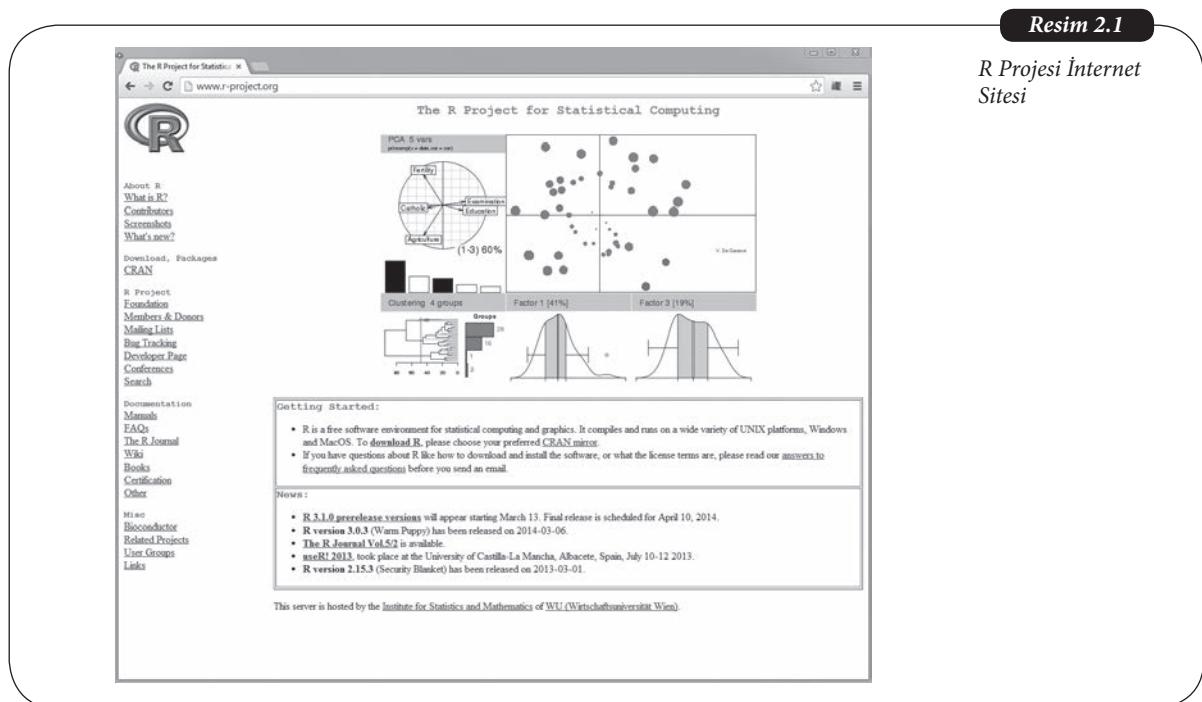


R Yazılımı

GİRİŞ

R yazılımı İnternet aracılığı ile ücretsiz olarak dağıtılan genel lisanslı bir programdır. Yazılım, lisans kapsamında serbest bir şekilde dağıtilabilir ve kullanılabilir. Ayrıca yazılımı elde eden herkes asıl kaynağı belirterek dağıtmaya ve kullanma hakkına sahiptir. Yazılımın kaynak kodu da açık bir şekilde sunulmaktadır. Dolayısıyla herhangi bir programlama bilgisine sahip kişiler bu kod üzerinde değişiklikler ve geliştirmeler yapma hakkına sahiptir. Yazılımın en büyük üstünlüklerinden biri de hemen hemen bütün işletim sistemlerinde çalışabiliyor olmasıdır. R yazılımı kullanılarak, istatistiksel analiz, grafik çizme ve veri işleme işlemleri yapılabilir. Temel olarak R, Becker and Chambers tarafından geliştirilen S dilinin bir çeşididir. S dili daha sonra S-Plus paket programı hâline dönüşerek ticari bir marka hâline gelmiştir. Günlük bir kullanıcı kolaylıkla bu iki dil arasında geçiş yapabilmektedir. R ise, R yazılımı adı altında paket program hâline gelmiştir. Resim 2.1'de R projesi İnternet sitesinin görüntüsü yer almaktadır.

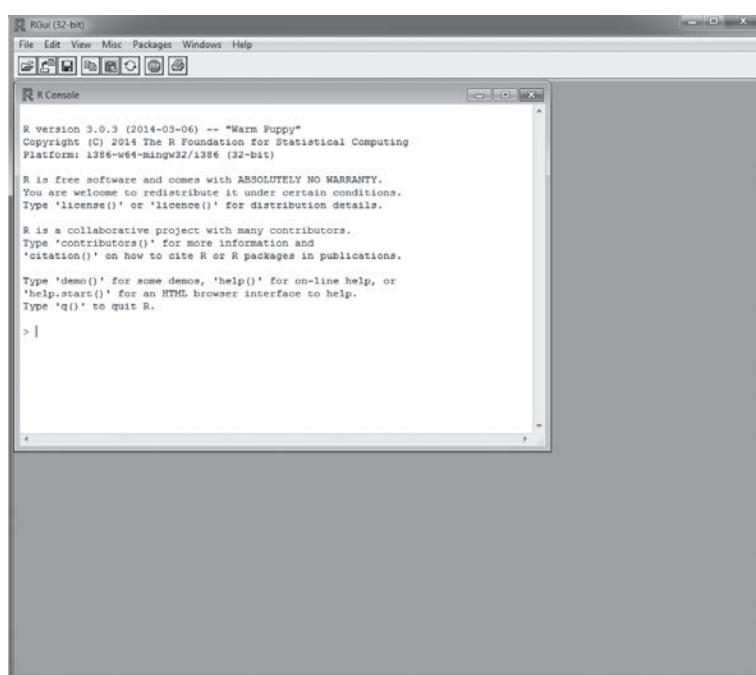
Resim 2.1



1997 yılından bu güne toplam 17 kişilik bir ekip R dilinin ve R yazılımının geliştirilmesindeki temel grubu oluşturmaktadır. R yazılımı, kullanılan işletim sistemine göre <http://www.r-project.org> adresinden ücretsiz olarak temin edilebilmektedir. Belki de ürün için yapılacak tek ödeme evlerden yapılacak olan İnternet bağlantı ücretinden ileriye geçmeyecektir. Bu temel ekip dışında dünya çapında özellikle istatistik teknikleri üzerinde çalışma yapan bilim adamları R dilinin ve R yazılımının gelişmesine büyük bir katkı sağlamaktadırlar. R yazılımı İnternet sitesinde günlük kullanıcılar tarafından tüm dünya bilim adamlarının kullanımına olanak sağlayacak yaklaşık 600 ek kütüphane bulunmaktadır. Ayrıca birçok bilimsel dergide yayınlanan makalelerde R yazılımı kaynak kodu ile beraber gelmektedir. R yazılımı ilk bakışta kod yazımı tabanlı olduğu için kullanıcılarla zor gelebilmektedir. Fakat 15 dakikalık kısa bir çalışma, daha önce programlama dilleri ile herhangi bir etkileşimi olmayan kişilerin bile kolaylıkla veri girişi yapabilmesine, çeşitli grafik ve istatistiksel analizler yapabilmesine olanak vermektedir. R yazılımı uygun işletim sistemine göre yüklenildikten sonra kolaylıkla çalıştırılabilir ve makalelerde R yazılımının kullanımı, hem de bu yazılımın içerisinde yer alan fonksiyonların detayları hakkında yardım dosyaları da gelmektedir. Özellikle kullanım kılavuzları kişilerin baskısını alıp çalışabilecekleri PDF dosyaları olarak gelmektedir. Kullanıcıların dile kolaylıkla alışabilmeleri için yardım dosyaları bir çok örnek ile sunulmaktadır. Ayrıca daha ileri problemlerin çözümünde yine isteyen herkesin katılabileceği 4 ayrı tartışma liste bulunmaktadır. Resim 2.2'de Microsoft Windows 7 işletim sistemi altında derlenerek çalıştırılan R for Windows 3.0.3 yazılımı arayüzü görülmektedir.

Resim 2.2

R for Windows 3.0.3
Yazılımı Arayüzü



R yazılımı çevre birimi kullanıcılarla etkin bir veri işleme ve depolama olağlığı, dizi ve matris hesaplamaları için komutlar grubu, veri analizi için ileri düzeyde teknikler topluluğu, verinin ekranda ya da basılı bir eserde görüntülenebilmesine olanak veren geniş grafiksel özellikler, kolay programlamaya uygun fakat karmaşık programlama dilinin özelliklerine sahip bir programlama dilinin olanaklarını sunmaktadır.

R YAZILIMININ ELDE EDİLMESİ

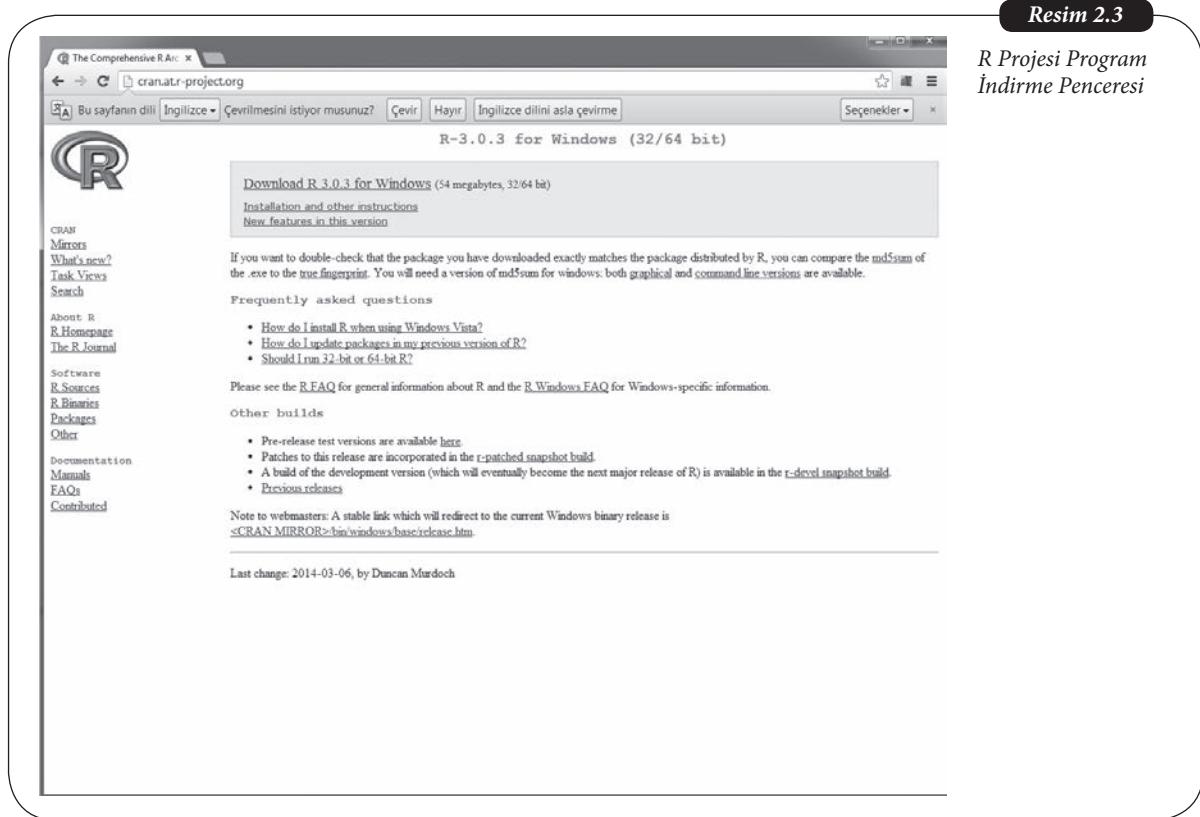
R istatistiksel bilgisayar yazılımı İnternet aracılığı ile dağıtılmaktadır. İsteyen kullanıcılar programın ana sitesini kullanarak ücreti karşılığında CD üzerinden de programı elde edebilmektedirler. Programın lisansı, genel kullanıcı lisansı türündendir. Bu lisans, kullanıcılarla ellişinde bulundurdukları programı serbestçe dağıtmaya ve kullanma hakkını vermektedir. Sadece bu lisans kapsamında ürünü elde eden kişiler aynı hakka sahip olabilmektedir. Ayrıca kullanıcılar kaynak kodun kendisini de ücretsiz olarak elde edebilmektedirler. Microsoft Windows, Linux ve Unix sistemleri ve Apple MacOS işletim sistemleri için çeşitli sürümler elde edilebilmektedir.

İnternet erişimine sahip olan kullanıcılar için programın elde edilmesi oldukça kolay bir iştir. Ana siteden indirilebilecek bir program gereklı yüklemeye işlemlerini otomatik olarak yapacaktır. Kullanıcı programı ilk çalıştırduğunda karşısına çıkacak olan pencere sistemindeki menüleri kullanarak programın yeteneklerini dünyaki diğer kullanıcıların geliştirdikleri alt programları da yükleyerek genişletebilirler.

R istatistiksel bilgisayar yazılımını İnternet aracılığıyla, kullanılan bilgisayara yükleme işlemi aşağıdaki gibi gerçekleştirilebilir. Burada dikkat edilecek nokta; işlemlerin Microsoft Windows işletim sistemi için yapılacağıdır. Farklı işletim sistemleri için izlenecek adımlar ana sitede yer almaktadır.

Öncelikle R Project İnternet sitesine <http://www.r-project.org> adresinden erişilebilir. Bu ana sitede R yazılımının tarihçesi ile ilgili bir çok bilgiye ulaşmak mümkündür. Bu sitede daha sonra sol kısmında yer alan menüde “Download” bölümünde bulunan “CRAN” (Comprehensive R Archive Network) linkine ulaşılır.

Resim 2.3



Windows işletim sistemi için derlenmiş program "Download R for Windows" linkinde yer almaktadır.

Resim 2.3'te R projesi program indirme penceresi görülmektedir. Daha sonra bu sitede yer alan işletim sistemlerine göre derlenmiş sürümlerin bulunduğu bölüme ulaşılır. İsteyen kullanıcılar programın kaynak kodunu da buradan alarak kendileri derleyebilirler. Bu ünitede Microsoft Windows için derlenmiş program kullanılacaktır. Önceden derlenmiş dağıtımlar listesinden "**Download R for Windows**" linki seçilir. Yeni gelen ekranda öncelikle ana programı elde etmek için "base" linki seçilir. Bu linkin seçilmesiyle hazır durumda sorunsuz olarak çalışan en son sürüm programı elde edilebilir (Resim 2.3). Sayfada yer alan "Download R 3.0.3 for Windows (54 megabites, 32/64 bit)" linki, istenilen programı içeren dosyadır. Bu dosyayı bilgisayarınızdaki sabit sürücüye indirmek gerekmektedir. Buradaki önemli husus; dosyanın büyüklüğüdür (ortalama 54 MB). İndirilen dosya çalıştırılarak bilgisayar kurulum işlemi tamamlanmış olur. Microsoft Windows başlangıç menüsüne R menüsü, yükleme sırasında eklenmektedir. Programın ilk kez çalıştırılmasıyla Resim 2.2'de yer alan R for Windows 3.0.3 yazılımı arayüzü elde edilir.

R YAZILIMIN TEMELLERİ

İlk olarak, R yazılımının bir veri işleme ve grafik çizme programı olduğu unutulmamalıdır. Microsoft Windows versiyonu bir pencere içinde kullanıcının gerekli komutları interaktif olarak ele almasına olanak tanımaktadır. Her ne kadar R yazılımı için Minitab ve SPSS benzeri menü sistemi geliştirilmeye çalışılsa da temel olarak yapılması gereken, komutların yazılıarak çıktıların görüntülenmesi işlemidir. Resim 2.2'de komutların girilmesi için bir bölge bulunmaktadır. Bu bölge "**R Console**" olarak adlandırılmıştır. İzleyen alt bölümlerde sıradan bir kullanıcının başlangıçta ihtiyaç duyacağı bazı temel komutlar ve tanımlar verilmiştir. Komut satırlarında yer alan > işaretti komut satırının kendisini temsil etmektedir.

Temel Komutlar

Herhangi bir atama yapılması ya da matematiksel bir ifadenin hesaplanması için en basit komutlar olarak meydana çıkan komutlar grubuna **temel komutlar** denir.

Herhangi bir atama yapılması ya da matematiksel bir ifadenin hesaplanması için en basit komutlar olarak meydana çıkan komutlar grubuna **temel komutlar** denir.

komutu yazilarak Enter'a basıldığında

```
[1] 117
```

sonucu ekranda görüntülenecektir. Matematiksel işlemin hemen sonucunu elde etmek yerine sonuçlar herhangi bir değişkene de atanabilir. Bu atama işlemi için "değişken <- işlem" yapısı kurulmalıdır. Örneğin önceki toplam x gibi bir değişkene atanmak istenirse

```
> x <- 72+45
```

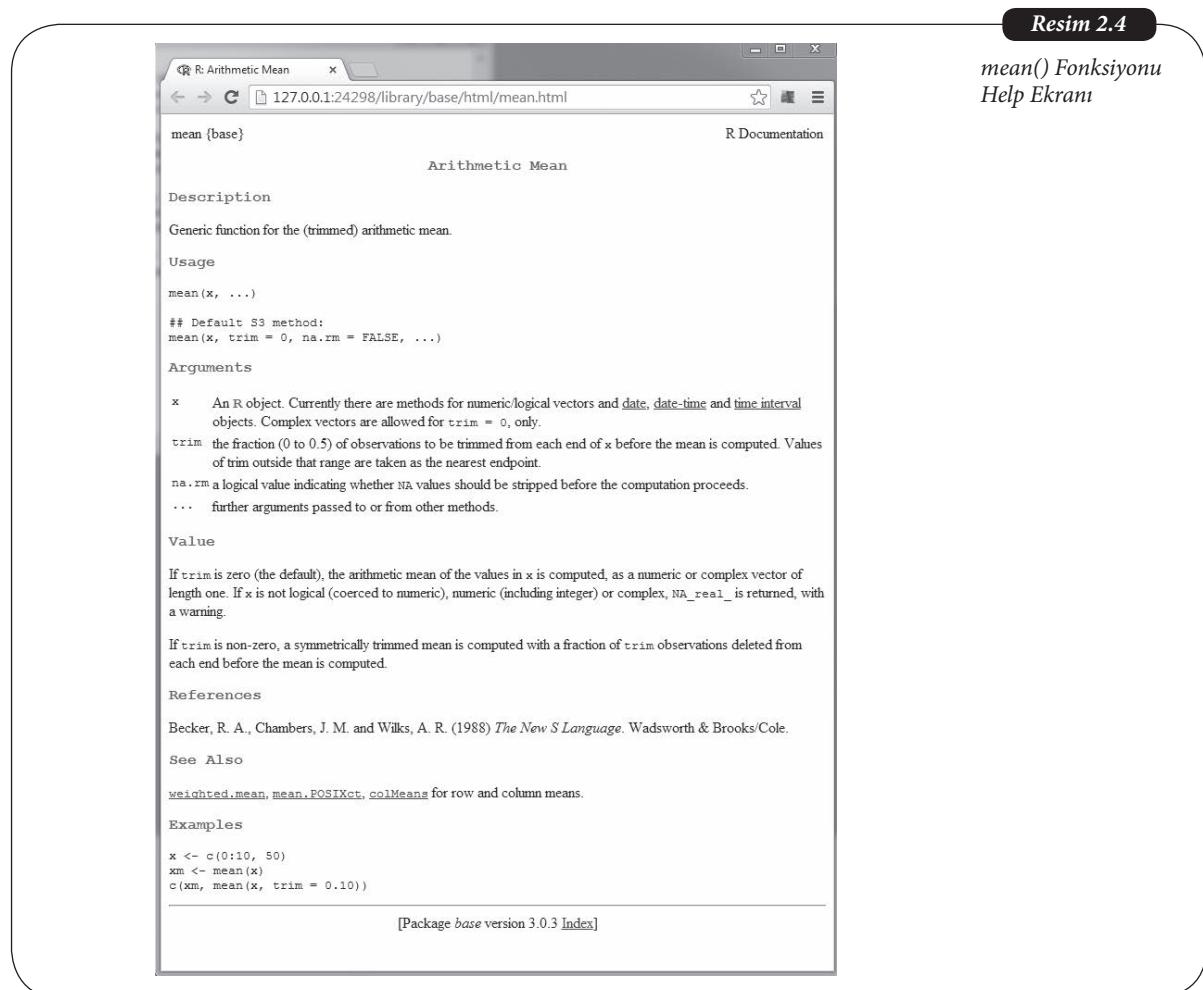
komutunun verilmesi yeterli olacaktır. Yeni bir atama yapılmadığı sürece x değişkeni bu toplamın sonucundan oluşacaktır.

R ile çalışırken herhangi bir fonksiyon ya da kitaplık hakkında yardım almanın iki yolu vardır. Öncelikle aritmetik ortalama hesabında kullanılan mean() komutunu bildiğimizi varsayıyalım. Bu fonksiyonun hangi parametreleri aldığı ve diğer ayrıntıları görebilmek için

```
help(mean)
```

komutunun verilmesi yeterli olacaktır. Bu komut sonucunda gelecek ekranda fonksiyon ile ilgili ayrıntılar bulunabilecektir (Resim 2.4).

Resim 2.4



Yardım almanın diğer bir yolu ise R arayüzü help menüsünün kullanılmasıdır. Bu menü sayesinde hem basit kullanım kılavuzlarına ulaşılabilimekte hem de yardım sayfalarında arama yapılmaktadır.

Vektörler

R yazılımının en büyük özelliklerinden biri de değişkenler ile çalışırken vektör ve matris kullanımına olanak tanımıştır. En basit şekilde bir vektörü oluşturabilmek için **c() fonksiyonu** kullanılmaktadır. Daha önce kullanılan x değişkenini 5 birimlik bir vektör hâline dönüştürme işlemi ve sonucu aşağıda verilmiştir.

```
> x <- c(1,2,3,4,5)
> x
[1] 1 2 3 4 5
```

Gördüğü gibi burada ilk satırda x vektörüne 5 adet değer atanmakta ikinci satırda ise x'e ataması yapılan değerlerin görüntülenmesi komutu verilmektedir. Bu noktada önemli olan konu; R yazılımının büyük ve küçük harfe olan duyarlılığıdır. X ve x değişkenleri tamamen farklı değişkenlerdir.

Önceden tanımlanmış bir vektörün birim sayısını öğrenmek için **length() fonksiyonu** kullanılır. Örnekteki birim sayısı aşağıdaki gibi öğrenilebilir.

Bir vektörü en basit şekilde yaratmak için **c() fonksiyonu** kullanılır.

Önceden tanımlanmış bir vektörün birim sayısını öğrenmek için **length() fonksiyonu** kullanılır.

```
> length(x)
[1] 5
```

c() fonksiyonu karakter değişkenleri yaratmak için de kullanılır.

ÖRNEK 1

4 isimden oluşan isim değişkenini c() fonksiyonunu kullanarak oluşturunuz.

4 isimden oluşan isim değişkeni aşağıdaki gibi oluşturulabilir.

```
> isim <- c("Defne", "Kuzey", "Alara", "Miray")
> isim
[1] "Defne" "Kuzey" "Alara" "Miray"
```

Ayrıca, c() fonksiyonu birden fazla vektörün tek bir vektör olarak birleştirilmesinde ya da karakter değişkeninin sayılarla birleştirilmesinde de kullanılabilir.

SIRA SİZDE



c() fonksiyonunu kullanarak döviz değişkeninde 3 farklı döviz ismi oluşturunuz.

ÖRNEK 2

Örnek 1'deki isim değişkenine 17, 22, 45 rakamlarını ekleyerek yenix değişkenini oluşturunuz.

İsim değişkenine 17, 22, 45 rakamlarını ekleyerek yenix değişkenini oluşturmak için aşağıdaki işlemleri yapılmalıdır.

```
> yenix <- c(isim, 17, 22, 45)
> yenix
[1] "Defne" "Kuzey" "Alara" "Miray" "17" "22" "45"
```

Belirli bir düzene sahip olan vektörlerin oluşturulmasında da seq() fonksiyonu kullanılır. Bu fonksiyonun genel yazılımı seq(altnumber, number, artısmak) şeklindedir.

ÖRNEK 3

Sıfırdan sekize kadar 1'er artan rakamlardan oluşan vektörü oluşturunuz.

Sıfırdan sekize kadar 1'er artan rakamlardan oluşan vektör için aşağıdaki işlemleri yapılır.

```
> seq(0, 8, 1)
[1] 0 1 2 3 4 5 6 7 8
```

ÖRNEK 4

Dörtten on altiya kadar 4'er artan rakamlardan oluşan vektörü oluşturunuz.

Dörtten on altiya kadar 4'er artan rakamlardan oluşan vektör için aşağıdaki işlemleri yapılır.

```
> seq(4, 16, 4)
[1] 4 8 12 16
```

ÖRNEK 5

Sırasıyla sıfırdan dörde kadar 1'er artan ve sıfırdan ona kadar 2'ser artan rakamlardan oluşan vektörü oluşturunuz.

Sırasıyla sıfırdan dörde kadar 1'er artan ve sıfırdan ona kadar 2'ser artan rakamlardan oluşan vektör için aşağıdaki işlemleri yapılır.

```
> y (seq(0, 4, 1), seq (0, 10, 2))
[1] 0 1 2 3 4 0 2 4 6 8 10
```

Belirli bir düzene sahip verilerin oluşturulması için **rep()** **fonksiyonu** kullanılır. Örneğin istatistikteki varyans analizi işlemi gerçekleştirebilmek için ilgilenilen değişkenin her seviyesi için birim sayısı kadar isim girilmesi gerekmektedir. Komut, rep (istenen-duzen,tekrar-sayıısı) şeklinde yapılandırılır.

Belirli bir düzene sahip verilerin oluşturulması için **rep()** **fonksiyonu** kullanılır.

8 adet 3 rakamını içeren vektörü oluşturunuz.

ÖRNEK 6

8 adet 3 rakamını içeren vektörü oluşturmak için aşağıdaki işlemler yapılır.

```
> rep(3,8)
[1] 3 3 3 3 3 3 3 3
```

1'den 5'e kadar olan rakamları 4 tekrar olacak biçimde içeren vektörü oluşturunuz.

ÖRNEK 7

1'den 5'e kadar olan rakamları 4 tekrar olacak biçimde içeren vektör aşağıdaki işlemler yardımıyla oluşturulur.

```
> rep(1:5,4)
[1] 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5
```

1'den 6'ya kadar olan rakamları 2 tekrar olacak biçimde içeren vektörü oluşturunuz.

ÖRNEK 8

1'den 6'ya kadar olan rakamları 2 tekrar olacak biçimde içeren vektör aşağıdaki işlemler yardımıyla oluşturulur.

```
> rep(seq(1,6),2)
[1] 1 2 3 4 5 6 1 2 3 4 5 6
```

İzleyen örneklerde seq() ve rep() fonksiyonları yardımıyla çok daha kompleks yapılmış vektörler oluşturulmuştur. Bu fonksiyonların kullanımı pratik yapıldıkça daha da kolaylaşmaktadır.

1'den 5'e kadar ve her birinden beş adet olacak biçimde rakamlardan oluşan vektörü oluşturunuz.

ÖRNEK 9

1'den 5'e kadar ve her birinden beş adet olacak biçimde rakamlardan oluşan vektörü oluşturmak için aşağıdaki işlemler yapılır.

```
> rep(seq(5),rep(5,5))
[1] 1 1 1 1 1 2 2 2 2 3 3 3 3 4 4 4 4 4 5 5 5 5
```

1'den 6'ya kadar her birinden kendi sayısı kadar olacak biçimde rakamlardan oluşan vektörü oluşturunuz.

ÖRNEK 10

1'den 6'ya kadar her birinden kendi sayısı kadar olacak biçimde rakamlardan oluşan vektörü oluşturmak için aşağıdaki işlemler yapılır.

```
> rep(seq(6),c(1,2,3,4,5,6))
[1] 1 2 2 3 3 4 4 4 4 5 5 5 5 6 6 6 6 6
```

Bir vektörün herhangi bir elemanına ulaşmak mümkündür. Bu da köşeli parantezler içinde istenilen vektör elemanın sıra numarası verilerek yapılabilir.

ÖRNEK 11

Bir değişkene altı tane birim atayarak bu birimlerden dördüncüsünün değerini görüntüleyiniz. Daha sonra da birinci ve altıncı birimlerin değerlerini gösteriniz.

```
> isim<-c("Ali", "Can", "Nedim", "Ümit", "Fuat", "Volkan")
> isim[4]
[1] "Ümit"
> isim[c(1,6)]
[1] "Ali" "Volkan"
```

Bir değişkenin karekökü **sqrt()** fonksiyonu yardımıyla hesaplanabilir.

Oluşturulan vektörler arasındaki tüm aritmetik işlemler de kolaylıkla yapılabilir. Bir değişkenin karekökü **sqrt()** fonksiyonu yardımıyla hesaplanabilir.

ÖRNEK 12

5 öğrencinin yaşlarından oluşan bir değişkende her bir yaşın karekökünü hesaplayınız.

Bu işlem için öncelikle yaşları değişkeni oluşturulur. Daha sonra bu değişkenin karekökü hesaplanır. Böylece her bir yaşa ilişkin değerin karekökü alınmış olur.

```
> yaşlar<-c(21,24,22,21,25)
> sqrt(yaşlar)
[1] 4.582576 4.898979 4.690416 4.582576 5.000000
```

Matrisler

R yazılımında matris oluşturmak için **matrix()** fonksiyonu kullanılır.

Birçok araştırmada, yapılan analizler sırasında matris oluşturulması gerekmektedir. R yazılımında matris oluşturmak için **matrix()** fonksiyonu kullanılır. Bu fonksiyonun genel yazımı;

```
matrix(veri, nrow(satırsayı), ncol(sütünsayı),
      byrow=F(veri sütun olarak girilsin))
```

şeklindedir. 2 değişken ve 6 gözlem değerinden oluşan veri seti için iki sütun ve altı satırlık bir matris oluşturalım. Veriyi hem **matrix()** komutu içerisinde hem de bir değişken kullanarak atayalım.

Veri, **matrix()** komutu içinde aşağıdaki gibi oluşturulabilir.

```
> matrix(c(6,5,4,3,2,1,1,2,3,4,5,6),ncol=2)
      [,1] [,2]
[1,]    6    1
[2,]    5    2
[3,]    4    3
[4,]    3    4
[5,]    2    5
[6,]    1    6
```

Veri, bir değişken kullanılarak aşağıdaki gibi oluşturulabilir.

```
> veri<-c(6,5,4,3,2,1,1,2,3,4,5,6)
> matrix(veri,ncol=2)
[,1] [,2]
[1,] 6 1
[2,] 5 2
[3,] 4 3
[4,] 3 4
[5,] 2 5
[6,] 1 6
```

Eğer öncelikle satırlar veri kısmında tanımlanıyor ise byrow=T parametresinin kullanılması gerekmektedir. Yukarıdaki örnek için tanımlanan veri değişkeninin aslında satırlar olarak oluşturulduğu varsayıımı altında byrow=T parametresi yardımıyla,

```
> matrisim<-matrix(veri,ncol=2,byrow=T)
> matrisim
[,1] [,2]
[1,] 6 5
[2,] 4 3
[3,] 2 1
[4,] 1 2
[5,] 3 4
[6,] 5 6
```

matrisi elde edilecektir. Daha önce tanımlanan köşeli parantezler kullanılarak, matrisin herhangi bir elemanına ait olan satır ve sütun sayısı girilerek ulaşılabilir. Örneğin matrisin ikinci satır, ikinci sütun elemanı,

```
> matrisim[2,2]
[1] 3
```

olarak elde edilecektir. Matrisin herhangi bir satırına [satırno,] ve herhangi bir sütununa da [,sütunno] köşeli parantezleri yardımıyla ulaşılabilir. Örnek matrisimizin birinci satırı ve ikinci sütunu;

```
> matrisim[1,]
[1] 6 5
> matrisim[,2]
[1] 5 3 1 2 4 6
```

olarak görüntülenecektir. Aritmetik işlemler de matrisler üzerinde kolaylıkla uygulanabilir.

```
> matris2<-matrix(c(2,7,1,4),ncol=2,byrow=T)
> matris2
[,1] [,2]
[1,] 2 7
[2,] 1 4
> matris2*matris2
[,1] [,2]
[1,] 4 49
[2,] 1 16
> matris2+matris2
[,1] [,2]
[1,] 4 14
[2,] 2 8
```

Dikkat edilirse örnekte, çarpma işlemi matris elamanları için bire bir gerçekleşmiştir. İki matrisin çarpılması ise "%*%" ile gerçekleştirilir.

```
> matris2 * matris2
[,1] [,2]
[1,] 4 49
[2,] 1 16
> matris2 %*% matris2
[,1] [,2]
[1,] 11 42
[2,] 6 23
```

Matrisin evriğinin elde edilmesi için t() fonksiyonu kullanılır.

```
> matris2
[,1] [,2]
[1,] 2 7
[2,] 1 4
> t(matris2)
[,1] [,2]
[1,] 2 1
[2,] 7 4
> matris2 %*% t(matris2)
[,1] [,2]
[1,] 53 30
[2,] 30 17
```

Mantık Operatörleri

R yazılımı ile mantık tipi değişkenlerin oluşturulması mümkündür. Doğru (T) ve Yanlış (F) olmak üzere iki mantıksal değer vardır. Çeşitli operatörler ve anlamları Tablo 2.1'de sunulmuştur.

Operatör	Kullanımı
<	Küçüktür
>	Büyükür
<=	Küçük ya da eşittir
>=	Büyük ya da eşittir
==	Eşittir
!=	Eşit değildir
&	Ve
	Veya
!	Değil

Tablo 2.1
Mantıksal Operatörler

Mantık operatörleri, karşılaştırma yaparken ve vektörler ile matrislerin belirli elemanlarını belirlerken çok kullanışlıdırlar. Bir mantık operatörü yardımıyla bir değişkene Doğru (T) ya da Yanlış (F) değeri atamasını sağlanabilir.

```
> değişken<-23 == 4
> değişken
[1] FALSE
```

Burada değişkenin değeri olan 23'ün 4'e eşit olup olmaması durumu mantık değeri olarak tekrar değişkene atanmıştır. Artık değişkenimizin değeri FALSE, yani yanlıştır. Benzer şekilde doğru bir sonuçta elde edebiliriz.

```
> değişken<-123 < 514
> değişken
[1] TRUE
```

R yazılımında mantık operatörlerine ek olarak **mantık fonksiyonları** da sunmaktadır. Bu fonksiyonlar yardımı ile ilgilenilen değişkenin bir karakter değişkeni mi yoksa sayısal bir değişken mi olduğunu anlama şansına sahip olunabilir. Bununla ilgili birkaç örnek aşağıda verilmiştir.

Mantık fonksiyonları yardımı ile ilgilenilen değişkenin bir karakter değişkeni mi yoksa sayısal bir değişken mi olduğu anlaşılabılır.

```
> is.character("Kuzey")
[1] TRUE
> is.character("1")
[1] TRUE
> is.character(1)
[1] FALSE
> 25/0
[1] Inf
> is.numeric(25/0)
[1] TRUE
> is.infinite(25/0)
[1] TRUE
> is.numeric(27)
[1] TRUE
```

şeklinde verilebilir.

Bir vektörden çeşitli elemanların elde edilmesinde mantık vektörü kullanılabilir.

ÖRNEK 13

23, 12, 43, 23, 11, 24, 21, 17, 15, 14 rakamlarından oluşan x vektörünü önce küçükten büyüğe doğru sıralayınız, daha sonra 20'den büyük değerleri elde ediniz.

23, 12, 43, 23, 11, 24, 21, 17, 15, 14 rakamlarından oluşan x vektörü küçükten büyüğe doğru aşağıdaki gibi sıralanabilir.

```
> x<-c(23,12,43,23,11,24,21,17,15,14)
> sort(x)
[1] 11 12 14 15 17 21 23 23 24 43
```

x vektöründe 20'den büyük değerleri elde etmek için aşağıdaki işlem yapılır.

```
> x[x>20]
[1] 23 43 23 24 21
```

Bir başka kullanım yönteminde ise herhangi bir değişkenin değerleri bir başka değişkenin eşli değerlerine bağlı olarak elde edilebilir. Örneğin; A değişkeni bir araştırmada 10 birimlik birim sıra numarasını verirken, B değişkeni de ölçüm değerleri olsun. 25'ten büyük ölçüm değerlerine sahip birimlerin sıra numaraları

```
> A<-c(1:10)
> A
[1] 1 2 3 4 5 6 7 8 9 10
> B<-c(28,13,35,73,12,12,98,34,26,10)
> A[B>25]
[1] 1 3 4 7 8 9
```

olarak hesaplanır.

List Nesneleri

Çeşitli istatistiksel analizler için oluşturulan farklı nesnelerin bir araya getirilmesinde **List Nesnelerinden** faydalansılır. Örneğin; ilgilenilen veri kümesi ile bunlara ait korelasyon matrisi aynı nesne içerisinde görüntülenebilir (ya da hafızada birlikte saklanması sağlanabilir).

İzleyen kodlamada bir veri seti oluşturularak korelasyon matrisi ile birlikte tek bir veri seti hâline getirilmiştir.

```
x<-cbind(c(3,2,4,5,6,7,6,1,3,8),c(4,7,9,2,4,
+ 6,2,5,7,8),c(4,5,3,2,5,6,7,8,9,1))
>x [,1] [,2] [,3]
[1,] 3 4 4
[2,] 2 7 5
[3,] 4 9 3
[4,] 5 2 2
[5,] 6 4 5
[6,] 7 6 6
[7,] 6 2 7
[8,] 1 5 8
[9,] 3 7 9
[10,] 8 8 1
> korx<-cor(x)
> korx
      [,1]      [,2]      [,3]
[1,] 1.00000000 -0.06077558 -0.4543695
[2,] -0.06077558  1.00000000 -0.1248409
[3,] -0.45436947 -0.12484087  1.0000000
> birlikte<-list(x,korx)
> birlikte
[[1]]
      [,1] [,2] [,3]
[1,] 3 4 4
[2,] 2 7 5
[3,] 4 9 3
[4,] 5 2 2
[5,] 6 4 5
[6,] 7 6 6
[7,] 6 2 7
[8,] 1 5 8
[9,] 3 7 9
[10,] 8 8 1
[[2]]
      [,1]      [,2]      [,3]
[1,] 1.00000000 -0.06077558 -0.4543695
[2,] -0.06077558  1.00000000 -0.1248409
[3,] -0.45436947 -0.12484087  1.0000000
```

Kullanıcı artık hem veriyi hem de ilgili korelasyon matrisini birlikte görebilmektedir. Bu yeni oluşturulan değişken içerisindeki korelasyon matrisi kullanılmak istenirse, List Nesnesi içerisindeki sıra numarası iki köşeli parantezde olacak şekilde tanımlanır. Örnekte, korelasyon matrisi 2 numaralı liste elemanı olduğu için “birlikte” değişkeninin korelasyon elemanı

```
> birlikte[[2]]
      [,1]     [,2]     [,3]
[1,] 1.00000000 -0.06077558 -0.4543695
[2,] -0.06077558 1.00000000 -0.1248409
[3,] -0.45436947 -0.12484087 1.0000000
```

olarak görüntülenecektir. List değişkenleri oluşturulurken bu değişken içerisinde yer alan elemanlar isimleriyle de atanabilir. Örnekte ilk List elemanı “veri” ikinci List elemanında “korelasyon” olarak adlandırılabilir. Bu işlem için,

```
> birlikte<-list(veri=x,korelasyon=korx)
> birlikte
$veri
      [,1] [,2] [,3]
[1,]  3   4   4
[2,]  2   7   5
[3,]  4   9   3
[4,]  5   2   2
[5,]  6   4   5
[6,]  7   6   6
[7,]  6   2   7
[8,]  1   5   8
[9,]  3   7   9
[10,] 8   8   1
$korelasyon
      [,1]     [,2]     [,3]
[1,] 1.00000000 -0.06077558 -0.4543695
[2,] -0.06077558 1.00000000 -0.1248409
[3,] -0.45436947 -0.12484087 1.0000000
```

kodlaması gerçekleştirilir. Değişken içerisinde yer alan her bir eleman için verilen isimler, daha sonra bu elemanların çağrıması işleminde de kullanılabilir. List değişken adından sonra ilgilenilen eleman adı \$ işaretini yardımıyla

```
> birlikte$korelasyon
      [,1]     [,2]     [,3]
[1,] 1.00000000 -0.06077558 -0.4543695
[2,] -0.06077558 1.00000000 -0.1248409
[3,] -0.45436947 -0.12484087 1.0000000
```

olarak elde edilebilir.

Data Frame

R yazılımında veri seti içerisindeki faktör listeleri ve gözlem birimleri **data frame** olarak bir araya getirilirler.

Bir çok araştırmada ilgilenilen değişkenin çeşitli seviyeleri ve bu seviyeler için gözlem değerleri bulunmaktadır. R yazılımında veri seti içerisindeki faktör listeleri ve gözlem birimleri data frame olarak bir araya getirilirler. “**data.frame**” fonksiyonunda her sütunda eşit sayıda birim yer almaktadır. Her satır bir gözlem birimini temsil etmektedir. Örneğin; 8 adet öğrencinin 4 farklı dersten aldığıları başarı puanları bir değişkende bir araya getirilsin.

```
> betimsel<-c(23,54,65,13,87,56,34,76)
> karar<-c(27,65,46,14,96,68,46,64)
> bilgisayar<-c(45,25,12,21,42,32,14,54)
> matris<-c(34,65,76,12,37,83,90,48)
> isim<-c("Tuncay", "Serhat", "Volkan",
+ "Rüştü", "Ümit", "Önder", "Zafer", "Selçuk")
> öğrencinot<-data.frame(isim,betimsel,
+ karar,bilgisayar,matris)
> öğrencinot
```

	isim	betimsel	karar	bilgisayar	matris
1	Tuncay	23	27	45	34
2	Serhat	54	65	25	65
3	Volkan	65	46	12	76
4	Rüştü	13	14	21	12
5	Ümit	87	96	42	37
6	Önder	56	68	32	83
7	Zafer	34	46	14	90
8	Selçuk	76	64	54	48

List nesnelerinde olduğu gibi istenen sütun \$ işaretini yardımcıyla görüntülenebilir.
Örneğin; öğrencilerin betimsel istatistik notları için tanımlayıcı istatistikler

```
> summary(öğrencinot$betimsel)
Min. 1st Qu. Median Mean 3rd Qu.
13.00 31.25 55.00 51.00 67.75

Max.
87.00
```

olarak elde edilebilir. Eğer herhangi bir gözlem biriminin değeri elde edilememiş, yani kayıp değer ise bu işlem gözlem birimine NA değerinin atanması ile gerçekleştirilir. Yukarıdaki örnekte Tuncay isimli öğrenci karar dersi olmadığı için başarı notuna sahip olmasın. Dolayısıyla bu öğrenci karar notu kayıp değer olarak,

```
> karar<-c(NA,65,46,14,96,68,46,64)
> karar
[1] NA 65 46 14 96 68 46 64
```

atanır. R yazılımı fonksiyonlarının kayıp değerler için farklı yaklaşımları olabilmektedir. Bazı fonksiyonlar kayıp değerin yer aldığı değişkeni ya da gözlem birimini tamamen gözardı ederek işlem yaparken, kimi fonksiyonlarda işlemin yürütülmesi sırasında hata mesajı verebilirler. Kayıp değerlere sahip kullanıcıların, herhangi bir fonksiyonu kullanmadan önce, ilgili fonksiyonun kayıp değerler için yardım penceresi yardımıyla nasıl bir yol izlediğini öğrenmesi gerekmektedir.

KİŞİSEL FONKSİYON YAZMA

Bu aşamaya kadar R yazılımında işlemlerin fonksiyonlar ve bu fonksiyonların seçenekleri (ya da parametreleri) ile nasıl yürütüldükleri anlatıldı. Fakat hazır yazılmış fonksiyonlar bazen analizler için yeterli olmayabilir. Bu tür durumlar için kullanıcılar kendi fonksiyonlarını yazabilirler. R yazılımında bu işlem function (parametreler) komutu yardımıyla gerçekleştirilir. Örneğin; araştırmalarda sıkılıkla verinin tanımlayıcı istatistikleri ile his-

togram ile kutu grafiklerine ihtiyaç duyduğunuzu varsayıyalım. Bu amaçla her seferinde 3 komut kullanmak yerine bir tek fonksiyon oluşturularak bu işlem bir komut yardımıyla hazırlanabilir. Bu amaçla “ozetle” isimli bir fonksiyon oluşturalım.

```
> ozetle<-function(veri){
  # Bu işaret yardımıyla fonksiyon için yardımcı
  # bilgiler girilmesi mümkündür.
  # BU FONKSİYON VERİNİN ÖZETLEYİCİ İSTATİSTİKLERİNP
  # GÖRÜNTÜLER ve HİSTOGRAM - KUTU GRAFİĞİ ÇİZER
  +ozet<-summary(veri)
  +par(mfrow=c(1,2))  # Bir satırda 2 adet grafik
  +hist(veri)
  +boxplot(veri)
  +return(ozet)
}
```

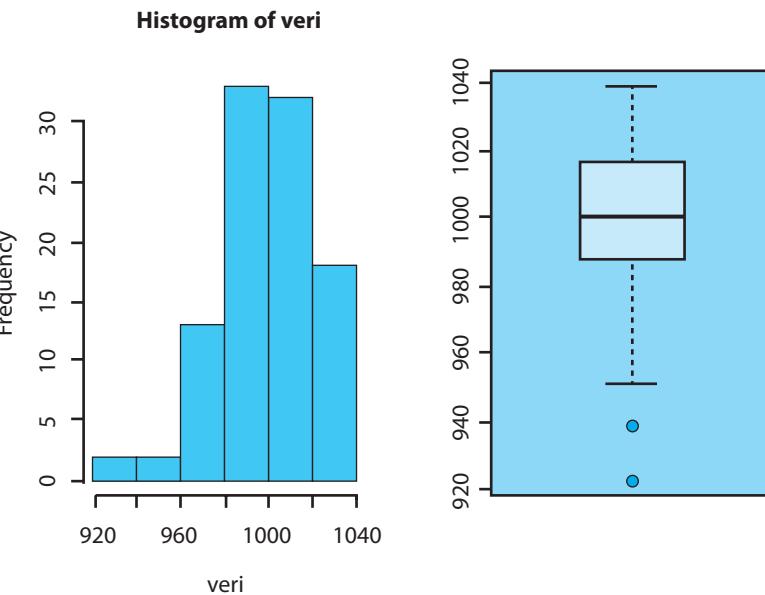
Daha önceden hafızaya alınmış bir x veri seti için bu fonksiyon çalıştırılsın.

```
> ozetle(x)
Min. 1st Qu. Median Mean 3rd Qu. Max.
922.7 988.5 1000.0 1000.0 1016.0 1039.0
```

Fonksiyon veri ile ilgili özetleyici bilgileri vermekle beraber Şekil 2.1'de verilen grafikleri de çizecektir.

Şekil 2.1

ozetle Fonksiyonu
Grafikleri



Kullanıcı daha sonra bu fonksiyon üzerinde ihtiyaçlarına göre düzenlemeler yapabilecektir. `function()` komutu, `return()` komutu ile sonlandırılır. Eğer bu fonksiyonun belli bir değeri isteniyor ise `return` içeresine bu değerin ait olduğu değişken yazılır. Yukarıdaki örnekte `return()` komutu içine özetleyici istatistiklerin atıldığı “`ozet`” değişken ismi verilmemizse, fonksiyon tanımlayıcı istatistikleri hesaplayarak bir değişkene atamasına rağmen, sonucu göstermeyecektir. Varsayıyalım ki kullanıcı tanımlayıcı istatistiklere ek olarak değişkenin varyansını da ayrıca elde etmek istesin. Fonksiyona varyans için var()

komutunun da eklenmesi gerekmektedir. Bu işlemde, fix(ozetle) komutu yardımcıyla gerçekleştirilebilir. Bu komut notepad'in açılarak içerisinde ilgili fonksiyon için düzeltmelerin yapılmasını sağlayacaktır. Varyansın da fonksiyona eklenmesi ile fonksiyon,

```
> ozetle<-function(veri){
# Bu işaret yardımcıla fonksiyon için yardımcı
# bilgiler girilmesi mümkündür.
# BU FONKSİYON VERİNİN ÖZETLEYİCİ İSTATİSTİKLERİНИ
# GÖRÜNTÜLER ve HİSTOGRAM - KUTU GRAFİĞİ ÇİZER
+ozet<-summary(veri)
+sapma<-var(veri)
+par(mfrow=c(1,2)) # Bir satırda 2 adet grafik
+hist(veri)
+boxplot(veri)
+return(ozet,sapma)
+}
```

hâlini alacaktır. Bu fonksiyon çalıştırıldığında hem özetleyici istatistikleri, hem de varyansı görüntüleyecektir.

```
> ozetle(x)
$ozet
Min. 1st Qu. Median Mean 3rd Qu. Max.
922.7 988.5 1000.0 1000.0 1016.0 1039.0
$sapma
[1] 486.8335
```

Dikkat edilirse fonksiyon için istenen 2 sonuç \$ işaretini ile alt değişkenlere atanmış olarak görüntülenmektedir. Eğer ozetle() fonksiyonundan yalnızca varyansın elde edilmesi gerekirse, ilgili fonksiyona sadece istenen sonucun \$ işaretini eklenmesi gereklidir.

```
> ozetle(x)$sapma
[1] 486.8335
```

Kullanıcıların kendi fonksiyonlarını yazmaları ve üzerinde işlemler yapabilmeleri R yazılımın en büyük üstünlüklerindendir. Bu sayede yeni önerilmiş bir teknik için program yazılabilmesi ve bu program yazılrken de hâli hazırda bulunan fonksiyonların kullanılabilmesi mümkün olmaktadır. Kullanıcılar birkaç fonksiyon yazımından sonra fonksiyon kullanımını konusunda birer uzman hâline gelebilmektedir.

Daha önce verilen ozetle fonksiyonuna medyan değeri eklenirse, yeni fonksiyon nasıl yazılmalıdır?



SIRA SİZDE

HAZIR VERİ AKTARIMI

Çoğunlukla veri setleri başka programlardan hazır olarak elde edilirler. Verinin R yazılımına okutulabilmesi için bir kaç farklı teknik bulunmaktadır. Bu işlem için kullanabilecek fonksiyonlar sırasıyla; scan() düşük seviyeli veri okutma işlemi, read.table() dosyalardan formatlanmış data frame elde edilmesi işlemi, read.fwf() belirgin bir genişlik tanımlanmış veri dosyalarından okuma işlemi, read.csv() değişkenlerin virgülle ayrıldığı dosyalardan okuma işlemi olur.

Özellikle Microsoft Excel dosyalarından okuma işlemleri gerçekleştirilirken, her bir çalışma sayfası “csv” dosyası olarak kaydedilerek daha sonra bunların her biri `read.csv()` fonksiyonu ile elde edilebilir. Örneğin, Microsoft Excel’de 3 değişken için 10’ar gözlem olduğunu varsayıyalım. Bu veri setinin veriseti.csv adı altında kaydedilsin. Şimdi veri seti R yazılımına okutulup istenilen bir değişkene atanabilir.

Resim 2.5



```
> read.csv("e:\\veriseti.csv",header=T,sep=";")
```

	A	B	C
1	5	8	7
2	3	9	4
3	2	5	5
4	6	4	9
5	9	3	8
6	6	6	1
7	5	5	2
8	4	5	3
9	8	5	2
10	7	3	3

```
> verisetim<-read.csv("e:\\veriseti.csv",header=T,sep=";")
```

Genellikle veri aktarımı `scan()` fonksiyonu ile yapılmaktadır. Bu fonksiyon özellikle vektörlerin tek tek okunmasında büyük kolaylık sağlamamaktadır.

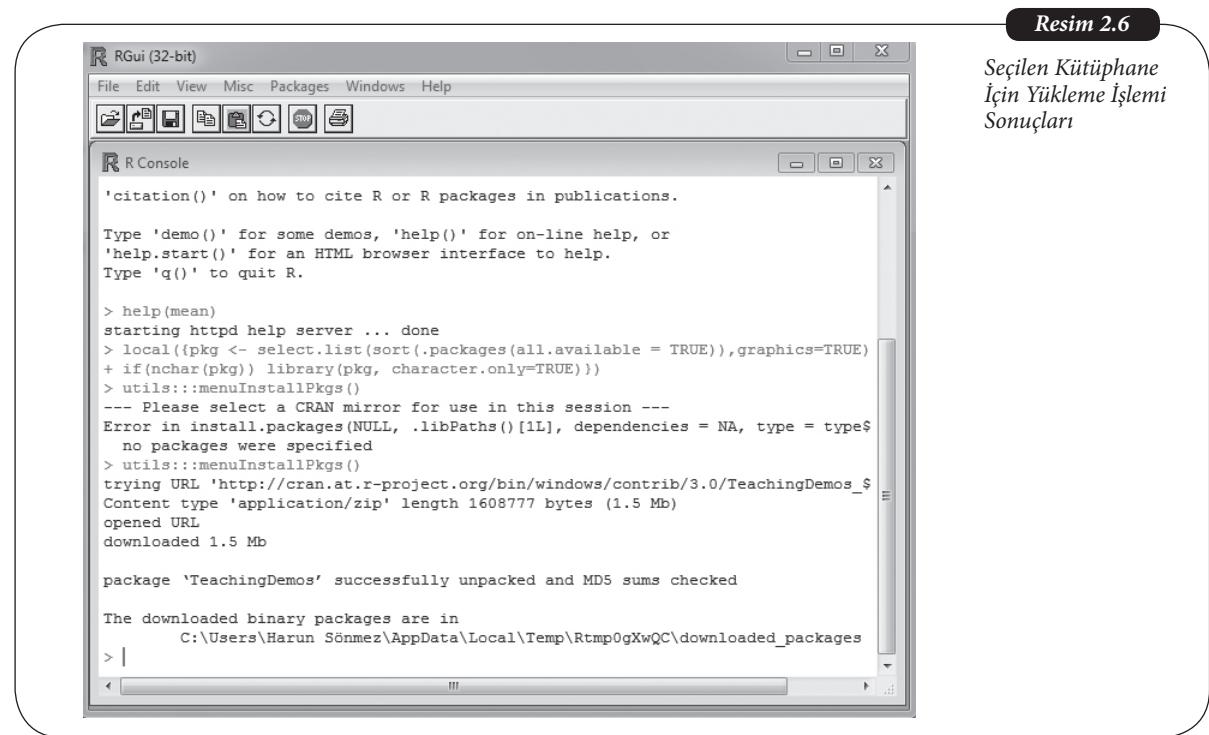
EK KÜTÜPHANE YÖNETİMİ

Başa kullanıcılar ya da R projesi ekibi tarafından oluşturulan ek kütüphaneler “Packages” menüsünden “Install Package(s)” seçeneği yardımıyla yürütülebilir.

Bu seçenek bir diyalog ekranı getirir ve bu ekranın o an için bulunan kütüphanelerin listesi sunulur (Resim 2.5).

Daha sonra listeden istenen kütüphane ya da kütüphaneler seçilerek OK’ye basılarak bu kütüphanelerin yüklenmesi işlemi gerçekleştirilir. Kütüphane yükleme işlemi sonrası R ekranı Resim 2.6’da sunulmuştur. Bu kütüphaneler daha sonra kullanılmak istenildiğinde `library(kütüphaneismi)` fonksiyonu yardımıyla erişilebilir hâle getirilirler. Kütüphaneler içinde yer alan fonksiyonlar için bilgiler “help” menüsü “Html Help” seçeneği yardımıyla elde edilebilir.

Resim 2.6



Kütüphaneler içinde yer alan fonksiyonlar için bilgiler hangi menü, hangi seçenek yardımıyla elde edilebilir?



SIRA SİZDE

Zaman zaman bu kütüphanelerde meydana gelen değişimlerin güncellenmesi faydalı olacaktır. Bu işlemde yine “Packages” menüsü “update packages” seçeneği yardımıyla gerçekleştirilebilir. Hâlen kullanıcıların elde edebileceği 600’ün üzerinde kütüphane bulunmaktadır.

Özet



R yazılımının temel komutlarını kullanmak

Herhangi bir atama yapılması ya da matematiksel bir ifadenin hesaplanması için en basit komutlar olarak meydana çıkan komutlar grubuna temel komutlar denir. R Console'da

> 27+54

komutu yazılarak Enter'a basıldığında

[1] 81

sonucu ekranada görüntülenecektir. Matematiksel işlemin hemen sonucunu elde etmek yerine sonuçlar herhangi bir değişkene de atanabilir. Bu atama işlemi için değişken <- işlem yapısı kurulmalıdır. Örneğin önceki toplam x gibi bir değişkene atanmak istenirse

> x <- 27+54

komutunun verilmesi yeterli olacaktır. Bir vektörü en basit şekilde yaratmak için c() fonksiyonu kullanılır. Önceden tanımlanmış bir vektörün birim sayısını öğrenmek için length() fonksiyonu kullanılır. Belirli bir düzene sahip olan vektörlerin yaratılmasında da seq() fonksiyonu kullanılır. Belirli bir düzene sahip vektorlerin oluşturulması için rep() fonksiyonu kullanılır. Bir değişkenin karekökü sqrt() fonksiyonu yardımıyla hesaplanabilir. R yazılımında matris oluşturmak için matrix() fonksiyonu kullanılır. R yazılımı ile mantık tipi değişkenlerin oluşturulması mümkündür. Doğru (T) ve Yanlış (F) olmak üzere iki mantıksal değer vardır. Mantık fonksiyonları yardımı ile ilgilenilen değişkenin bir karakter değişkeni mi yoksa sayısal bir değişken mi olduğu anlaşılabılır. Çeşitli istatistiksel analizler için oluşturulan farklı nesnelerin bir araya getirilmesinde List Nesnelerinden faydalанılır. R yazılımında veri seti içerisindeki faktör listeleri ve gözlem birimleri data frame olarak bir araya getirilirler.



R yazılımında kişisel fonksiyon oluşturmak

Hazır yazılmış fonksiyonlar bazen analizler için yeterli olmayabilir. Bu tür durumlar için kullanıcılar kendi fonksiyonlarını yazabilirler. R yazılımında bu işlem function(parametreler) komutu yardımıyla gerçekleştirilir. function() komutu, return() komutu ile sonlandırılır. Eğer bu fonksiyonun belli bir değeri vermesi isteniyor ise return içerisine bu değerin ait olduğu değişken yazılır. Kullanıcıların kendi fonksiyonlarını yazmaları ve üzerinde işlemler yapabilme-leri R yazılımın en büyük üstünlüklerindendir. Bu sayede yeni önerilmiş bir teknik için program yazılabilmesi ve bu program yazılarken de hâlihazırda bulunan fonksiyonların kullanılabilmesi mümkün olmaktadır.



R yazılımında ek kütüphane kullanmak

Başka kullanıcılar ya da R projesi ekibi tarafından oluşturulan ek kütüphaneler "Packages" menüsünden "Install Package(s)" seçeneği yardımıyla yürütülebilir. Listedeki istenen kütüphane ya da kütüphaneler seçilir. Bu kütüphaneler daha sonra kullanılmak istenildiğinde library(kütüphaneismi) fonksiyonu yardımıyla erişilebilir hâle getirilirler. Kütüphaneler içinde yer alan fonksiyonlar için bilgiler "help" menüsü "Html Help" seçeneği yardımıyla elde edilebilir.

Kendimizi Sınayalım

- 1.** R yazılımının elde edilmesiyle ilgili olarak aşağıdaki ifadelerden hangisi doğrudur?
 - a. R yazılımı, İnternet üzerinden ücretli elde edilebilir.
 - b. R yazılımı, İnternet üzerinden ücretsiz elde edilebilir.
 - c. R yazılımı, programın ana sitesini kullanarak ücretsiz CD üzerinden elde edilebilir.
 - d. R yazılımı, bilgisayar yazılım şirketinden ücretli elde edilebilir.
 - e. R yazılımı, bilgisayar yazılım şirketinden ücretsiz CD üzerinden elde edilebilir.
- 2.** R yazılımı ile ilgili aşağıdakilerden hangisi doğrudur?
 - a. Sadece veri işleme programıdır.
 - b. Sadece grafik çizme programıdır.
 - c. Veri işleme ve grafik çizme programıdır.
 - d. Resim düzenleme programıdır.
 - e. Sözcük işlemci programıdır.
- 3.** R yazılımında komutların girilmesi için kullanılan bölge ne ad verilir?
 - a. R Console
 - b. r-project
 - c. CRAN
 - d. help
 - e. length
- 4.** R yazılımında bir değişkenin karekökü aşağıdaki fonksiyonlardan hangisi ile hesaplanır?
 - a. length()
 - b. mean()
 - c. help()
 - d. sqrt()
 - e. rep()
- 5.** R yazılımında bir değişkenin aritmetik ortalaması aşağıdaki fonksiyonlardan hangisinin yardımıyla hesaplanabilir?
 - a. length()
 - b. mean()
 - c. help()
 - d. sqrt()
 - e. rep()
- 6.** R yazılımında bir vektör oluşturmak için aşağıdaki fonksiyonlardan hangisi kullanılır?
 - a. length()
 - b. mat()
 - c. help()
 - d. sqrt()
 - e. c()
- 7.** R yazılımında bir matris oluşturmak için aşağıdaki fonksiyonlardan hangisi kullanılır?
 - a. length()
 - b. matrix()
 - c. help()
 - d. sqrt()
 - e. c()
- 8.** R yazılımında function() komutu aşağıdaki komutlardan hangisi ile sonlandırılmalıdır?
 - a. return()
 - b. matrix()
 - c. help()
 - d. sqrt()
 - e. c()
- 9.** R yazılımında düşük seviyeli veri okutma işlemi aşağıdakilerden hangi ile yapılır?
 - a. read.fwf()
 - b. read.csv()
 - c. c()
 - d. sqrt()
 - e. scan()
- 10.** R yazılımında ek kütüphaneler hangi menü üzerinden yürütülebilir?
 - a. File
 - b. Edit
 - c. Packages
 - d. Install Package(s)
 - e. Help

Kendimizi Sınavalım Yanıt Anahtarı

1. b Yanınız yanlış ise “R Yazılımının Elde Edilmesi” konusunu yeniden gözden geçiriniz.
2. c Yanınız yanlış ise “R Yazılımının Temelleri” konusunu yeniden gözden geçiriniz.
3. a Yanınız yanlış ise “R Yazılımının Temelleri” konusunu yeniden gözden geçiriniz.
4. d Yanınız yanlış ise “R Yazılımının Temelleri” konusunu yeniden gözden geçiriniz.
5. b Yanınız yanlış ise “R Yazılımının Temelleri” konusunu yeniden gözden geçiriniz.
6. e Yanınız yanlış ise “R Yazılımının Temelleri” konusunu yeniden gözden geçiriniz.
7. b Yanınız yanlış ise “R Yazılımının Temelleri” konusunu yeniden gözden geçiriniz.
8. a Yanınız yanlış ise “Kişisel Fonksiyon Yazma” konusunu yeniden gözden geçiriniz.
9. e Yanınız yanlış ise “Hazır Veri Aktarımı” konusunu yeniden gözden geçiriniz.
10. c Yanınız yanlış ise “Ek Kütüphane Yönetimi” konusunu yeniden gözden geçiriniz.

Sıra Sizde Yanıt Anahtarı

Sıra Sizde 1

c() fonksiyonunu kullanarak döviz değişkeninde 3 farklı doviz ismi aşağıdaki oluşturulabilir.

> doviz <- c("Dolar", "Avro", "Lira")

> doviz

[1] "Dolar" "Avro" "Lira"

Sıra Sizde 2

ozetle fonksiyonuna medyan değeri eklenirse yeni fonksiyon aşağıdaki gibi yazılabilir.

> ozetle<-function(veri){

Bu işaret yardımıyla fonksiyon için yardımcı

bilgiler girilmesi mümkündür.

BU FONKSİYON VERİNİN ÖZETLEYİCİ İSTATİSTİKLERİNİ

GÖRÜNTÜLER ve HİSTOGRAM - KUTU GRAFİĞİ ÇİZER

+ozet<-summary(veri)

+sapma<-var(veri)

+medyan<-median(veri)

+par(mfrow=c(1,2)) # Bir satırda 2 adet grafik

+hist(veri)

+boxplot(veri)

+return(ozet,sapma,medyan)

}

Sıra Sizde 3

Kütüphaneler içinde yer alan fonksiyonlar için bilgiler “help” menüsü “Html Help” seçeneği yardımıyla elde edilebilir.

Yararlanılan ve Başvurulabilecek Kaynaklar

- Becker, R.A., Chambers, J.M. and Wilks, A.R. (1988). **The NEW S Language**. Chapman&Hall, London.
- Chambers, J.M. and Hastie, T.J. (1992). **Statistical Models in S**. Chapman&Hall, London.
- Dalgaard, P. (2002). **Introductory Statistics with R**. Springer, New York.
- Er, F. (2003). **Açıklayıcı Veri Analizi**. Kaan Kitabevi, Eskişehir.
- Everitt, B. (2004). **An R and S-Plus Companion to Multivariate Analysis**. Springer, New York.
- Greg Snow (2005). **Teaching Demos: Demonstrations for Teaching and Learning**. R package version 1.1.
- R Development Core Team (2006). **R: A language and environment for Statistical Computing**. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL <http://www.R-project.org>
- Verzani, J. (2004). **Simple R: Using R for Introductory Statistics**. Chapman&Hall, Florida.

3

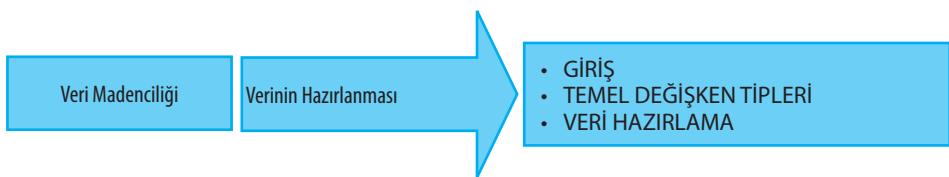
Amaçlarımız

- Bu üniteyi tamamladıktan sonra;
- Temel değişken tiplerini tanımlayabilecek,
 - Veri madenciliği için veriyi hazırlayabilecek,
 - Veride dönüşümler gerçekleştirebilecek bilgi ve becerilere sahip olabileceksiniz.

Anahtar Kavramlar

- İsimsel (Nominal) Değişken
- İkili (Binary) Değişken
- Sıra Gösteren (Ordinal) Değişken
- Tamsayılı (Integer) Değişken
- Aralıklı Ölçümlendirilmiş (Interval-Scaled) Değişken
- Oranlı Ölçümlendirilmiş (Ratio-Scaled) Değişken
- Veri Temizleme
- Veri Birleştirme
- Veri İndirgeme
- Veri Dönüştürme

İçindekiler



Verinin Hazırlanması

GİRİŞ

Veri madenciliği Radyo Frekansı ile Tanımlama (RFID), barkod, karekod, veri depolama araçları gibi teknolojilerle birlikte 1990'lı yillardan beri gelişmektedir. Sürekli bir gelişim içinde olan veri madenciliğinin o günün şartlarına göre yapılan tanımı da kullanım yerine ve zamanına göre farklılık göstermektedir. Bu tanımlardan biri de daha önceden bilinmeyen geçerli ve uygulanabilir bilgilerin geniş veritabanlarından elde edilmesi ve işletme kararları verilirken bu bilgilerin kullanılması olarak söylenebilir. Bu tanımda geçen “önceden bilinmeyen” söylemiyle anlatılmak istenen elde edilecek olan sonucun tahmin edilememesidir. Önceden bilinen veya tahmin edilebilen sonuçlar için veri madenciliği maliyeti nedeniyle tercih edilmemesi uygun olacaktır.

Veri madeninde bulunan **veri** insan tarafından oluşturulmuş bir bilgisayar dosyasından, verileri tasarlamak ve yönetmek için kullanılan bir işletme veri tabanı yönetim sisteminde, standart bir veri tabanı sisteminden, otomatik bilgi kaydı oluşturan bir araçtan, uydu üzerinden ve bunlara benzer şekilde kaynaklardan gelmiş olabilir. Farklı kaynaklardan gelen veri geliş kaynağının özelliğine göre çok çeşitli yapılarda, şekillerde ve tiplerde bulunabilir. Bu yapıdaki veri büyük olmasının yanı sıra çeşitli hatalar, kayıp değerler veya aykırı değerler içeriyor olabilir. Bir madenden çıkarılmayı bekleyen değerli taşlar gibi bu veri de çeşitli analizlerde kullanılmak üzere veritabanında bekler.

Toplanan ham veri diğer bir deyişle işlenmemiş verinin veri madenciliğinde analize hazır duruma getirilmesi amacıyla yapılan işlemler bütününe veri hazırlama adı verilir. Literatürde veri hazırlamaya ilgili izlenmesi gereken aşamalar araştırmacıdan araştırmaçıya göre farklı isimler ve farklı sayıda aşamalar olarak verilse de sonuçta amaç hepsinde aynıdır. Verinin hazırlanmasındaki amaç ham verinin yapısında bulunan ve onu degersizleştirilen hataları ve sorunları ortadan kaldırmaktır.

Verinin temizlemesi, birleştirilmesi, indirgenmesi, dönüştürülmesi ve anlaşılması gibi veri hazırlama işlemi veri analistinin zamanının %80'ini harcamasına sebep olur. Veri hazırlama aşamasında yapılan çalışmalar veri madenciliği çıktı kalitesini artıracı yönde olur.

Bu üniteye veri madenciliğinde verinin hazırlanmasıyla ilgili olarak temel değişken tipleriyle veri hazırlamada kullanılan verinin temizlenmesi, birleştirilmesi/bütünleştirilmesi, indirgenmesi, dönüştürülmesi (normalleştirme), kayıp veri ve aykırı değer üzerinde durulacaktır.

Veri, insan tarafından oluşturulmuş bir bilgisayar dosyasından, verileri tasarlamak ve yönetmek için kullanılan bir işletme veri tabanı yönetim sisteminde, standart bir veri tabanı sisteminden, otomatik bilgi kaydı oluşturan bir araçtan, uydu üzerinden ve bunlara benzer şekilde kaynaklardan gelmiş olabilir.

TEMEL DEĞİŞKEN TİPLERİ

Özellik, hakkında bilgi edinilmek istenen canlı, cansız varlıklar veya olayların sahip oldukları ve birbirinden ayırt edilmesine yardımcı olan değişkenler veri madenciliğinde bir veri setinin sunumunda kullanılan tablo gösteriminde sütunlarda yer alır.

Şekil 3.1

Veri Setinin Tablo Gösterimi

		ÖZELLİKLER					
		—	A	B	C	D	E
NESNELER	K1						
	K2						
	K3			C			
	K4						
	K5						

Verilen nesne için özelliğin aldığı değer

Veri madenciliğinde bir veri setinin sunumunda kullanılan tablo gösteriminde süturlarda **nesneler** yer alır.

Birimlerin sahip olduğu özelliklerin derecesinin belirlenerek sonuçların sayısal olarak ifade edilmesine **ölçme** adı verilir. Diğer bir deyişle gözlem ya da deney sonucunda elde edilen verilerin nicel olarak belirtilebilmesi amacıyla ölçmeye başvurulur. Sonuç olarak ölçümede bir tanımlama söz konusudur ve ölçmenin hangi ölçek ile yapılarak değerlendirildiği önemlidir. Örnek olarak bir markette satılan ürünlerin türlerine göre sınıflanması, market çalışanlarının yönetim katından en alt çalışanına kadar sıralanması, market alışverişinde satın alınacak bir ürünün ağırlığının ölçülmesi ve çalışanların aylık performanslarına göre değerlendirilerek ölçülmesi işlemlerinin tamamında bir ölçme işlemi vardır. Bu ölçme işlemleri arasındaki fark, her birinde kullanılan ölçeklerin farklı olmasıdır. Burada ölçek kavramı ölçmeye konu olan özelliklerin sınıflanması, sıralanması, derecelenmesi ya da miktar ve derecelerinin belirlenebilmesi için uyulması gereken kurallarla kısıtlamaları belirleyen ölçme aracı olarak tanımlanır.

Bir nesnenin özelliklerinin ölçme şekline göre bir çok değişken tipi tanımlanabilir. Değişken tiplerinin aralarındaki farkların tam olarak bilinmemesi veri analizinde çeşitli problemlere yol açabilir. Bu bölümde ilgili değişken tipleri açıklanacaktır.

İsimsel (Nominal) Değişkenler

Sınıflayıcı ölçek, gözlem değerlerinin tek tek nitel kategori ya da sınıflara atanması sonucu oluşan öklärktir. Daha önce verilen bir markette satılan ürünlerin türlerine göre sınıflanmasıörneğinde sınıflayıcı ölçek kullanılır. Cinsiyet sınıflaması veya hastaneye başvuran hastaların rahatsızlıklara göre sınıflandırılması sınıflayıcı ölçüye örnek olarak verilebilir.

İsimsel değişken sayısal bir formda olabilir. Ancak bu sayısal değer matematiksel bir hesaplama ya da işlem yapmak için uygun değildir. Örneğin; 5 kişi 1, 2, 3, 4, 5 olarak sayılarla ifade edilebilir. Buradaki sayılar üzerinde aritmetik bir işlem yapmak anlamlı olmayacağından, örnekteki sayılar sadece bir etiket görevi görecektir.

İkili (Binary) Değişkenler

İsimsel değişkenlerin özel bir şekli olan ikili değişkenler 0 ve 1, doğru ve yanlış, pozitif ve negatif, cinsiyet özelliğinde olduğu gibi erkek ve kadın gibi sonuçları sadece iki şekilde ortaya çıkan değişkenlerdir.

Sıra Gösteren (Ordinal) Değişkenler

Bu değişken tipi de isimsel değişken tipine benzerdir. Ancak değişkenin almış olduğu değer derecesi bakımından sıraya dizilmesinde önemlilik gösteriyorsa sıra gösteren değişken söz konusu olur. Market çalışanlarının yönetim katından en alt kademeye kadar sıralanması örneği sıra gösteren değişkene örnek olarak verilebilir. Çalışanların konumları arasında bir başka çalışmaya göre daha yüksek ünvan ya da kıdem yönünden derecelendirme söz konusudur. Bunun yanında aynı ünvana ya da kıdem sahip kişiler arasında ise eşitlikten söz edilebilir. Sınıflayıcı değişken yalnızca eşitlik ölçüsüne dayandırılıyordu. Sırayayı gösteren hem eşitlik hem de sıralama ölçüsünü kullandığından isimsel değişkeni de kapsar. Sıra gösteren değişkene başka bir örnek olarak öğrencilerin üniversite bitirme dereceleri yönünden sıralanması verilebilir.

Tam sayılı (Integer) Değişkenler

Alacağı değerler 0, 1, 2, ... gibi tam sayılar olarak belirtilebilen değişkenlerdir. Bu nedenle tam sayılı değişkenlerin ondalıklı değerler alması söz konusu değildir. Markette bir gün içinde satılan ekmek sayısı, belli bir depodaki koli sayısı ya da palet sayısı, bir ailedeki çocuk sayısı örnek olarak verilebilir. Tam sayılı değişkenlerle toplama, çıkarma ve çarpma işlemleri yapmak anlamlıdır.

Aralıklı Ölçümlendirilmiş (Interval-Scaled) Değişkenler

Sıra gösteren (ordinal) değişkenin tüm özelliklerini içermek ve ürettiği bilgileri üretmekle beraber birimler arasında özellik farkları matematiksel olarak belirlenebilir. Nicel değişkenlerin ölçümünde kullanılır. Belirli bir başlangıç noktası olmamakla birlikte ölçü birimi vardır. İfadeleri sayısal olarak sıralanabilmesine olanak vermektedir. Her ne kadar eşit aralıklı ölçekte ilgilenilen değişken matematiksel sonuçlar verse de kullanılan ölçüm için belirli bir yokluk anlamına gelmeyen sıfır ölçme düzeyi bulunabilir. Örneğin; hava sıcaklığı nicel ölçme düzeyine sahiptir ve yokluk anlamına gelmeyen sıfır değeri bulunabilir. Buradaki sıfır ölçme düzeyi havada sıcaklığın olmadığı anlamına gelmez. Bu değişken için matematiksel işlemler uygun olmakla beraber oran hesaplamaları için uygun değildir.

Oranlı Ölçümlendirilmiş (Ratio-Scaled) Değişkenler

Oranlı ölçümlendirilmiş (ratio-scaled) değişkenler aralıklı ölçümlendirilmiş (interval-scaled) değişkenlere benzer olmakla beraber bu değişkende sıfır başlangıç noktası tüm ölçüm araçlarında aynı anlamı taşır. Örneğin; bir varlığın ağırlığı için “sıfır” ifadesi kullanıldığından ölçüm metrik türne bakılmadan bu varlığın ağırlığının olmadığı anlamı çıkarılır. Diğer bir deyişle sıfır kilogram ve sıfır gram aynı anlamı taşır. Oranlı ölçümlendirilmiş (ratio-scaled) değişkenler daha önce ele alınan değişken tiplerinin tüm özelliklerini içerir. En büyük özelliği yokluk anlamına gelen belirli bir sıfır değerini barındırıyor olması bu nedenle ölçme düzeyleri arasında oransal analizler yapılabilmesine olanak tanıyor olmasıdır.

Yukarıda açıklamaları yapılan değişken tipleri özelliklerine göre kategorik ve sürekli değişkenler olarak iki grupta toplanabilir. Kategorik değişkenler grubunda isimsel (nominal), ikili (binary) ve sıra gösteren (ordinal) değişkenler girerken sürekli değişkenler grubuna tam sayılı (integer), aralıklı ölçümlendirilmiş (interval-scaled) ve oranlı ölçümlendirilmiş (ratio-scaled) değişkenler girer.

Veri madenciliğinde tenel değişken tipleri isimsel (Nominal), ikili (Binary), Sıra Gösteren (Ordinal), Tamsayılı (Integer), Aralıklı Ölçümlendirilmiş (Interval-Scaled), Oranlı Ölçümlendirilmiş (Ratio-Scaled) değişkenler olmak üzere gruplandırılabilir.

SIRA SİZDE

1

İsimsel (nominal) değişken ile sıra gösteren (ordinal) değişken arasındaki fark nedir?

VERİ HAZIRLAMA

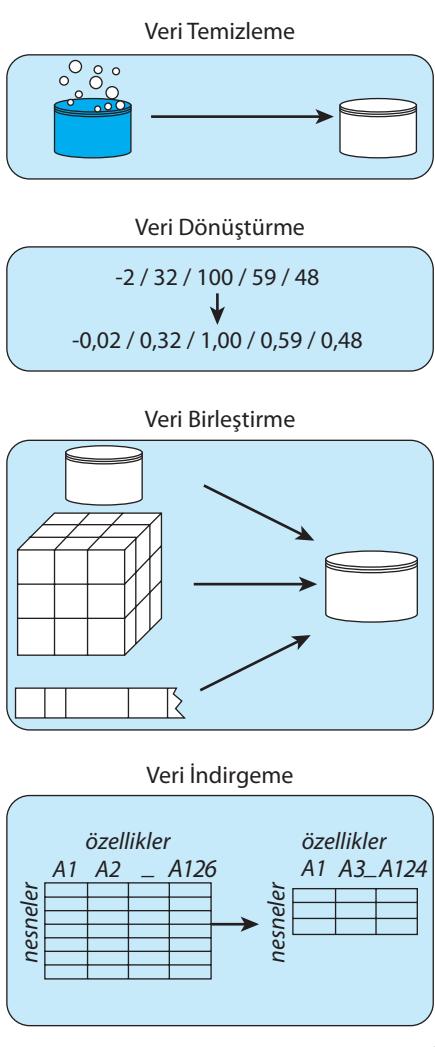
Veri madenciliğinde veri hazırlama aşaması veri kalitesini anlamak ve iyileştirmek konusuna odaklanmak veri madenciliği **çıktı kalitesini** arttırmıştır.

Veritabanlarında yer alan verilerin mükemmel olması çoğu zaman mümkün değildir. Veri madenciliği tekniklerinin çoğu verilerdeki kusurları göz ardı edebilmesine rağmen veri kalitesini anlamak ve iyileştirmek konusuna odaklanmak veri madenciliği **çıktı kalitesini** arttırır. Veri kalitesi kavramı verideki gürültü ve aykırı değerler, eksik, tutarsız veya tekrarlı verilerin varlığı ile ölçülebilir. Veri kalitesinin düşük olması verinin analiz yapan kişiye yanıtmasına yani hedeflenen sonuca ulaşamamasına neden olur. Verilerin veri madenciliğine uygun hale getirilebilmesi kusurlarının araştırılarak giderilmesi gerekmektedir. Verilerdeki kusurların giderilmesi için birtakım ön hazırlık süreçleri uygulanır. Veri hazırlamaya ilgili olan veri temizleme, veri birleştirme, veri indirgeme ve veri dönüşüm süreçleri Şekil 3.2'de sembolik olarak verilmiştir.

Şekil 3.2

Veri Hazırlama Süreci

Kaynak: Jiawei Han ve M. Kamber, Data Mining, USA: Academic Press, 2001



Birçok veri madenciliği uygulaması veri hazırlama süreçlerinden sadece biri değil birden fazlasının uygulanmasını gerektirebilir.

Veri hazırlama süreçlerinden biri olan veri temizleme verideki tutarsızlıkların giderilmesi ve verideki gürültünün giderilmesi için uygulanır. Veri dönüştürme olarak normalleştirme kullanılabilir. Veri birleştirme farklı kaynaktan gelen veriyi uygun bir veri tabanında birleştirir. Veri indirgeme ise fazla olan bazı değişkenlerin çıkarılması, birleştirilmesi veya kümeleme yaparak veri büyüğünün azaltılması amaçlanır. Veri yapısına uygun olacak şekilde bu süreçlerden biri veya birkaç veri madenciliğinden önce uygulanarak elde edilen sonuçların kalitesi, güvenilirliği ve veri madenciliği aşamasında harcanacak zaman artırılabilir.

Veri Temizleme

Veri madenciliğinde veri kalite problemlerini engellemek için önce veri kalitesi problemlerinin farkına varılarak doğrulanması ve zayıf veri kalitesini göz ardı edebilen algoritmaların kullanılması üzerinde odaklanılır. Veri kalitesi problemlerinin farkına varılması ve doğrulanması veri temizleme olarak adlandırılır.

Ölçülen bir değerdeki hata ve yanlış özellik değerleri ki bunlar; hatalı veri toplama gereçlerinden, veri girişi problemlerinden,

veri iletimi problemlerinden, teknolojik kısıtlardan ve özellik isimlerindeki tutarsızlıktan gürültülü veri olarak tanımlanan veri oluşmasına neden olur.

Eksik verilerin tamamlanması, aykırı değerlerin teşhis edilmesi amacıyla gürültünün düzeltilmesi ve verilerdeki tutarsızlıkların giderilmesi gibi işlemler veri temizlemeyle mümkün olur.

Veri temizleme için temel yöntemler eksik veri, gürültülü veri ve tutarsızlık olmak üzere üç temel başlıkta gruplanabilir.

Eksik Veri

Madenciliği yapılacak verinin bazı özellik değerleri boş, diğer bir deyişle eksik olabilir. Özellikle değerlerinde eksik veya boş değer olmasının birçok nedeni vardır. Veritabanında yer alan verilerin anket verisi olması ve bilgisi toplanan bireyin bilgi vermek istemesi, yanlış anlamaya veya veri giren personelin hatası, diğer veri özellikleriyle tutarsızlığı yüzünden silinmesi gibi nedenler eksik veri oluşmasına neden olabilir. Bazı durumlarda değerin boş olması eksik veri değil her nesne için uygulanabilir bir özellik olmamasından kaynaklanabilir. Bir kimlik tablosunda bayanlara ait kayıt alanlarında askerlik bilgisinin yer almaması bu duruma örnek verilebilir. Bu durumda benzer verilerin eksik değer olarak algılanması ve giderilmesi hataya neden olabilecektir. Eksik değer ile ilgili stratejiler üç ana grupta toplanabilir. Bu stratejiler aşağıdaki bölümlerde incelenmiştir.

Veri nesne veya özelliklerini elemek: Eksik veriyle ilgili nesneleri diğer bir deyişle eksik değer olan kayıtları çıkartmak basit ve etkili bir stratejidir. Ancak eksik verilere sahip olan nesnelerin çıkartılması nesnelerin diğer özelliklerinde yer alan enformasyonun kaybına neden olacağından analizin güvenirligini azaltır. Bununla birlikte bir veri kümesi sadece birkaç eksik veriye sahip nesne içeriyorsa bu nesneleri çıkarmak uygun olabilir. Diğer bir strateji de eksik verilere sahip özelliklerin analizden çıkartılmasıdır. Çıkarılacak özellikler analiz için önemli olabileceğiinden bu stratejinin uygulanmasına dikkat edilmelidir.

Eksik verinin tahmin edilmesi: Bazı durumlarda eksik veri güvenilir bir şekilde tahmin edilebilir. Örneğin birkaç tane geniş alana yayılmış eksik veriye sahip zaman serisini düşürelim. Bu durumda eksik veri diğer veriler kullanılarak tahmin edilebilir.

Eksik verinin tahmin edilmesi için kullanılan başlıca stratejiler aşağıda verilmiştir.

- Eksik verinin el ile doldurulması; bu strateji zaman alıcıdır ve eksik verinin fazla olduğu büyük veri kümelerinde kullanılması uygun değildir.
- Eksik verinin tamamlanmasında genel bir sabitin kullanılması; tüm eksik verinin belirlenecek bir sabit değer ile değiştirilmesidir. Bu değişiklik uygulandığında veri madenciliği algoritmalarını olumsuz etkileyebilir. Bu nedenle basit bir strateji olmasına rağmen tercih edilmez.
- Eksik verinin verinin özelliğin diğer veriler dikkate alınarak tamamlanması; bu stratejide eksik veri, aynı özelliğin eksik olmayan kayıtları göz önüne alınarak ortalamaya, medyan, mod gibi verinin tamamını temsil eden tek bir değer ile değiştirilir.
- Eksik verinin kendi sınıfında yer alan değerlerin ortalaması ile tamamlanması; eksik verinin tamamlanması öncesinde veri üzerinde bir sınıflama çalışması yapılarak eksik verinin ait olduğu sınıflar belirlenir. Her eksik verinin bulunduğu sınıf eksik olmayan özellik verilerinin ortalaması ile tamamlanır.
- Eksik verinin tamamlanmasında en uygun değerin kullanılması; eksik verinin bulunduğu özelliğin en uygun değeri regresyon yönteminin kullanıldığı sonuç çıkarmaya dayalı araçlar veya karar ağacıları kullanılarak belirlenebilir. Diğer stratejilere kıyasla bu strateji eksik veriyi tahmin etmede mevcut enformasyondan en fazla faydalanan yöntemdir. Bu nedenle en sık kullanılan stratejidir.

Eksik verinin gözardı edilmesi: Birçok veri madenciliği yaklaşımı eksik veriyi göz ardi edecek şekilde düzenlenebilir. Örneğin, kümleme analizinde nesne çiftleri arasındaki benzerlik hesaplamalarını düşünelim. Eğer bir çift nesnenin biri veya her ikisi bazı özellikler için eksik veriye sahipse o zaman benzerlik sadece eksik veri içermeyen özellikler kullanılarak düzenlenebilir.

nilarak hesaplanabilir. Özelliklerin toplam sayısı az veya eksik verinin sayısı çok olmadığı sürece benzerlik hesaplaması hemen hemen doğru olacaktır. Bundan başka pek çok sınıflama yaklaşımlarında eksik veri göz ardı edilerek düzenlemeler yapılabilir.

Gürültülü Veri

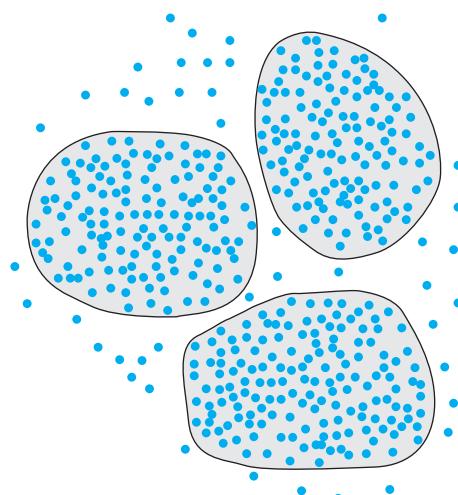
Gürültü, veri madenciliği teknigi ile analiz edilmek istenilen verilerdeki beklenen değerlerden sapan aykırı değerler veya hatalardır. Gürültülü veri büyük veritabanları ve veri ambarlarında karşılaşılan yaygın problemlerdendir. Ölçülen bir değerdeki hata veya hatalı veri toplama, veri girişi problemleri, teknolojik kısıtlar gibi yanlış nitelik değerleri gürültülü verinin olası nedenleridir. Veri madenciliği uygulanmadan önce bu değerlerin neden olduğu gürültü düzeltilmelidir. Verideki gürültünün belirlenip giderilmesi için bölmemeleme, kümeleme, bilgisayar ve insan denetiminin birleştirilmesi ve regresyon yöntemleri kullanılabilir.

Bölmeye yöntemlerinde öncelikle veriler artan sıradan sıralanır. Bölme sayısı belirlenerek veriler eşit sayıda bölmelere ayrılır. Farklı düzeltme seçenekleri kullanılarak her bölmeye veriler düzleştirilir. Örneğin, her bölmeye sayılar ilgili bölmenin ortalaması ile değiştirilir. Bölmeye mede verilerin sıralanmasıyla yapılan düzeltme, komşu değerlere yakınlık sağlayacağından yerel bir düzleştirme sağlar.

Aykırı değerler kümeleme analizi ile ortaya çıkarılabilir. Kümeleme analizinde benzer değerler gruplar veya kümeler hâlinde bir araya getirildiğinden aykırı değerler Şekil 3.3'teki gibi belirlenir.

Şekil 3.3

Kümeleme Analiziyle
Aykırı Değerlerin
Belirlenmesi



Aykırı değerler bilgisayar ve insan denetiminin birleşimi ile belirlenebilir. Örneğin, bir el yazısı karakter tanıma uygulamasında aykırı değer örüntüleri karakterin tahmin edilmesinde yardımcı olabilir. Bu örnekte aykırı değer örüntülerinin faydalı ya da işe yaramaz olup olmadığı insan tarafından daha kolay ayırt edilebilir.

Tutarsız Veri

Bazı veritabanı kayıt işlemlerinde verilerde tutarsızlıklar oluşabilir. Bazı tutarsızlıklar dış veri kaynakları kullanılarak elle düzeltilebilir. Örneğin, bir veri girişinde yapılan hata verinin girildiği kaynak belgelerden kontrol edilerek düzeltilebilir. Bilgi mühendisliği araçları bilinen veri sınırlamalarını bozan verileri ortaya çıkaran araçlara sahiptir. Örneğin, özellikler arasındaki işlevsel bağımlılıkların bu hataları bulabildiği bilinmektedir.

Veri Birleştirme

Veri birleştirme çoklu kaynaklardan gelen verinin uygun bir veri ambarına birleştirilmesidir. Çoklu veri kaynakları veritabanları, veri küpleri veya dış dosyalardan oluşabilir. Veri birleştirmede şema birleştirmesi, fazla veri sorunları ve veri değer karmaşalarının belirlenmesi ve çözümlenmesi olmak üzere üç temel konu ön plana çıkar.

Şema birleştirme iki farklı kaynaktan gelen verilerin eşleştirilmesi için aynı varlıklar belirlenerek veriler şemalar yardımıyla birleştirilir. Şema birleştirme işleminde hataları engellemek için meta veri kullanılabilir. Veritabanları ve veri ambarlarında yer alan meta veri kavramı veri hakkında depolanan veri olarak tanımlanır.

Veri birleştirmede ikinci önemli konu olan veri fazlalığı, bir varlığın özelliklerinin birden fazla kaynaktan toplanması durumunda ortaya çıkar. Bazı veri fazlalığı korelasyon analizi ile ortaya çıkarılabilir. Korelasyon analizi iki değişken arasındaki ilişkinin yönünün, büyülüğünün ve önemini gösteren istatistiksel bir yöntemdir.

Veri birleştirmede üçüncü önemli konu veri değer karmaşıklığının belirlenmesi ve çözümlenmesidir. Farklı veri kaynaklarından gelen özellik değerleri ölçekte, birim sistemi veya gösterimdeki farklılıklar yüzünden birbirlerinden farklı olabilirler. Örneğin ağırlık özelliği farklı kaynaklarda farklı birim sistemiyle depolamış olabilir. Veri bütünlendirme işlemlerinde verinin bu tür heterojenliği dikkate alınmalıdır.

Veri İndirgeme

Oldukça karmaşık olan ve çok büyük veri kümelerinin madenciliğinin yapılması çok uzun zaman alındıktan bu tür verilerin olduğu gibi alınarak analiz edilmesi uygulanabilir ve pratik olmamaktadır. Bu nedenle veri indirgeme yöntemleri çok daha küçük hacimde indirgenmiş veri kümelerinin oluşturulması için kullanılır. Veri indirgeme işlemi sonrasında elde edilen veri seti üzerinde uygulanan madencilik sonucu verinin tamamından elde edilen sonuçoan çok farklı olmamalıdır. Veri indirgeme yöntemleri aşağıdaki bölümlerde açıklanmıştır.

Veri indirgeme yöntemleri olarak veri küpü birleştirme, boyut indirgeme, veri sıkıştırma ve büyük sayıların indirgenmesi yöntemleri ortaya çıkar.

Veri Küpü Birleştirme

Veri madenciliğinin veri kaynağının bir Online Analitik Süreç (OLAP:On Line Analytical Processing) sistemi olması durumunda ihtiyaç duyulan verilerin ön hesaplama ve özetlenmesi daha hızlı gerçekleştirilebilir. Veri küpleri çok boyutlu birleştirilmiş verileri saklar. Bazı durumlarda tüm verinin veri madenciliği algoritmalarında işlenmesi yerine özet bilgilerin kullanılması gerekebilir. Bu durumda OLAP küplerinin sağladığı özetleme fonksiyonlarından faydalanylabilir. Aylık satış fiyatlarının yıllık temelde daha küçük veri seti haline dönüştürülmesi örnek olarak verilebilir.

Boyut İndirgeme

Veri kümeleri analizle ilgisi olmayan veya gereksiz yüzlerce özellik içerebilir. Gereksiz olan özelliklerin indirgenmesi bir başka deyişle boyut indirgeme pek çok veri madenciliği algoritmasının daha verimli çalışmasını, daha anlaşılabılır bir modelin oluşturulmasını, verilerin daha kolay görselleştirilmesini ve veri madenciliği algoritmaları için gerekli olan işlemci süresi ve hafızasını azaltır. İyi bir özellik alt kümesi asıl özelliklerden seçilir. Asıl özelliklerin sayısı “d” ise olası alt küme sayısı “ 2^d ” olmaktadır. En iyi (veya en kötü) özellikler istatistiksel anlamlılık testleri kullanılarak belirlenir. Bu testler özelliklerin birbirinden bağımsız olduklarını kabul eder. Asıl özelliklerin sayısı fazla olduğunda en iyi alt kümeyi belirlemenin için yapılacak araştırma maliyetli olabilecektir. Bu nedenle indirgenmiş özellik uzayını araştıran sezgisel yöntemler yaygın olarak kullanılır. Özellik alt küme seçiminde “ileriye doğru seçme”, “geriye doğru eleme” ve “ileriye doğru seçme ve geriye doğru eleme birleşimi” gibi tekniklerin uygulandığı sezgisel yöntemler kullanılır.

Diğer bir yöntem de sınıflama için karar ağacılarının oluşturulmasında kullanılan enformasyon kazanma (information gain) gibi ölçümlerin kullanılmasıdır. Eğer madencilik görevi sınıflamaya ve madencilik algoritması özellik alt kümesini belirlemede kullanılıyorsa bu özellik seçme yöntemine “sarmalama” (wrapper) yaklaşımı denilir. Aksi takdirde bu bir süzme yaklaşımı olarak ifade edilir. Sarmalama yaklaşımı ile özellikler çıkartılırken algoritmanın değerlendirme ölçümünü en iyilediği için daha geçerli sonuçlara ulaşılır. Ancak süzme yaklaşımından çok daha fazla hesaplama gerektirir.

Veri Sıkıştırma

Veri sıkıştırımda veri kodlama veya dönüşümleri asıl verinin indirgenmiş veya sıkıştırılmış gösterimini elde etmek için uygulanır. Asıl veri herhangi bir enformasyon kaybı olmaksızın sıkıştırılmış veriden tekrar elde edilebiliyorsa o zaman veri sıkıştırma işlemi “ka-yıpsız” (lossless) olarak nitelendirilir. Bundan başka asıl verinin gerçeğe yakın bir değeri oluşturulabilirse o zaman veri sıkıştırma kayıplı (lossy) olarak nitelendirilir. Metin verilerin sıkıştırılmasında kullanılan algoritmalar kayıpsız sıkıştırma yöntemleri olmalarına rağmen verinin sınırlı olarak işlenmesine neden olurlar. Bu nedenle daha yaygın ve etkili olan kayıplı yöntemler tercih edilir.

Büyük Sayıların İndirgenmesi

Verilerde yer alan büyük sayıların daha küçük şekilleri seçilerek veri hacminin indirgenmesi için uygulanan yöntemlerdir. Veri hacmi parametrik veya parametrik olmayan yöntemler kullanılarak indirgenir. Parametrik yöntemlerde gerçek veri yerine sadece veri parametreleri saklanır ve sıkıştırılan veriyi tahmin etmek için bir model kullanılır. Parametrik olmayan veri indirgeme yöntemlerine histogramlar, kümemeleme ve örneklemeye gösterilebilir.

Parametrik olan regresyon ve logaritmik doğrusal regresyon modelleri verinin parametrelere dayalı gösterimini oluşturarak verinin indirgenmesinde kullanılabilir. Doğrusal regresyon veriyi bir düz doğruya uydurarak modellerken çoklu regresyon birden fazla özellik vektörü kullanılarak veriye modeller. Logaritmik doğrusal regresyon kesikli çok boyutlu olasılık dağılımları yaklaşımını uygular ve kesikli özellikler kümesi için çok boyutlu veri kümelerinin her hücresinin tahmin edilmesinde kullanılır.

Parametrik olmayan veri indirgeme yöntemlerinin en yaygını histogram yöntemidir. Histogram yöntemi verileri farklı yöntemlerle aralıklara bölgerek veriye ilişkin dağılımı elde eder. Diğer bir yöntem ise kümememdir. Kümemeleme veri azaltmada kullanılan verilerin kümelenerek daha küçük bir aralığa indirgenmesiyle gerçekleştirilir. Örneklemeye de veri indirgeme için kullanılan parametrik olmayan yöntemlerden biridir. Örneklemeye geniş bir veri kümесinin çok daha küçük bir alt kümese ile gösterilmesini sağlayabilir. Veri indirgemede örneklemeye yöntemi bir gruplama sorgusunun cevabını tahmin etmek için yaygın olarak kullanılır.

SIRA SİZDE



Veri indirme yöntemlerinin isimlerini sayınız.

Veri Dönüşürme

Bazı durumlarda orijinal veri kümelerindeki özellikler gerekli enformasyonu içerdığı halde veri madenciliği algoritmaları için uygun yapıda olmayabilirler. Bu durumda orijinal özelliklerinden oluşturulan bir veya daha fazla yeni özellik orijinal özelliklerden daha faydalı olabilir. Veri dönüşümünde verilerin veri madenciliği için uygun formlara dönüştürülmesi düzeltme, bir araya getirme, genelleme, normalleştirme ve özellik oluşturma işlemleriyle gerçekleştirilir.

- Düzeltme; bölmeleme, kümeleme ve regresyon gibi teknikler kullanılarak verilerdeki gürültünün temizlenmesidir.
- Bir araya getirme; veriler bir araya getiren graplama fonksiyonları kullanılarak gerçekleştirilir. Günlük temelde bulunan bir veri özelliğinin aylık temele dönüştürülmesi örnek verilebilir.
- Genelleme; düşük düzeydeki verinin kavram hiyerarşisi kullanılarak daha yüksek seviyeye dönüştürülmesidir. Örneğin; yaş gibi sayısal verilerin kategorik olan genç, orta yaşlı veya yaşlı gibi değerlere dönüştürülmesi ya da cadde isimlerinden oluşan kategorik verilerin şehir veya ülke şeklinde daha yüksek kavramlara dönüştürülmesidir.
- Normalleştirme veya standartlaştırma; bir değişkenin standartlaştırılması veya normalleştirilmesi yaygın olarak kullanılan veri dönüşüm tekniğidir. Veri madenciliği terminolojisinde her iki terim birbiri yerine kullanılmaktadır. Ancak buradaki normalleştirme terimi, istatistikte kullanılan bir değişkenin normal dağılmış bir değişkene dönüştürülmesi ile karıştırılmamalıdır. Standartlaştırma veya normalleştirmenin amacı sayısal veri değerlerinin küçük bir bölgede yer alması için ölçeklenmesidir. Normalleştirilmiş veriler sınıflama için kullanılan yapay sinir ağları algoritmalarının öğrenme aşamasının hızlanmasına yardım edecektir. Kümeleme gibi mesafe ölçümülerine dayalı algoritmalarla normalleştirilmiş verilerin kullanılması faydalı olacaktır.
- Özellik oluşturma; yeni özellikler madencilik sürecine yardımcı olmak için verilen özellikler kümesinden oluşturulur ve düzenlenir. Özellik oluşturma karar aacı algoritmaları sınıflama için kullanıldığında bölümleme problemini azaltmaya yardımcı olabilir. Yükseklik ve genişlik özelliklerinden alan özelliğinin oluşturulması bu duruma bir örnek olarak verilebilir.

Normalleştirme veya standartlaştırma en çok kullanılan veri dönüştürme işlemidir. Normalleşirmede enk-enb normalleştirme, z skor normalleştirme ve ondalık ölçekteme yöntemleri kullanılır.

Enk-Enb Normalleştirme

Orijinal veri üzerinde doğrusal bir dönüşüm yapan bu yöntem veri içindeki en büyük ve en küçük sayısal değerin belirlenerek diğer değerleri buna uygun bir şekilde dönüştürülmesiyle yapılır. Enk-Enb normalleştirme sonucunda veri sıfır (en küçük değer) ile bir (en büyük değer) arasında sayısal bir değere dönüşür. Dönüşürme için aşağıdaki eşitlikten yararlanılır.

$$X^* = \frac{X - X_{enk}}{X_{enb} - X_{enk}}$$

Bu eşitlikte;

X^* : Dönüştürülmüş değeri

X : Gözlem değerini

X_{enk} : Verideki en küçük gözlem değeri

X_{enb} : Verideki en büyük değeri

ifade eder.

Veri dönüşümünde verilerin veri madenciliği için uygun formlara dönüştürülmesi düzeltme, bir araya getirme, genelleme, normalleştirme ve özellik oluşturma işlemleriyle gerçekleştirilir.

ÖRNEK 1

Aşağıdaki tabloda verilen X değişkenine ilişkin gözlem değerlerini Enk-Enb normalleştirme yöntemini kullanarak veri dönüşümünü sağlayınız.

X
251
148
166
244
472
356
379

Tabloda verilen X değişkenine ilişkin gözlem değerlerinden en küçük ve en büyük gözlem değerleri sırasıyla $X_{enk}=148$ ve $X_{enb}=472$ olarak belirlenir. Verilen eşitlik yardımıyla hesaplamalar aşağıdaki tablodaki gibi yapılır.

X	$X^* = \frac{X - X_{enk}}{X_{enb} - X_{enk}}$	X*
251	$\frac{251 - 148}{472 - 148}$	0,318
148	$\frac{148 - 148}{472 - 148}$	0
166	$\frac{166 - 148}{472 - 148}$	0,056
244	$\frac{244 - 148}{472 - 148}$	0,296
472	$\frac{472 - 148}{472 - 148}$	1
356	$\frac{356 - 148}{472 - 148}$	0,642
379	$\frac{379 - 148}{472 - 148}$	0,713

Bu örneği R yazılımı ile yapmak için öncelikle R paketlerinin içinden "cluster.Sim" paketini yüklemek gerekir. R yazılımının "cluster.Sim" paketinde 16 tane veri normalleştirme yöntemi bulunmaktadır. Normalleştirme işlemini gerçekleştirmek için "data.Normalization(x, type="n0", normalization="column")" komutu kullanılır. Bu komutun işlevleri aşağıdaki gibi açıklanır.

x: Değişken değerlerini içeren vektör

type: Normalleştirme için seçilen yöntem

normalization: Normalleştirme işleminin sütuna göre yapılmış yapılmayacağı bilgisi

Paket yüklenikten sonra X değişkenine ilişkin değerleri girmek için aşağıdaki komut yazılmalıdır.

```
> x <- c(251,148,166,244,472,356,379)
```

Bu komut ile X değişkeninin değerleri bir vektör olarak programa girilmiş olur. Daha sonra aşağıdaki komut yardımıyla normalleştirme işlemi gerçekleşir.

```
> data.Normalization(x,type="n4")
```

Bu komutta “n4” gözlem değerlerinin en küçük değerden çıkarıp, en büyük değer ile en küçük değer arasındaki farka bölen işlevi tanımlamaktadır. Komut çalıştırıldığında aşağıdaki sonuca ulaşılır.

```
[1] 0.31790123 0.00000000 0.05555556 0.29629630
[5] 1.00000000 0.64197531 0.71296296
```

Sonuçların elle yapılan hesaplamalar ile aynı olduğu görülebilir.

z-Skor Normalleştirme

z-skor normalleştirme diğer dönüştürme yöntemleri içinde uygulamada en çok kullanılan dönüştürme yöntemidir. Bir değişkene (özellik) ilişkin aritmetik ortalama ve standart sapma hesaplamasından sonra elde edilir. z-skor normalleştirme sonucunda veri sıfır ile bir arasında sayısal bir degere dönüşür. Dönüşürme için aşağıdaki eşitlikten yararlanılır.

$$X^* = \frac{X - \bar{X}}{s}$$

Bu eşitlikte;

X^* : Dönüştürülmüş değeri

\bar{X} : Gözlem değerini

X : Değişkenin (özellik) aritmetik ortalamasını

s : Değişkenin (özellik) standart sapmasını

ifade eder.

Örnek 1'de verilen X değişkenine ilişkin gözlem değerlerini z-skor normalleştirme yöntemini kullanarak veri dönüşümünü sağlayınız.

ÖRNEK 2

X
251
148
166
244
472
356
379

Tabloda verilen X değişkenine ilişkin aritmetik ortalama ve standart sapma aşağıdaki gibi hesaplanır.

$$\bar{X} = \sum_{i=1}^n \frac{X_i}{n} = \frac{251 + 148 + \dots + 379}{7} = \frac{2016}{7} = 288$$

$$s = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}} = \sqrt{\frac{(251-288)^2 + (148-288)^2 + \dots + (379-288)^2}{6}} = 118,71$$

Verilen eşitlik yardımıyla hesaplamalar aşağıdaki tablodaki gibi yapılır.

X	$X^* = \frac{X - \bar{X}}{S}$	X^*
251	$\frac{251 - 288}{118,71}$	-0,312
148	$\frac{148 - 288}{118,71}$	-1,179
166	$\frac{166 - 288}{118,71}$	-1,028
244	$\frac{244 - 288}{118,71}$	-0,371
472	$\frac{472 - 288}{118,71}$	1,550
356	$\frac{356 - 288}{118,71}$	0,573
379	$\frac{379 - 288}{118,71}$	0,770

Bu örneği R yazılımı ile yapmak için öncelikle R paketlerinin içinden "cluster.Sim" paketini yüklemek gerekir. R yazılımının "cluster.Sim" paketinde 16 tane veri normalleştirme yöntemi bulunmaktadır. Normalleştirme işlemini gerçekleştirmek için "data.Normalization(x,type="n0",normalization="column")" komutu kullanılır. Bu komutun işlevleri aşağıdaki gibi açıklanır.

x: Değişken değerlerini içeren vektör

type: Normalleştirme için seçilen yöntem

normalization: Normalleştirme işleminin sütuna göre yapılip yapılmayacağı bilgisi

Paket yüklenikten sonra X değişkenine ilişkin değerleri girmek için aşağıdaki komut yazılmalıdır.

```
> x <- c(251,148,166,244,472,356,379)
```

Bu komut ile X değişkeninin değerleri bir vektör olarak programa girilmiş olur. Daha sonra aşağıdaki komut yardımıyla normalleştirme işlemi gerçekleşir.

```
> data.Normalization(x,type="n1")
```

Bu komutta "n1" değişkene ilişkin gözlem değerlerinin aritmetik ortalamadan olan farklarının değişkene ilişkin standart sapma değerine bölünmesi işlemini tanımlamaktadır. Komut çalıştırıldığına aşağıdaki sonuca ulaşılır.

```
[1] -0.3116883 -1.1793613 -1.0277291 -0.3706564
[5] 1.5500176  0.5728326  0.7665848
```

Sonuçların elle yapılan hesaplamalar ile aynı olduğu görülebilir.

Ondalık Ölçekleme

Ondalık ölçekleme yönteminde değişkene (özellik) ilişkin gözlem değerlerinin ondalık bölümü hareket ettirilerek normalleştirme gerçekleştirilir. Hareket ettirilecek ondalık bölüm değişkenin maksimum mutlak değeri ile bağlantılıdır. Dönüşürme için aşağıdaki eşitlik kullanılır.

$$X^* = \frac{X}{10^J}, \quad J = \text{enb}(|X^*|) < 1$$

Bu eşitlikte;

X^* : Dönüştürülmüş değer

X : Gözlem değerini

ifade eder.

Örnek 1'de verilen X değişkenine ilişkin gözlem değerlerini ondalık ölçekte normalleştirme yöntemini kullanarak ve $j=3$ olacak şekilde veri dönüşümünü sağlayınız.

ÖRNEK 3

X
251
148
166
244
472
356
379

Tabloda verilen X değişkenine ilişkin X^* değişkeni değerleri $X_1=251$ örnek olarak şekilde aşağıdaki gibi hesaplanır.

$$X^* = \frac{X}{10^J} = \frac{251}{10^3} = \frac{251}{1.000} = 0,251$$

Bu hesaplama göre X değerlerinin ondalık normalleştirme dönüşümü yapılmış tablosu aşağıda gibi olur.

X	X*
251	0,251
148	0,148
166	0,166
244	0,244
472	0,472
356	0,356
379	0,379

X değişkeninin değerleri sırasıyla 10, 21, 14, 29, 37 ve 45 olarak verilmiştir. z-skor normalleştirme yöntemini uygulayarak ilgili değişkene ilişkin gözlem değerlerini dönüştürünüz.



SIRA SİZDE

3

Özet



Temel değişken tiplerini tanımlamak

Bir nesnenin özelliklerinin ölçme şekline göre bir çok değişken tipi tanımlanabilir. Değişken tiplerinin aralımdaki farkların tam olarak bilinmemesi veri analizinde çeşitli problemlere yol açabilir. Birimlerin sahip olduğu özelliklerin derecesinin belirlenerek sonuçların sayısal olarak ifade edilmesine ölçme adı verilir. Ölçek kavramı ölçmeye konu olan özelliklerin sınıflanması, sıralanması, derecelenmesi ya da miktar ve derecelerin belirlenebilmesi için uyulması gereken kurallarla kısıtlamaları belirleyen ölçme aracı olarak tanımlanır. Temel değişken tipleri isimsel (nominal) değişken, iki-li (binary) değişken, sıra gösteren (ordinal) değişken, tamsayılı (integer) değişken, aralıklı ölçümlendirilmiş (interval-scaled) değişken ve oranlı ölçümlendirilmiş (ratio-scaled) değişken olarak ortaya çıkar.



Veride dönüşümler gerçekleştirmek

Bazı durumlarda orijinal veri kümelerindeki özellikler gerekli enformasyonu içerdiği hâlde veri madenciliği algoritmaları için uygun yapıda olmayabilirler. Bu durumda orijinal özelliklerinden oluşturuluran bir veya daha fazla yeni özellik orijinal özelliklerden daha faydalı olabilir. Veri dönüşümünde verilerin veri madenciliği için uygun formlara dönüştürülmesi düzeltme, bir araya getirme, genelleme, normalleştirme ve özellik oluşturma işlemleriyle gerçekleştirilir.



Veri madenciliği için veriyi hazırlamak

Veri madenciliği tekniklerinin çoğu verilerdeki kusurları göz ardi edebilmesine rağmen veri kalitesini anlamak ve iyileştirmek konusuna odaklanmak veri madenciliği çıktı kalitesini arttırr. Veri kalitesi kavramı verideki gürültü ve aykırı değerler, eksik, tutarsız veya tekrarlı verilerin varlığı ile ölçülebilir. Veri kalitesinin düşük olması verinin analiz yapan kişiyi yanılmmasına yani hedeflenen sonuca ulaşamamasına neden olur. Verilerin veri madenciliğine uygun hale getirilebilmesi kusurlarının araştırılarak giderilmesi gerekmektedir. Verilerdeki kusurların giderilmesi için birtakım ön hazırlık süreçleri uygulanır. Veri hazırlamaya ilgili olan veri temizleme, veri birleştirme, veri indirgeme ve veri dönüştürme süreçleri uygulanır. Bir çok veri madenciliği uygulaması veri hazırlama süreçlerinden sadece biri değil birden fazlasının uygulanmasını gerektirebilir.

Veri hazırlama süreçlerinden biri olan veri temizleme verideki tutarsızlıkların giderilmesi ve verideki gürültünün giderilmesi için uygulanır. Veri dönüştürme olarak normalleştirme kullanılabilir. Veri birleştirme farklı kaynaktan gelen veriyi uygun bir veri tabanında birleştirir. Veri indirgemeyle ise fazla olan bazı değişkenlerin çıkarılması, birleştirilmesi veya kümeleme yaparak veri büyüğüğünün azaltılması amaçlanır. Veri yapısına uygun olacak şekilde bu süreçlerden biri veya birkaçı veri madenciliğinden önce uygulanarak elde edilen sonuçların kalitesi, güvenilirliği ve veri madenciliği aşamasında harcanacak zaman artturılabilir.

Kendimizi Sınayalım

- 1.** Aşağıdakilerden hangisi veri madenciliğinde kullanılan değişken tiplerinden biri **değildir**?
- İsimsel değişken
 - Ondalıklı ölçümlendirilmiş değişken
 - Tam sayılı değişken
 - İkili değişken
 - Sıra gösteren değişken
- 2.** Aşağıdakilerden hangisi gözlem değerlerinin tek tek nitel kategori veya sınıflara atanması sonucu ortaya çıkan değişken tipidir?
- Tam sayılı değişken
 - İkili değişken
 - İsimsel değişken
 - Sıra gösteren değişken
 - Oranlı ölçümlendirilmiş değişken
- 3.** Aşağıdakilerden hangisi sonuçları sadece iki şekilde ortaya çıkan değişken tipidir?
- İkili değişken
 - Tam sayılı değişken
 - Sıra gösteren değişken
 - Aralıklı ölçümlendirilmiş değişken
 - İsimsel değişken
- 4.** Aşağıdakilerden hangisi veri temizlemede kullanılan temel yöntemlerden biridir?
- Veri küpü birleştirme
 - Veri indirgeme
 - Veri birleştirme
 - Eksik veri
 - Boyut indirgeme
- 5.** Aşağıdakilerden hangisi eksik veri için kullanılan stratejilerden biridir?
- Gürültülü veriyi temizlemek
 - Veri nesne veya özelliklerini elemek
 - Tutarsız veriyi düzeltmek
 - Veriyi birleştirmek
 - Veriyi indirgemek
- 6.** Aşağıdakilerden hangisi verinin tahmin edilmesi için kullanılan stratejilerinden biridir?
- Veriyi birleştirmek
 - Veriyi indirgemek
 - Veri nesne veya özelliklerini elemek
 - Gürültülü veriyi temizlemek
 - Eksik verinin el ile doldurulması
- 7.** Aşağıdakilerden hangisi veri madenciliğinde analiz edilmek istenen verideki beklenen değerden sapan aykırı değere sahip veridir?
- Tutarsız
 - Gürültülü
 - Eksik
 - Birleşmiş
 - Boyutu indirgenmiş
- 8.** Veri küpü birleştirme yöntemi hangi veri hazırlama sürede kullanılır?
- Veri indirgeme
 - Veri temizleme
 - Veri dönüştürme
 - Veri sıkıştırma
 - Veri birleştirme
- 9.** Aşağıdakilerden hangisi R yazılımında gözlem değerlerinin minimum değerden olan farklarının maksimum değere minimum değer arasındaki farka bölen normalleştirme işlemini yapan işlevi tanımlar?
- n0
 - n1
 - n2
 - n3
 - n4
- 10.** Minimum değeri 560 maksimum değeri 720 olan bir değişkenin, 600 değerini enk-enb normalleştirme yöntemine göre dönüşümü sonucu kaçtır?
- 0,75
 - 0,50
 - 0,25
 - 0,50
 - 0,75

Kendimizi Sınavalım Yanıt Anahtarı

1. b Yanınız yanlış ise “Temel Değişken Tipleri” konusunu yeniden gözden geçiriniz.
2. c Yanınız yanlış ise “Temel Değişken Tipleri” konusunu yeniden gözden geçiriniz.
3. a Yanınız yanlış ise “Temel Değişken Tipleri” konusunu yeniden gözden geçiriniz.
4. d Yanınız yanlış ise “Veri Hazırlama” konusunu yeniden gözden geçiriniz.
5. b Yanınız yanlış ise “Veri Hazırlama” konusunu yeniden gözden geçiriniz.
6. e Yanınız yanlış ise “Veri Hazırlama” konusunu yeniden gözden geçiriniz.
7. b Yanınız yanlış ise “Veri Hazırlama” konusunu yeniden gözden geçiriniz.
8. a Yanınız yanlış ise “Veri Hazırlama” konusunu yeniden gözden geçiriniz.
9. e Yanınız yanlış ise “Veri Hazırlama” konusunu yeniden gözden geçiriniz.
10. c Yanınız yanlış ise “Veri Hazırlama” konusunu yeniden gözden geçiriniz.

Sıra Sizde Yanıt Anahtarı

Sıra Sizde 1

Sıra gösteren değişken tipi de isimsel değişken tipine benzerdir. Ancak değişkenin almış olduğu değer derecesi bakımından sıraya dizilmesinde önemlilik gösteriyorsa sıra gösteren değişken söz konusu olur.

Sıra Sizde 2

Veri indirgeme yöntemleri veri küpü birleştirme, boyut indirgeme, veri sıkıştırma ve büyük sayıların indirgenmesi olarak sıralanabilir.

Sıra Sizde 3

Verilen X değişkenine ilişkin aritmetik ortalama ve standart sapma aşağıdaki gibi hesaplanır.

$$\bar{X} = \frac{\sum_{i=1}^n X_i}{n} = \frac{10 + 14 + \dots + 45}{6} = \frac{156}{6} = 26$$

$$\begin{aligned}s &= \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}} \\&= \sqrt{\frac{(10-26)^2 + (14-26)^2 + \dots + (45-26)^2}{6}} \\&= 13,535\end{aligned}$$

Verilen eşitlik yardımıyla hesaplamalar aşağıdaki tablodaki gibi yapılır.

x	$x^* = \frac{x - \bar{x}}{s}$	x^*
10	$\frac{10 - 26}{13,535}$	-1,182
14	$\frac{14 - 26}{13,535}$	-0,887
21	$\frac{21 - 26}{13,535}$	-0,370
29	$\frac{29 - 26}{13,535}$	0,222
37	$\frac{37 - 26}{13,535}$	0,813
45	$\frac{45 - 26}{13,535}$	1,404

Yararlanılan ve Başvurulabilecek-Kaynaklar

- Aydin, S. (2007). **Veri Madenciliği ve Anadolu Üniversitesi Uzaktan Eğitim Sisteminde Bir Uygulama.** Anadolu Üniversitesi, Sosyal Bilimler Enstitüsü, İşletme Anabilim Dalı, Doktora Tezi, Eskişehir.
- Berry, Michael J.A. and Gordon Linoff. (1997). **Data Mining Techniques for Marketing, Sales, and Customer Support.** John Wiley & Sons, Inc., USA.
- Bramer, M. (2007). **Principles of Data Mining.** Springer-Verlag, London.
- CRISP-DM Consortium. (2000). **CRISP-DM 1.0 Step-by-Step Data Mining Guide.** www.crisp-dm.org.
- Dunham, M.H. (2003). **Data Mining Introductory and Advanced Topics.** Pearson Education, Inc., New Jersey.
- Guidici, P. (2005). **Applied Data Mining: Statistical Methods for Business and Industry.** John Wiley and Sons Ltd., England.
- Kantardzic, M. (2003). **Data Mining: Concepts, Models, Methods, and Algorithms.** IEEE Press, New Jersey.
- Jiawei, H. and Kamber, M. (2001). **Data Mining: Concept and Techniques.** Academic Press, USA.
- Pendharkar, P.C. (2003). **Managing Data Mining Technologies in Organizations: Techniques and Applications.** Idea Group Inc., USA.
- Tan, Pang-Ning, Steinbach, M. and Kumar, V. (2006). **Introduction to Data Mining.** Pearson Education, Inc., USA.

4

Amaçlarımız

- Bu üniteyi tamamladıktan sonra;
- 🕒 Benzerlik ve uzaklık kavramlarını tanımlayabilecek,
 - 🕒 Dönüşümleri uygulayabilecek,
 - 🕒 Başlıca benzerlik ve uzaklık ölçülerini hesaplayabilecek bilgi ve becerilere sahip olabilesiniz.

Anahtar Kavramlar

- Benzerlik
- Uzaklık
- Yakınlık
- Dönüşümler
- Benzerlik ve Uzaklık Ölçüleri

İçindekiler



Benzerlik ve Uzaklık Ölçüleri

GİRİŞ

Veri madenciliği uygulamalarının büyük bir kısmında, veri kümelerinde birbirine benzer olan ya da farklı olan nesne, desen, nitelik ve olayların belirlenmesi istenmektedir. Bir diğer deyişle, veriyi oluşturan birimler arasındaki benzerliğin veya farklılığın miktarının matematiksel olarak belirlenmesi gereklidir. Benzerlik ve uzaklık ölçüleri kümleme analizi, aykırı değer tespiti ve sınıflandırma gibi veri madenciliği tekniklerinin çözümünde kullanılmasından dolayı oldukça önemlidir.

Genel bir tanımı olmamasına rağmen, iki nesne arasındaki benzerlik, iki nesnenin birbirine benzeme derecesinin sayısal bir ölçüsü olarak tanımlanabilir. Veri madenciliği çerçevesinde ise benzerlik genellikle nesnelerin özelliklerini temsil eden boyutlara sahip bir uzaklık olarak tanımlanabilir. Dolayısıyla, benzerlikler birbirine daha çok benzeyen nesne çiftleri için daha yüksektir. Benzerlikler temel olarak $[-1,1]$ arasında bir sayısal değer ile ifade edilebilmelerine rağmen, genellikle normalleştirilerek $[0,1]$ arasında ölçeklendirilirler. Bu durumda “0” nesneler arasında hiç benzerliğin olmadığını, “1” ise ilgili nesnelerin tam benzer olduklarını, bir diğer ifadeyle aynı (özdeş) nesneler olduğunu ifade eder.

İki nesne arasındaki uzaklık ise iki nesnenin birbirinden farklılık derecesinin sayısal bir ölçüsüdür. Coğulukla, uzaklık kavramı farklılık kavramının yerine kullanılmasına rağmen aslında uzaklık, farklılıkların özel bir sınıfını ifade etmek için kullanılır. Farklılık, çeşitli özelliklere dayalı olarak iki nesne arasındaki zıtlık ya da uyumsuzlukların bir ölçü olarak nitelendiğinde, uzaklık iki nesne arasındaki düzensizliğin veya bozukluğun bir ölçüsü olarak düşünülebilir. Kısaca uzaklık ölçüleri yardımıyla iki nesne arasındaki farklılığın derecesi ölçülmemektedir. Dolayısıyla, birbirine benzemeyen nesne çiftleri için farklılıklar fazla ve uzaklık ölçüsünün alacağı değer de o oranda büyük olurken birbirine daha çok benzeyen nesne çiftleri için farklılıklar daha az ve uzaklık ölçüsünün alacağı değer de o oranda küçük olacaktır. Farklılıklar kimi zaman $[0,1]$ aralığına düşecek şekilde tanımlansa da genel olarak $[0,\infty)$ aralığındadır. **Uzaklık** için izleyen tanım yazılabilir.

İki nesne arasındaki düzensizliğin veya bozukluğun bir ölçüsü olan **uzaklık**, farklılığın özel bir sınıfı, alt kümesidir.

Tanım (Uzaklık): X bir küme olmak üzere $d: X \times X \rightarrow \mathbb{R}$ şeklinde tanımlanan bir fonksiyon, tüm $x, y \in X$ için;

- i. $d(x, y) = 0$, $x = y$ ise (Özdeşlik)
- ii. $d(x, y) \geq 0$ (Negatif olmama)
- iii. $d(x, y) = d(y, x)$ (Simetri)

koşullarını sağlıyorsa d , X üzerinde bir *uzaklık* olarak adlandırılır. Bu koşullara ilave olarak eğer tüm $x, y, k \in X$ için;

- iv. $d(x, y) \leq d(x, k) + d(k, y)$ (Üçgen eşitsizliği)

koşulu da sağlanıyor ise d , X üzerinde *metrik uzaklık* olarak adlandırılır.

Aynı şekilde benzerlik için izleyen tanım yazılabilir.

Tanım (Benzerlik): X bir küme olmak üzere $s: X \times X \rightarrow \mathbb{R}$ şeklinde tanımlanan bir fonksiyon, tüm $x, y \in X$ için;

- i. $s(x, y) = s(y, x)$ (Simetri)
- ii. $s(x, y) = 1$, $x = y$ ise ($0 \leq s \leq 1$) (Maksimum benzerlik)

koşullarını sağlıyorsa s , X üzerinde *benzerlik* olarak adlandırılır.

Hesaplamalar sonucunda elde edilen **benzerlik değeri** arttıkça iki nesne arasındaki benzerliğin de arttığı anlaşılırken bunun tam tersine elde edilen uzaklık değeri azaldıkça bu iki nesne arasındaki benzerliğin arttığı anlaşılmaktadır.

Bu ifadelerden yola çıkarak iki nesne arasındaki benzerlik $s(x, y)$ olarak tanımlandığında, ilgili iki nesne arasındaki uzaklık $d(x, y) = 1 - s(x, y)$ olarak tanımlanır. Veri madenciliği çalışmalarında benzerlik ve uzaklık kavramlarının ortak ifadesi olarak *yakınlık* ifadesi de kullanıldığından dolayı ünitenin izleyen kesimlerinde bu ifadeye de yer verilecektir.

İki nesne arasındaki yüksek **benzerlik değeri** nesnelerin benzer olduklarını, yüksek uzaklık değeri ise nesnelerin benzer olmadığını ifade eder.

SIRA SİZDE

1

A ve B nesnelerinin ortak özelliklerinin sırasıyla az, çok ve aynı(özdeş) olması durumunda bu iki nesnenin benzerlikleri hakkında ne söylenebilir?

DÖNÜŞÜMLER

Dönüşümler genellikle benzerlik ve uzaklıklara ilişkin ölçüm değerlerinin birbirlerine dönüştürülmesinde veya her ikisi için farklı aralıklarda elde edilmiş ölçüm değerlerinin $[0,1]$ gibi belirli bir aralık içerisinde ölçeklendirilmesi amacıyla kullanılırlar. Örneğin, elde edilen benzerlik ölçüm değerleri $[1,100]$ aralığında yer alınsın. Bir bilgisayar yazılımı aracılığı ile ilgili benzerlik değerleri kullanarak sınıflama veya kümeleme gibi farklı analizler yapmak istenebilir. Bilgisayar yazılımının kullandığı algoritma gereği sadece uzaklık ölçüm değerleri elde edildiyse veya $[0,1]$ aralığına standartlaşırılmış benzerlik değerleri üzerinden işlem yapılmışsa bu tür durumlarda istenilen değerleri elde etmek için dönüşüm yapmak durumunda kalınabilir.

Birçok veri madenciliği uygulamasında özellikle benzerlik ölçüm değerlerinin $[0,1]$ aralığında tanımlanmış veya bu aralıktaki değerlere dönüştürülmüş olması beklenir. Sonlu bir aralıktaki değerler alan benzerlik ölçüm değerleri $[0,1]$ aralığına uyacak şekilde dönüştürmek istendiğinde,

$$s' = \frac{s - enk(s)}{enb(s) - enk(s)} \quad (4.1)$$

eşitliğinden yararlanılır. Eşitlik yardımıyla elde edilecek s' değeri, dönüştürülmesi istenilen s benzerlik ölçüm değerinin $[0,1]$ aralığına düşen değerini ifade eder.

Nesneler arasında 1 hiç benzerliğin olmadığını, 100 ise tam benzerliğin olduğunu göstermek üzere elde edilmiş olan 1, 30, 45, 70 ve 100 benzerlik değerlerinin [0,1] aralığına düşecek şekilde dönüşüm yapılmış karşılıklarını elde ediniz.

ÖRNEK 1

Her bir değerin [0,1] aralığındaki karşılığı 4.1 eşitliği yardımıyla izleyen biçimde elde edilir.

$$s_1 = 1 \text{ için } s'_1 = \frac{1-1}{100-1} = \frac{0}{99} = 0$$

$$s_2 = 30 \text{ için } s'_2 = \frac{30-1}{100-1} = \frac{29}{99} = 0,29$$

$$s_3 = 45 \text{ için } s'_3 = \frac{45-1}{100-1} = \frac{44}{99} = 0,44$$

$$s_4 = 70 \text{ için } s'_4 = \frac{70-1}{100-1} = \frac{69}{99} = 0,70$$

$$s_5 = 100 \text{ için } s'_5 = \frac{100-1}{100-1} = \frac{99}{99} = 1$$

Elde edilen sonuçlara göre, [0,100] aralığında yer alan benzerlik değerlerinin tamamı [0,1] aralığında yer alacak şekilde dönüştürülmüştür.

Kimi durumlarda sonlu bir aralıkta değerler alan uzaklık ölçüm değerleri de [0,1] aralığına uyacak şekilde dönüştürülmek istenebilir. Bu durumda ise,

$$d' = \frac{d - enk(d)}{enb(d) - enk(d)} \quad (4.2)$$

eşitliğinden yararlanılır. Burada elde edilecek d' değeri, dönüştürülmesi istenilen d uzaklık değerinin [0,1] aralığına düşen değerini ifade eder.

Yakınlık ölçüm değerleri her zaman sonlu aralıkta olmayabilir. Hatırlanacağı gibi yakınlık ölçüm değerleri genellikle matematiksel olarak $[0, \infty)$ aralığında değerler almaktadır. Bu durumda yakınlık ölçüm değerlerini [0,1] sonlu aralığında ifade etmek için doğrusal olmayan bir dönüşüm uygulanır. Örnek olarak $[0, \infty)$ aralığında değerler alan bir uzaklık ölçümü için,

$$d' = \frac{d}{1+d} \quad (4.3)$$

eşitliği yardımıyla ölçüm değerleri [0,1] sonlu aralığına dönüştürülmüş olur.

Örneğin bir araştırmada ilgilenilen değişkenin $[0, \infty)$ aralığında değerler aldığı varsalım. Bu araştırmada nesneler arasındaki uzaklık değerleri 0, 2, 10, 100 ve 1000 olarak elde edilmiş olsun. Bu uzaklık değerlerine 4.3 eşitliği yardımıyla dönüşüm uyguladığımızda elde edilecek yeni uzaklık değerleri sırasıyla 0; 0,667; 0,909; 0,990 ve 0,999 olacaktır.

Son olarak benzerlik ve uzaklık değerleri arasındaki geçişlerden söz etmekte fayda bulunmaktadır. Daha önce ele alındığı gibi ölçüm değerlerinin sonlu aralıkta olup olmamasına göre iki farklı durum söz konusudur.

Benzerlik değerlerinin [0,1] sonlu aralığında olduğu ilk durumda, ilgili uzaklık değerleri,

$$d = 1 - s \quad (4.4)$$

eşitliği yardımıyla elde edilebilir. Aynı şekilde [0,1] kapalı aralığındaki uzaklık değerlerine karşı gelen benzerlik değerleri elde edilmek istediğiinde ise

$$s = 1 - d \quad (4.5)$$

eşitliğinden faydalанılır. $[0,1]$ kapalı aralığında tanımlanmış benzerlik ve uzaklık değerleri, toplamları 1 olan yakınlık ölçüleridir.

İkinci durum ise yakınlık değerlerinin sonlu aralıktaki ölçümlenmemiş olma durumudur. Örneğin bir araştırmada elde edilen uzaklık değerleri $[0,\infty)$ aralığında değerler alıyor iken istenilen benzerlik değerlerini elde edebilmek için,

$$\begin{array}{lll} \text{(a)} & \text{(b)} & \text{(c)} \\ s = \frac{1}{1+d}, & s = e^{-d} \quad \text{veya} & s = 1 - \frac{d - enk(d)}{enb(d) - enk(d)} \end{array} \quad (4.6)$$

eşitliklerinden faydalанılır. Aslında benzerlik ve uzaklık değerlerinin birbirlerine dönüştürülmüş için herhangi bir monoton azalan fonksiyon da kullanılabilir. Ancak bu dönüşümler yapılrken probleme özgü diğer faktörlerin de göz önünde bulundurulmasında yarar vardır.

ÖRNEK 2

0 tam benzerliği, 100 ise hiç benzerliğin olmadığını göstermek üzere tanımlanmış 0, 1, 10 ve 100 uzaklık değerlerinin $[0,1]$ aralığına düşecek şekilde benzerlik değerlerini 4.6'da verilen üç farklı dönüşüm eşitlikleri yardımıyla ayrı ayrı hesaplayınız.

İlk olarak $s = \frac{1}{1+d}$ dönüşümünü kullanıldığında,

$$d = 0 \text{ uzaklık değeri için } s = \frac{1}{1+d} = \frac{1}{1+0} = 1,00$$

$$d = 1 \text{ uzaklık değeri için } s = \frac{1}{1+d} = \frac{1}{1+1} = 0,50$$

$$d = 10 \text{ uzaklık değeri için } s = \frac{1}{1+d} = \frac{1}{1+10} = 0,09$$

$$d = 100 \text{ uzaklık değeri için } s = \frac{1}{1+d} = \frac{1}{1+100} = 0,01$$

değerleri elde edilir. $s = e^{-d}$ dönüşümü kullanıldığında,

$$d = 0 \text{ uzaklık değeri için } s = e^{-0} = 1$$

$$d = 1 \text{ uzaklık değeri için } s = e^{-1} = 0,37$$

$$d = 10 \text{ uzaklık değeri için } s = e^{-10} = 0,00$$

$$d = 100 \text{ uzaklık değeri için } s = e^{-100} = 0,00$$

değerleri elde edilir. Son olarak, $s = 1 - \frac{d - enk(d)}{enb(d) - enk(d)}$ dönüşümü kullanıldığında ise

$$d = 0 \text{ uzaklık değeri için } s = 1 - \frac{0 - 0}{100 - 0} = 1$$

$$d = 1 \text{ uzaklık değeri için } s = 1 - \frac{1 - 0}{100 - 0} = 0,99$$

$$d = 10 \text{ uzaklık değeri için } s = 1 - \frac{10 - 0}{100 - 0} = 0,9$$

$$d = 100 \text{ uzaklık değeri için } s = 1 - \frac{100 - 0}{100 - 0} = 0,00$$

değerleri elde edilir.

Benzerlik ve uzaklık değerlerinden birisi vasıtasyyla diğerinin hesaplanmasıında doğası gereği herhangi bir monoton azalan fonksiyon kullanılabilir.



DİKKAT

Yapılan bir çalışmada ilgilenilen değişkenin [10,100] aralığında değerler aldığı bilinmektedir ve iki nesne arasındaki uzaklık ölçüm değeri 80 olarak elde edilmiştir. Bu ölçüm değeri nin [0,1] aralığına düşen benzerlik ölçüm değeri nedir?



SIRA SİZDE

2

BASIT NİTELİKLER ARASINDAKİ YAKINLIK

Bir dizi niteliğe sahip nesnelerin yakınlığı, nesnelerin her bir niteliği için elde edilecek yakınlıklarının birleşimi olarak tanımlanır. Ancak konunun özünün daha iyi kavranabilmesi için öncelikle tek bir niteliğe sahip nesneler arasındaki yakınlığın incelenmesi gerekmektedir. Elbette ki incelenen yakınlık ölçüsünün değeri, nesnelerin sahip oldukları nitelik türüne göre farklı olacaktır.

Şayet iki nesne *sınıflayıcı* bir nitelik açısından değerlendirilmeye çalışılıyorsa bu iki nesnenin ilgili nitelik açısından aynı olup olmadıklarından başka bir bilgi verilemez. Bu durumda benzerlik değeri, nesneler ilgilenilen nitelik bakımından aynı ise "1" olurken farklı ise "0" değeri ile ifade edilir. Uzaklık değeri ise benzerliğin tam tersi şekilde ifade edilir. Yani nesneler ilgilenilen nitelik bakımından aynı ise ilgili uzaklık değeri "0" olurken farklı ise "1" değerini alır. Örneğin bir güvercin ile ari uçma yetileri bakımından karşılaştırıldığında benzerlik değeri "1" olurken büyülü bakımdan karşılaştırıldığında ise benzerlik değeri "0" olacaktır.

Bir diğer nitelik türü olan *sıralayıcı* nitelik bakımından iki nesne karşılaştırıldığında durum biraz daha karmaşıklaşır. Örneğin bir araştırmada üretilen bir ürünün kalitesinin {kötü, zayıf, orta, iyi, mükemmel} olarak değerlendirildiğini varsayıyalım. Bu şekilde nitelendirilen iki ürünün benzerliklerini ölçmek için ilk olarak niteliğin her bir sonucuna 0 veya 1'den başlamak suretiyle {kötü = 0, zayıf = 1, orta = 2, iyi = 3, mükemmel = 4} şeklinde tamsayı değerler atanır. İlgilenilen niteliğin ortaya çıkış biçimleri bu şekilde tam sayı değerlerle ifade edildikten sonra, nesneler arası uzaklık değeri

$$d(x, y) = |x - y| \quad (4.7)$$

eşitliği yardımıyla elde edilir. Elde edilen uzaklık değeri [0,1] aralığında değer alacak biçimde dönüştürülmemek istendiğinde ise

$$d(x, y) = |x - y| / (n - 1) \quad (4.8)$$

eşitliği kullanılır. Eşitlikte n , niteliğin ortaya çıktıgı sonuç sayısıdır. Dönüşümler konusunda ele alındığı üzere, uzaklık değeri yardımıyla benzerlik değeri

$$s(x, y) = 1 - d \quad (4.9)$$

eşitliği yardımıyla hesaplanır.

Örneğin bir araştırmada araştırmacının üzerinde çalıştığı X ve Y gibi iki ürün bulunduğu ve araştırmacının ilgili ürünlerin çeşitli özelliklerini bakımından değerlendirilmesi için {kötü = 0, zayıf = 1, orta = 2, iyi = 3, mükemmel = 4} değerlerini kullandığını varsayıyalım. İlgili ürünlerin kalite açısından elde edilen değerlendirme sonuçlarının sırasıyla *iyi* ve *orta* olarak tespit edildiğini varsayıduğumuzda, ürünlerin kalitelerinin değerlendirme değerleri X = 3 ve Y = 2 olacaktır. Dolayısıyla bu iki ürün arasındaki uzaklık değeri

$$d(x, y) = |3 - 2| = 1$$

olarak elde edilir. Hesaplanan bu uzaklık değeri [0,1] aralığında ifade edilmek istendiğinde

$$d(x,y) = \frac{|x-y|}{(n-1)} = \frac{|3-2|}{(5-1)} = 0,25$$

değeri elde edilir. Burada ürün özelliklerinin değerlendirilmesinde kötü, zayıf, orta, iyi ve mükemmel olmak üzere beş sonuç söz konusu olduğu için $n=5$ olur. Dolayısıyla bu iki ürün arasındaki benzerlik değeri ise

$$s(x, y) = 1 - d = 1 - 0,25 = 0,75$$

olarak elde edilir.

Son olarak *aralıklı* veya *oransal* ölçekte ölçümlenmiş bir nitelik bakımından iki nesne arasındaki uzaklık ölçüm değeri belirlenmek istendiğinde ise ölçüm değerlerinin mutlak farklarının alınması gerekmektedir. Bu durumda uzaklık ölçüm değeri

$$d(x, y) = |x - y| \quad (4.10)$$

eşitliği yardımıyla elde edilir.

Örneğin geçen yıla göre 10 kg daha ağır olduğunuzu söylediğinizde, aradaki 10 kg'lık fark değeri geçen yılı ağırlığınız ile bu yılı ağırlığınız arasındaki uzaklık değeri olacaktır. İnsan ağırlığı için tanımlı olan üst sınır hekimler tarafından yaklaşık 340 kg olarak tahmin edilmekle beraber tarih boyunca 400 kg üzerinde ağırlığa sahip insanlarla da karşılaşıldığı için ağırlık değişkeninin değişim aralığı $[0, \infty)$ aralığı olarak tanımlanabilir. Bu durumda benzerlik değeri ünitenin önceki kesimlerinde incelenen dönüşümler yardımıyla hesaplanabilir.

Yapılacak çalışmalarda X ve Y gibi iki nesne söz konusu olduğunda, nesnelerin sahip oldukları niteliğin türüne göre benzerlik $s(x, y)$ ve uzaklık $d(x, y)$ ölçüm değerlerinin nasıl hesaplanabileceği Tablo 4.1'de toplu bir şekilde verilmiştir.

Tablo 4.1
Nesnelerin Niteliklerine
Göre Benzerlik ve
Uzaklık Formülleri

Kaynak: Tan, P.N.,
Steinbach, M., &
Kumar, V. (2005).
*Introduction to Data
Mining*. Sf:69.

Nitelik Türü	Uzaklık	Benzerlik
Sınıflayıcı	$d(x, y) = \begin{cases} 0, & x = y \text{ ise} \\ 1, & x \neq y \text{ ise} \end{cases}$	$s(x, y) = \begin{cases} 1, & x = y \text{ ise} \\ 0, & x \neq y \text{ ise} \end{cases}$
Sıralayıcı	$d(x, y) = \frac{ x - y }{(n-1)}^*$	$s(x, y) = 1 - d$
Aralıklı/ Oransal	$d(x, y) = x - y $	$s(x, y) = -d, \quad s(x, y) = \frac{1}{1+d},$ $s(x, y) = e^{-d}$ $s(x, y) = 1 - \frac{d - \text{enk}(d)}{\text{enb}(d) - \text{enk}(d)}$

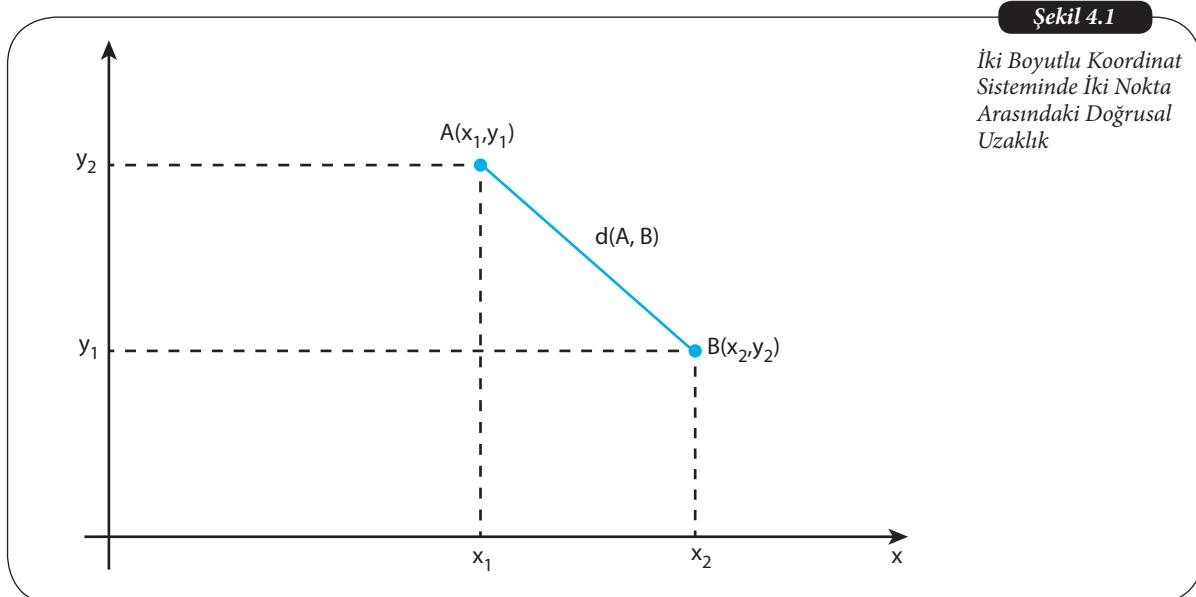
*Veriler $[0, n - 1]$ arasında tamsayılar ile eşleştirilir ki burada n veri niteliğin ortaya çıktığı sonuç sayısıdır.

BENZERLİK VE UZAKLIK ÖLÇÜLERİ

Birim ya da değişkenler arası benzerlik ya da uzaklık değerleri hesaplanırken geometrik yaklaşımlardan yararlanılır. Geometride koordinat sistemindeki iki nokta arasındaki en yakın uzaklık Pisagor bağıntısına göre elde edilir. Dolayısıyla koordinat sisteminde yer alan A ve B noktaları arasındaki doğrusal uzaklık, A noktasının koordinat değerleri $A(x_1, y_1)$ ve B noktasının koordinat değerleri $B(x_2, y_2)$ olmak üzere Pisagor bağıntısına göre;

$$d(A, B) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (4.11)$$

eşitliği yardımıyla hesaplanır. İki nokta arasındaki uzaklığın iki boyutlu uzayda grafiksel gösterimi Şekil 4.1'de verilmiştir.



Yakınlık ölçüleri, temel olarak ilgilenilen değişkenlerin nicel (sayısal) veya nitel (katagorik) olmasına göre farklılık gösterir. Değişkenlerin bu şekilde sınıflandırılmasının nedeni ise, değişkenlere ilişkin ölçüm değerlerinin matematiksel özelliklerine göre sınıflayıcı, sıralayıcı, aralıklı ve oransal olmak üzere dört **ölçek** ile ölçülmesidir. Sınıflayıcı ve sıralayıcı ölçek ile ölçülebilen değişkenler nitel, aralıklı ve oransal ölçek ile ölçülebilen değişkenler ise nicel değişkenler olarak adlandırılırlar.

NİCEL DEĞİŞKENLER İÇİN YAKINLIK ÖLÇÜLERİ

Nicel değişkenlerden elde edilen gözlem değerleri arasındaki yakınlığın belirlenmesinde Öklid uzaklığı, Karesel Öklid uzaklığı, Karl Pearson uzaklığı, Manhattan uzaklığı, Minkowski uzaklığı, Mahalanobis uzaklığı, Korelasyon uzaklığı ve Açısal benzerlik ölçülerinden yararlanılır.

ÖKLİD VE KARESEL ÖKLİD UZAKLIĞI

Uzaklık ölçüler arasında en yaygın kullanılan uzaklık ölçülerini Öklid ve Karesel Öklid uzaklık ölçüleridir. Her biri p tane nicel değişken içeren x_i ve x_j nesneleri arasındaki Öklid uzaklığı,

$$d_{ij} = \sqrt{\sum_{k=1}^p (x_{ik} - x_{jk})^2} \quad ; i = 1, 2, \dots, n \\ ; j = 1, 2, \dots, n \\ ; k = 1, 2, \dots, p \quad (4.12)$$

eşitliği yardımıyla hesaplanır. Eşitlikte,

n : nesne sayısını

p : değişken sayısını

d_{ij} : i 'inci ve j 'inci nesneler arasındaki uzaklığı

x_{ik} : i 'inci nesnenin k 'inci değişkendeki değerini

x_{jk} : j 'inci nesnenin k 'inci değişkendeki değerini

ifade eder.

Öklid uzaklılığı, i 'inci ve j 'inci nesnelerin p tane değişken için farklarının kareleri toplamının karekökü alınarak elde edilir. Öklid uzaklığı hesaplanırken veriler kullanılır. Dolayısıyla farklı ölçekler ve değişkenlerin farklı ölçü birimleri ile ölçülmüş olması, hesaplanacak uzaklık değerini etkileyecektir.

Öklid uzaklık ölçüsü, değişkenlerin birbirinden bağımsız olduklarını varsayar. Aynı zamanda L_2 norm olarak da bilinen Öklid uzaklığının hesaplanabilmesi için verilerin oransal ya da aralıklı olcekle ölçülmüş olması gereklidir. Öklid uzaklığı “*sıfır*” ile “*sonsuz*” arasında değerler alır yani tanım aralığı $[0, \infty)$ 'dur.

Karesel Öklid uzaklığı ise Öklid uzaklığının benzer biçimde hesaplanır. Tek farkı değişkenlere göre toplam uzaklığın karekök alınmadan hesaplanmasıdır. Yani Öklid uzaklığının karesidir. Karesel Öklid uzaklığı,

$$d_{ij}^2 = \sum_{k=1}^p (x_{ik} - x_{jk})^2 \quad ; i = 1, 2, \dots, n \\ ; j = 1, 2, \dots, n \\ ; k = 1, 2, \dots, p \quad (4.13)$$

eşitliği yardımıyla hesaplanır. Karesel Öklid uzaklığının hesabında karekök alınmadığından Öklid uzaklığının göre veri kümesi içerisinde yer alan aykırı değerlere (outliers) daha fazla ağırlık verme eğilimindedir. Aykırı değerler veri kümesinin genel özelliklerinden belirgin bir şekilde farklılık gösteren gözlem değerleridir. Bu farklılıkların analizler üzerinde etki gösterip göstermeyeceğine ilişkin karar araştırmacı tarafından verildikten sonra ilgili uzaklık ölçüsü hesaplanmalıdır.

ÖRNEK 3

Üç öğrencinin matematik ve istatistik derslerinden aldığı notlar Tablo 4.2'de verilmiştir. Bu iki dersten almış oldukları notlar bakımından öğrencilerin birbirlerine olan Öklid uzaklık değerlerini hesaplayınız.

Tablo 4.2
Üç Öğrencinin
Matematik ve İstatistik
Derslerinden Aldıkları
Notlar

Nesneler	Değişkenler	
	Matematik Notu	İstatistik Notu
Öğrenci I	60	80
Öğrenci II	75	55
Öğrenci III	70	45

Örneğimizde öğrenciler nesneleri, dersler ise değişkenleri ifade etmektedir. Dolayısıyla $n = 3$ ve $p = 2$ olur. Her bir öğrenci çifti için 4.12 eşitliği yardımıyla hesaplanacak Öklid uzaklık değerleri izleyen biçimde ortaya çıkacaktır.

Öğrenci I ile Öğrenci II arasındaki Öklid uzaklık değeri,

$$d_{12} = \sqrt{\sum_{k=1}^2 (x_{1k} - x_{2k})^2} = \sqrt{(60 - 75)^2 + (80 - 55)^2} = \sqrt{225 + 625} = 29,16$$

Öğrenci I ile Öğrenci III arasındaki Öklid uzaklık değeri,

$$d_{13} = \sqrt{\sum_{k=1}^2 (x_{1k} - x_{3k})^2} = \sqrt{(60 - 70)^2 + (80 - 45)^2} = \sqrt{100 + 1225} = 36,40$$

Öğrenci II ile Öğrenci III arasındaki Öklid uzaklık değeri,

$$d_{23} = \sqrt{\sum_{k=1}^2 (x_{2k} - x_{3k})^2} = \sqrt{(75 - 70)^2 + (55 - 45)^2} = \sqrt{25 + 100} = 11,18$$

olarak hesaplanır. Sonuç olarak elde edilen Öklid uzaklık değerleri incelemişinde, öğrencilerin ilgili derslerden almış oldukları notlar bakımından Öğrenci II ile Öğrenci III'ün en küçük uzaklık değerine sahip oldukları yani bu iki öğrencinin notlarının diğerlerine göre birbirine daha yakın olduğu söylenebilir. Yine hesaplanan değerler incelemişinde Öğrenci I ile Öğrenci III'ün bu çalışmada ilgilenecek olanlar bakımından birbirine en uzak öğrenciler oldukları söylenebilir.

Öklid ve Karesel Öklid Uzaklığının R Çözümü

R ile Öklid uzaklığını hesaplayabilmek için R'nin temel paketlerinden **stats** paketinde yer alan **dist()** fonksiyonundan yararlanılır. Her ne kadar R yazılımı çalıştırıldığında ilgili paket otomatik olarak yükleniyor olsa da R'nin komut satırından **library (stats)** komutunun verilmesi ile paketin kullanıma hazır hale getirilmesi gerekebilir.

dist() fonksiyonu yardımıyla veri matrisi olarak girilen x değişkenine ait nesneler arasındaki belirli uzaklık ölçüm değerleri hesaplanabilir. **dist()** fonksiyonunun temel parametreleri, veri matrisini ifade eden **x** ve hesaplanmak istenen uzaklık ölçüsü yönteminin seçimini sağlayan **method** parametreleridir. Bu fonksiyon ile ilgili yardım için, **help** ("dist") komutundan yararlanılabilir.

<https://cran.r-project.org/web/packages/stats/>



INTERNET

Örnek 3 için **dist()** fonksiyonu yardımıyla öğrenci çiftleri arasındaki Öklid uzaklık değerlerinin hesaplanması iləşkin komut dizisi ve hesaplama sonucu izleyen biçimde ortaya çıkacaktır.

```
> library("stats")
> x <- matrix(c(60, 80, 75, 55, 70, 45), byrow=TRUE, ncol=2, nrow=3)
> oklid <- dist(x, method = "euclidean")
> oklid
      1          2
2 29.15476
3 36.40055 11.18034
```

Komut dizisinin her bir satırında yer alan “>” işaretini yeni bir satırda işlem yapıldığını göstermektedir. Komut dizisinin ilk satırında, **stats** paketi kullanıma hazır hale getirilir. Komut dizisinin ikinci satırında, ilgili veri 2 satır ve 3 sütündan oluşan bir matris şeklinde tanımlanarak, matrisi oluşturan değerler girilmektedir. Komut dizisinin üçüncü satırında, **x** veri matrisi için Öklid uzaklıklar hesaplanır ve hesaplanan uzaklık değerleri daha sonra da kullanılabilmek için “oklid” isimli değişkene atanır. Komut dizisinin son satırında ise, “oklid” değişkenin aldığı değerlerin görülebilmesi için “oklid” yazarak hesaplama sonuçları görüntülenir.

Komut dizisinin en altında elde edilen “oklid” değişkeninin değerleri, üç öğrenci arasındaki Öklid uzaklık değerlerini vermektedir. Elde edilen uzaklık değerlerinin hangi nesneler arasındaki uzaklığını ifade ettiğini belirtmek üzere “oklid” değişkeni sonuç bilgisinde, sütunda ve satırda ilgili nesnelerin sıra numaraları gösterilmektedir. R aracılığı ile elde edilen Öklid uzaklık değerleri ile Örnek 3'ün elle yapılan çözümünde elde ettiğimiz değerlerin aynı olduğu görülmektedir.

Örnek 3'te verilen veriler yardımıyla öğrencilerin Matematik ve İstatistik derslerinden almış oldukları notlar bakımından birbirlerine olan Karesel Öklid uzaklık değerlerini elde ediniz.



SIRA SİZDE

3

Karl Pearson Uzaklığı

Karl Pearson uzaklığı, Öklid uzaklığının değişkenin varyansına oranlanması ile elde edilen bir uzaklıktır. Bu özelliğinden dolayı standartlaştırılmış Öklid uzaklığı olarak da bilinmektedir. Öklid uzaklığı yaygın olarak tercih edilen bir uzaklık ölçüsü olmasına rağmen, değişkenlerin ölçü birimlerinden kolaylıkla etkilenmektedir. Dolayısıyla farklı ölçü birimlerine sahip değişkenler söz konusu olduğunda hesaplama yapmadan önce değişkenlerin standartlaştırılması gerekmektedir. Böyle bir durumda ölçü birimi farklılıklarını ortadan kaldırarak amaciyla Öklid uzaklığının standartlaştırılmış şekli olan Karl Pearson uzaklığı,

$$d_{ij} = \sqrt{\sum_{k=1}^p \left(\frac{x_{ik} - x_{jk}}{s_k^2} \right)^2} \quad (4.14)$$

eşitliği yardımıyla hesaplanır. Eşitlikte s_k^2 , k'inci değişkenin varyans değeridir. Öklid uzaklığının herbir değişkenin varyansı ile ağırlıklandırılması şeklinde hesaplanan Karl Pearson uzaklığı yardımıyla, büyük varyansa sahip değişkenlere, küçük varyansa sahip değişkenlere göre daha az ağırlık verilmektedir.

ÖRNEK 4

Örnek 3'te verilen veriler için Matematik dersi varyansının $s_1^2 = 8$ ve İstatistik dersi varyansının ise $s_2^2 = 15$ olduğu varsayıldığında öğrencilerin bu iki dersten almış oldukları notlar bakımından Karl Pearson uzaklık değerlerini hesaplayınız.

Her bir öğrenci çifti için 4.14 eşitliği yardımıyla Karl Pearson uzaklık değerleri izleyen biçimde elde edilir.

Öğrenci I ile Öğrenci II arasındaki Karl Pearson uzaklık değeri,

$$d_{12} = \sqrt{\sum_{k=1}^2 \left(\frac{x_{1k} - x_{2k}}{s_k^2} \right)^2} = \sqrt{\frac{(60-75)^2}{8} + \frac{(80-55)^2}{15}} = \sqrt{3,52 + 2,78} = 2,51$$

Öğrenci I ile Öğrenci III arasındaki Karl Pearson uzaklık değeri,

$$d_{13} = \sqrt{\sum_{k=1}^2 \left(\frac{x_{1k} - x_{3k}}{s_k^2} \right)^2} = \sqrt{\left(\frac{60-70}{8} \right)^2 + \left(\frac{80-45}{15} \right)^2} = \sqrt{1,56 + 5,44} = 2,65$$

Öğrenci II ile Öğrenci III arasındaki Karl Pearson uzaklık değeri,

$$d_{23} = \sqrt{\sum_{k=1}^2 \left(\frac{x_{2k} - x_{3k}}{s_k^2} \right)^2} = \sqrt{\left(\frac{75-70}{8} \right)^2 + \left(\frac{55-45}{15} \right)^2} = \sqrt{0,39 + 0,44} = 0,91$$

olarak elde edilir.

Manhattan (City-Block) Uzaklığı

Bir diğer sıklıkla kullanılan uzaklık ölçüsü ise Manhattan uzaklığıdır. Manhattan (City Block) uzaklığı, birimler arası farkların mutlak değerinin toplamı alınmak suretiyle

$$d_{ij} = \sum_{k=1}^p |x_{ik} - x_{jk}| \quad (4.15)$$

eşitliği yardımıyla hesaplanmaktadır.

Aynı zamanda L_1 norm olarak da bilinen Manhattan uzaklıği bir başka uzaklık ölçüsü olan Minkowski uzaklığının özel bir hâlidir. Manhattan uzaklığı, değişkenler arasında ilişkisi olmaması durumunda hesaplanması gereken bir uzaklık ölçüsüdür. Ayrıca Manhattan uzaklığının aykırı değerlere karşı hassasiyeti düşüktür.

DİKKAT

Diger uzaklıklara nazaran hesaplanması kolay olan Manhattan (City-Block) uzaklığı, değişkenler arasında yüksek derecede ilişki olması durumunda veya değişkenlerin ölçü birimleri farklı olduğunda kullanılmamalıdır.

ÖRNEK 5

Örnek 3 verilerini kullanarak öğrencilerin birbirlerine olan Manhattan uzaklık değerlerini hesaplayınız.

Her bir öğrenci çifti için 4.15 eşitliği yardımıyla hesaplanacak Manhattan uzaklık değerleri izleyen biçimde ortaya çıkacaktır.

Öğrenci I ile Öğrenci II arasındaki Manhattan uzaklık değeri,

$$d_{12} = \sum_{k=1}^2 |x_{1k} - x_{2k}| = |60 - 75| + |80 - 55| = 15 + 25 = 40$$

Öğrenci I ile Öğrenci III arasındaki Manhattan uzaklık değeri,

$$d_{13} = \sum_{k=1}^2 |x_{1k} - x_{3k}| = |60 - 70| + |80 - 45| = 10 + 35 = 45$$

Öğrenci II ile Öğrenci III arasındaki Manhattan uzaklık değeri,

$$d_{23} = \sum_{k=1}^2 |x_{2k} - x_{3k}| = |75 - 70| + |55 - 45| = 5 + 10 = 15$$

olarak elde edilir.

Manhattan (City-Block) Uzaklığının R Çözümü

R ile Manhattan uzaklığını hesaplayabilmek için R'nin temel paketlerinden **stats** paketinde yer alan **dist()** fonksiyonundan yararlanılır.

Örnek 3 için **dist()** fonksiyonu yardımıyla öğrenci ikilileri arasındaki Manhattan uzaklık değerlerinin hesaplanmasıına ilişkin komut dizisi ve hesaplama sonucu izleyen biçimde ortaya çıkacaktır.

```
> library("stats")
> x <- matrix(c(60, 80, 75, 55, 70, 45), byrow=TRUE, ncol=2, nrow=3)
> city <- dist(x, method = "manhattan")
> city
      1    2
2  40
3  45 15
```

Komut dizisinin en sonunda oluşturulan “city” değişkeni değerleri üç öğrenci arasındaki Manhattan uzaklıklarıdır. Gösterilen değerlerin Örnek 5'in çözümünde elde ettiğimiz değerler ile aynı olduğu görülmektedir.

Minkowski Uzaklılığı

n sayıda birim ve p sayıda değişken ile çalışılırken birimler yada değişkenler arasındaki uzaklıkları hesaplamak için kullanılan genel bir uzaklık ölçüsüdür. L_λ norm olarak da bilinen Minkowski uzaklığı,

$$d_{ij} = \left[\sum_{k=1}^p |x_{ik} - x_{jk}|^\lambda \right]^{1/\lambda}, \quad \lambda \geq 1 \quad (4.16)$$

eşitliği yardımıyla hesaplanır. Minkowski uzaklık ölçüsündeki λ değeri büyük ve küçük farklara verilen ağırlığı değiştirir. Farklı λ değerleri için farklı uzaklık ölçüleri elde edilebileceği için genel uzaklık ölçüsü olarak nitelendirilir. Örneğin, $\lambda = 1$ olması durumunda Manhattan (City-Block) uzaklışı elde edilirken, $\lambda = 2$ olması durumunda ise Öklid uzaklığı elde edilmektedir.

Minkowski Uzaklığının R Çözümü

R ile Minkowski uzaklığını hesaplayabilmek için R'nin temel paketlerinden **stats** paketinde yer alan **dist()** fonksiyonundan yararlanılır. Ancak Minkowski uzaklığını hesaplayabilmek için ayrıca bir p parametre değerinin girilmesi gerekmektedir. p parametresinin değeri aslında Minkowski uzaklık ölçüsünün kuvveti olan ve nesneler arası farklara verilen ağırlığı ifade eden λ değeridir.

Örnek 3 verileri için Minkowski uzaklık değerlerini hesaplarken;

- i. $p = 1$ alınması hâlinde Manhattan (City-Block) uzaklışı değerlerinin elde edilebildiği komut dizileri izleyen yapıya sahiptir.
- ii. $p = 2$ alınması hâlinde ise Öklid uzaklışı değerlerinin $p = 1$ için Minkowski uzaklık değerleri,

```
> library("stats")
> x <- matrix(c(60,80,75,55,70,45),byrow=TRUE,ncol=2,nrow=3)
> sonuc1 <- dist(x, method = "minkowski", p=1)
> sonuc1
  1    2
2  40
3 45 15
```

ii. $p = 2$ için Minkowski uzaklık değerleri,

```
> library("stats")
> x <- matrix(c(60,80,75,55,70,45),byrow=TRUE,ncol=2,nrow=3)
> sonuc2 <- dist(x, method = "minkowski", p=2)
> sonuc2
      1          2
2 29.15476
3 36.40055 11.18034
```

Komut dizilerinin son kısmında “sonuc1” olarak elde edilen değerlerin Manhattan (City-Block) uzaklışı değerleriyle ve “sonuc2” olarak elde edilen değerlerin de Öklid uzaklışı değerleriyle aynı olduğu görülür.

Pearson Korelasyon Katsayısı ve Korelasyon Uzaklığı

Doğrusal ilişki katsayısı olarak da bilinen Pearson korelasyon katsayısı, iki veya daha fazla ve en az aralıklı ölçüye uygun şekilde ölçümlenmiş n adet gözlem içeren değişkenler arasındaki doğrusal ilişkinin yönünün ve derecesinin belirlenmesinde kullanılan bir katsayıdır ve r simbolü ile gösterilir. Aynı zamanda Pearson korelasyon katsayısı iki değişkenin gözlem değerleri arasındaki benzerliğin de bir ölçüsüdür. Korelasyon katsayısının hesaplanması için değişkenlerin gözlem sayılarının eşit olması gerekmektedir. Dolayısıyla her biri n adet gözlem değeri içeren x ve y değişkenleri arasındaki benzerliği ortaya koymak amacıyla Pearson korelasyon katsayısı

$$r_{xy} = s_{xy} = \frac{\sum_{i=1}^n \left(x_i - \frac{\sum x_i}{n} \right) \left(y_i - \frac{\sum y_i}{n} \right)}{\sqrt{\sum_{i=1}^n \left(x_i - \frac{\sum x_i}{n} \right)^2 \sum_{i=1}^n \left(y_i - \frac{\sum y_i}{n} \right)^2}} \quad (4.17)$$

eşitliği yardımıyla hesaplanır. Bilindiği üzere korelasyon katsayısı [-1,+1] arasında değerler alır. -1 ve +1 değerleri incelenen iki değişken arasında tam/mükemmel bir ilişkiyi ifade ederken, 0 (sıfır) değeri ilgili değişkenler arasında hiç ilişkinin olmadığını ifade eder. Hesaplanacak katsayı değerinin eksi işaretli olması değişkenler arasında ters yönlü bir ilişki olduğunun, artı işaretli olması ise değişkenler arasında aynı yönlü bir ilişki olduğunun göstergesidir.

Bir diğer açıdan ele alındığında Pearson korelasyon katsayısı, değişkenlerin gözlem değerlerinin kendi ortalamalarından farkları alınmak suretiyle standartlaştırıldığı açısal benzerlik ölçüsüdür.

Korelasyon uzaklığı ise bir benzerlik ölçüsü olarak ele alınan Pearson korelasyon katsayısından yararlanarak, değişkenler arasındaki uzaklığını hesaplayan ve sürekli değişkenler için yaygın olarak kullanılan bir uzaklık ölçüsüdür. Korelasyon uzaklığını Pearson korelasyon katsayısını temel olarak hesaplandığı için iki değişkenin öznitelik değerleri arasındaki doğrusal ilişkinin yönü ve gücünün belirlenmesinde kullanılan bir uzaklık ölçüsüdür. Korelasyon uzaklığı

$$d_{xy} = \frac{1 - r_{xy}}{2} \quad (4.18)$$

eşitliği yardımıyla elde edilir. Her ne kadar Pearson korelasyon katsayısı [-1,+1] arasında değerleralsa da korelasyon uzaklığının değerleri [0,1] aralığında değerler almaktadır.

Bilindiği üzere, [0,1] kapalı aralığında tanımlanmış benzerlik ve uzaklık ölçüm değerlerinin toplamı 1'dir. Ancak burada bir benzerlik ölçüsü olan Pearson korelasyon katsayısı değeri ile uzaklık ölçüsü olan korelasyon uzaklığını değerinin toplamı, tanımlı oldukları aralıkların farklı olmasından dolayı 1 olmayacağından emin olmak gerekmektedir. Bu uyumsuzluğu gidermek için dönemler konusunda ele alındığı üzere uygun dönüşümün yapılması gerekmektedir.

İstatistik dersini alan 2 farklı grup öğrencinin aldığı notlar Tablo 4.3'te verilmiştir. İstatistik dersinden aldığı notlar bakımından 2 öğrenci grubu için Pearson korelasyon katsayısını ve korelasyon uzaklık değerlerini hesaplayınız.

ÖRNEK 6

	Değişkenler		<i>Tablo 4.3 İki Öğrenci Grubunun İstatistik Dersinden Almış Oldukları Notlar</i>
	A Grubu	B Grubu	
Öğrenci I	55	65	
Öğrenci II	85	55	
Öğrenci III	70	75	
Toplam	210	195	

Bu problemde İstatistik dersini alan öğrenci grupları (A ve B) birer değişken olarak değerlendirilir. İlgili öğrenci grupları (yani değişkenler) arasındaki benzerliği ifade eden korelasyon katsayısının değeri 4.17 eşitliği yardımıyla

$$\begin{aligned}
r_{AB} &= s_{AB} = \frac{\sum_{i=1}^n \left(x_{Ai} - \frac{\sum x_{Ai}}{n} \right) \left(x_{Bi} - \frac{\sum x_{Bi}}{n} \right)}{\sqrt{\sum_{i=1}^n \left(x_{Ai} - \frac{\sum x_{Ai}}{n} \right)^2 \sum_{i=1}^n \left(x_{Bi} - \frac{\sum x_{Bi}}{n} \right)^2}} \\
&= \frac{\left(55 - \frac{210}{3} \right) \left(65 - \frac{195}{3} \right) + \left(85 - \frac{210}{3} \right) \left(55 - \frac{195}{3} \right) + \left(70 - \frac{210}{3} \right) \left(75 - \frac{195}{3} \right)}{\sqrt{\left[\left(55 - \frac{210}{3} \right)^2 + \left(85 - \frac{210}{3} \right)^2 + \left(70 - \frac{210}{3} \right)^2 \right] \left[\left(65 - \frac{195}{3} \right)^2 + \left(55 - \frac{195}{3} \right)^2 + \left(75 - \frac{195}{3} \right)^2 \right]}} \\
&= \frac{-150}{\sqrt{450 \times 200}} = \frac{-150}{300} = -0,5
\end{aligned}$$

olarak elde edilir. A ve B grupları arasındaki Pearson korelasyon katsayısı $r_{AB} = -0,5$ olarak elde edilir. A ve B grupları arasında hesaplanan Pearson korelasyon katsayısı “-” olduğu için ters yönlü ve “0,5” olduğu için de orta kuvvette bir ilişki olduğu söylenir. Kısaca bu iki öğrenci grubu arasında ters yönlü ve orta kuvvette bir ilişki vardır.

$[-1, +1]$ sonlu aralığında değerler alan ve $r_{AB} = s_{AB} = -0,5$ olarak elde edilen korelasyon katsayısı (benzerlik) ölçüm değerinin, 4.1 eşitliği ile verilen uygun dönüşüm yardımıyla $[0, 1]$ aralığına uyacak şekilde dönüştürülmüş değeri

$$s' = \frac{s - \text{enk}(d)}{\text{enb}(s) - \text{enk}(s)} = \frac{(-0,5) - (-1)}{1 - (-1)} = \frac{0,5}{2} = 0,25$$

olarak elde edilir.

İstatistik dersini alan A ve B öğrenci grupları için korelasyon uzaklığı ise 4.18 eşitliği yardımıyla,

$$d_{AB} = \frac{1 - r_{AB}}{2} = \frac{1 - (-0,5)}{2} = 0,75$$

olarak elde edilir.

Sonuç olarak elde edilen benzerlik ve uzaklık değerlerine göre, İstatistik dersini alan A ve B grubu öğrencilerin alındıkları notlar bakımından pek benzer olmadıkları söylenebilir.

Pearson Korelasyon Katsayısı ve Korelasyon Uzaklığının R Çözümü

R ile Pearson Korelasyon katsayısını hesaplayabilmek için R'nin temel paketlerinden **stats** paketinde yer alan **cor()** fonksiyonundan yararlanılır.

cor() fonksiyonu yardımıyla her biri n adet gözlem değeri içeren x ve y değişkenleri sütun vektörleri arasındaki Pearson korelasyon katsayısı hesaplanır. **cor()** fonksiyonunun temel parametreleri, değişkenlerin gözlem değerleri vektörleri **x** ve **y** ile hesaplanmak istenen korelasyon katsayısı tipinin seçimi için **method** parametreleridir. Şayet **method** parametresi için herhangi bir atama yapılmazsa varsayılan olarak Pearson korelasyon katsayısı hesaplanır. Bu fonksiyon ile ilgili yardım için, **help("cor")** komutundan yararlanılabilir.

Örnek 6 için **cor()** fonksiyonu yardımıyla İstatistik dersini alan A ve B öğrenci gruplarının alındıkları notlar arasındaki Pearson korelasyon katsayısının ve buna bağlı olarak korelasyon uzaklığının hesaplanması için komut dizisi izleyen yapıya sahiptir.

```
> library("stats")
> a <- c(55, 85, 70)
> b <- c(65, 55, 75)
```

```

> korelasyon <- cor(a, b, method = c("pearson"))
> korelasyon
[,1]
[1,] -0.5
> sonuc <- (1-korelasyon) /2
> sonuc
[,1]
[1,] 0.75

```

Komut dizisinin 5. satırında elde edilen “korelasyon” değişkeninin değeri A ve B öğrenci grupları arasındaki benzerliğin bir ölçüsü olan Pearson korelasyon katsayısının, son satırında elde edilen “sonuc” değişkeninin değeri ise Pearson korelasyon katsayısi değeri ile hesaplanan korelasyon uzaklığının değeridir. R aracılığı ile elde edilen Pearson korelasyon katsayısi ve korelasyon uzaklıği değerlerinin daha önceden Örnek 6'nın çözümünde elde ettiğimiz değerler ile aynı olduğu görülmektedir.

Açısal Benzerlik (Cosine Similarity)

Açısal benzerlik, iki vektör arasındaki açı farkının kosinüsünün bu iki vektör arasındaki uzaklık olarak alınması suretiyle değişkenler arasındaki benzerliğin belirlenmesine yönelik bir benzerlik ölçüsüdür. İki vektör arasındaki açı farkı *sıfır* olduğunda yani vektörler birbirlerine paralel olduklarında kosinüs değeri 1 olurken bu iki vektör arasındaki açı farklı 90° olduğunda yani vektörler birbirlerine dik olduklarında kosinüs değeri 0 olur. Dolayısıyla elde edilen değerin 1 olması değişkenler arasında tam bir benzerliğin olduğunu, 0 olması ise değişkenlerin hiç benzerliğin olmadığını göstergesi olmaktadır.

Açısal benzerlik, özellikle belge ve çoklu ortam nesnelerinin kıyaslanması ve metin madenciliğinde kullanılmaktadır.

x ve y birer vektör olmak üzere, bu iki vektör arasındaki açının kosinüsü, dolayısıyla açısal benzerliği,

$$s_{xy} = \cos\theta = \frac{x \cdot y}{\|x\|\|y\|} = \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n x_i^2} \sqrt{\sum_{i=1}^n y_i^2}} \quad (4.19)$$

eşitliği yardımıyla hesaplanır. Eşitlikte yer alan “ $x \cdot y$ ” ifadesi x ve y vektörlerinin nokta (skaler) çarpımını ifade etmektedir. Açısal benzerlik ölçütü $[-1,1]$ aralığında değerler alır.

Açısal benzerlik iki vektör arasındaki açısal farklılığı temel aldığından dolayı vektörlerin büyüklükleri üzerinden bir bilgi sağlamayacaktır. Veri madenciliğinde özellikle kümeleme analizinde tespit edilen küme içi uyumu ölçmek için kullanılır. Açısal benzerliğin bir başka faydası ise seyrek(sparse) matrislerin olduğu veri madenciliği problemlerinde etkin bir hesaplama yöntemi olmasıdır.

İzleyen iki ayrı cümlenin birbirlerine ne oranda benzer olduklarını açısal benzerlik ile elde ediniz.

ÖRNEK 7

I. Cümle: Eskişehir'de öğrenci olmak Ankara'da öğrenci olmaktan rahattır.

II. Cümle: Öğrenci olmak hem Eskişehir'de hem Ankara'da rahattır.

Verilen cümlelerin birbirlerine benzerlikleri, her iki cümlenin de içeriği kelime sayılarının karşılaştırılması suretiyle elde edilebilir. Dolayısıyla öncelikle her iki cümlenin oluşumunda kullanılan ve birbirinden farklı olan tüm kelimelerin kaçar kez tekrarlandıklarının elde edilmesi gereklidir.

Tablo 4.4
**Örnek 7'de Verilen
 Cümlelerde Kullanılan
 Farklı Kelimeler ve
 Tekrar Sayıları**

	Eskişehir	Ankara	öğrenci	olmak	hem	rahat
Cümle I	1	1	2	2	0	1
Cümle II	1	1	1	1	2	1

Aslında her iki cümlede kullanılan birbirinden farklı kelimeleri ve bunların sayılarını içeren Tablo 4.4'ün her bir satırı (Cümle I ve Cümle II) birer vektördür. Dolayısıyla bu iki vektör arasındaki açısal benzerlik değeri 4.19 eşitliği yardımıyla,

$$\begin{aligned}
 s_{12} &= \frac{\sum_{i=1}^n x_{1i}y_{2i}}{\sqrt{\sum_{i=1}^n x_{1i}^2} \sqrt{\sum_{i=1}^n y_{2i}^2}} \\
 &= \frac{(1)(1)+(1)(1)+(2)(1)+(2)(1)+(0)(2)+(1)(1)}{\sqrt{1^2+1^2+2^2+2^2+0^2+1^2} + \sqrt{1^2+1^2+1^2+1^2+2^2+1^2}} \\
 &= \frac{7}{\sqrt{11}\sqrt{9}} = 0,70
 \end{aligned}$$

olarak hesaplanır. Elde edilen bu sonuca göre verilen iki cümle birbirine %70 oranında benzemektedir. Burada önemle vurgulanması gereken nokta, elde edilen benzerlik değerinin cümlelerin ifade ettikleri anlam açısından benzerliklerine ait bir bilgi olmadığıdır. Cümleler sadece içerdikleri kelimeler bakımından incelemiş ve bu iki cümlenin kullanılan kelimeler bakımından %70 benzer oldukları tespit edilmiştir.

Açısal Benzerlik (Cosine Similarity) R Çözümü

R ile Açısal Benzerlik (Cosine Similarity) değerini hesaplayabilmek için **lsa** paketinde yer alan **cosine()** fonksiyonundan yararlanılır. Dolayısıyla hesaplamalardan önce **library(lsa)** komutu ile paketin R'de kullanımına hazır hâle getirilmesi gerekir.

lsa paketi içerisinde yer alan **cosine()** fonksiyonu yardımıyla her biri n adet gözlem değeri içeren x ve y değişkenleri sütun vektörleri arasındaki açının kosinüsü, yani açısal benzerliği hesaplanır. **cosine()** fonksiyonunun temel parametreleri, veri vektörleri olan **x** ve **y** parametreleridir. Bu fonksiyon ile ilgili yardım için, **help("cosine")** komutundan yararlanılabilir.

INTERNET



<https://cran.r-project.org/web/packages/lia/>

Örnek 7 için **cosine()** fonksiyonu yardımıyla iki ayrı cümlenin birbirlerine ne kadar benzer olduklarını hesaplamak için açısal benzerlik komut dizisi izleyen yapıya sahiptir.

```

> library("lia")
> cumle1 <- c(1,1,2,2,0,1)
> cumle2 <- c(1,1,1,1,2,1)
> acisal <- cosine(cumle1,cumle2)
> acisal
[1,]
[1,] 0.7035265

```

Komut dizisinin sonunda elde edilen "acisal" değişkeninin değeri ikinci ve üçüncü komut satırlarında girişleri yapılan cumle1 ve cumle2 vektörleri arasındaki açısal benzerlik değeridir. R aracılığı ile elde edilen açısal benzerlik değerlerinin daha önceden Örnek 7'nin çözümünde elde ettiğimiz değer ile aynı olduğu görülmektedir.

Mahalanobis Uzaklılığı

Sürekli değişkenler arasındaki yakınlığın belirlenmesinde kullanılan bir diğer ölçü ise mahalanobis uzaklığıdır. Bu uzaklık ölçüsü, iki vektör veya değişken arasındaki uzaklığın belirlenmesinde verilerin kovaryans yapılarını da dikkate almaktadır. Her biri n boyutlu x ve y gözlem vektörleri arasındaki mahalanobis uzaklığı,

$$d_{xy} = D^2 = (x - y)^T S^{-1} (x - y) \quad (4.20)$$

eşitliği yardımıyla hesaplanır. Burada S , $n \times n$ boyutlu örneklem ya da küme içi kovaryans matrisidir. Temel olarak bir nesnenin D dağılımının ortalamasından kaç standart sapma uzaklıkta olduğu araştırılmaktadır. Eğer ilgili nesne D 'nin ortalamasında ise bu uzaklık doğal olarak *sıfır* olacaktır.

Mahalanobis uzaklığının hesaplanabilmesi için öncelikle S örneklem kovaryans matrisinin tersi olan S^{-1} matrisinin elde edilmesi gereklidir. Kimi durumlarda S^{-1} matrisini elde etmede sorun yaşanabilir. Şayet değişkenler arasında bir ilişki söz konusu değilse örneklem kovaryans matrisi S , birim matris yapısına sahip olur ki bu durumda Mahalanobis uzaklığı Öklid uzaklığına eşdeğer olur. Mahalanobis uzaklığı veri madenciliğinde özellikle kümeleme analizi ile sınıflama çalışmalarında sıkılıkla kullanılmaktadır. Ek olarak ilgilenilen veri kümesi içerisinde aykırı değerlerin varlığını araştırmak için de Mahalanobis uzaklığından faydalananır.

Mahalanobis Uzaklığının R Çözümü

Mahalanobis uzaklığının R ile hesaplanabilmesi için **stats** paketi içerisinde yer alan ***mahanobis()*** fonksiyonu kullanılmaktadır.

mahanobis() fonksiyonun temel parametreleri, uzaklık değerleri hesaplanmak istenen değerler vektörünü ifade eden ***x***, dağılımın ortalama vektörü olan ***center*** ve örneklem kovaryans matrisini ifade eden ***cov*** parametreleridir. Bu fonksiyon ile ilgili yardım için, **help("mahanobis")** komutundan yararlanılabilir.

Örneklem kovaryans matrisinin tersinin alınması işlemi içерdiği için Mahalanobis uzaklığı hesabını sadece R komutları ile ve varsayımsal bir örnek üzerinden yürütebiliriz. Bu amaçla öncelikle iki değişken ve 5 birimlik bir veriye sahip olduğumuzu varsayıyalım.

Birinci değişkenin aldığı değerler: 14, 17, 19, 12 ve 9 olsun.

İkinci değişkenin aldığı değerler ise: 35, 39, 41, 33, 28 olsun.

Bu iki değişken değerlerini bir matrise atayarak önce kovaryans matrisini elde edelim daha sonra da bu kovaryans matrisi yardımıyla ilgili iki değişken için 18 ve 40 değerlerine sahip bir nesne için Mahalanobis uzaklık değerinin R komut dizisi ve hesaplama sonucu izleyen biçimde ortaya çıkacaktır.

```
> veri <- matrix(c(14,17,19,12,9,35,39,41,33,28), nrow=5,
  ncol=2, byrow=F)
> veri
 [,1] [,2]
[1,] 14 35
[2,] 17 39
[3,] 19 41
[4,] 12 33
[5,] 9 28
> S <- var(veri)
> S
 [,1] [,2]
```

```
[1,] 15.7 20.2
[2,] 20.2 26.2
> mahalanobis(c(18,40),apply(veri,2,mean),S)
[1] 0.9575758
```

R sonucuna göre ilgili problemde ilgilenilen değişkenler için ölçüm sonucu elde edilen bir nesnenin ilgili veri kümесinin merkezine olan Mahalanobis uzaklık değeri yaklaşık olarak 0,96 olmaktadır. Aynı nesnenin veri kümeseinde yer alan her bir nesneye olan Mahalanobis uzaklıklar ise yine R yardımıyla hesaplanabilir. Bu durumda girilmesi gereken komut ve sonucu izleyen şekilde olacaktır.

```
> mahalanobis(veri,c(18,40),S)
[1] 1.1212121 0.4545455 0.4545455 4.7575758 6.0000000
```

R çıktısından görülebileceği gibi yeni nesne ölçüm değerleri veri kümeseinde yer alan iki ve üçüncü nesnelere aynı uzaklıkta yer almaktadır. Dolayısıyla yeni nesne ilgili değişkenler bakımından en çok bu iki nesneye benzemektedir.

Ancak veri madenciliği problemlerinde problemin bir bütün hâlinde ele alınması gerekmektedir. Dolayısıyla veri kümeseinde yer alan tüm nesnelerin veri kümese merkezine ya da ortalamasına olan Mahalanobis uzaklıklarının elde edilmesi için girilmesi gereken komut ve hesaplama sonucu izleyen biçimdedir.

```
> mahalanobis(veri,apply(veri,2,mean),S)
[1] 0.01818182 0.68484848 2.13939394 2.20000000 2.95757576
```

Dolayısıyla veri kümесinin merkezi göz önüne alındığında iki numaralı nesnenin merkeze ya da ortalamaya olan uzaklığı 0,68, üç numaralı nesnenin merkeze ya da ortalamaya olan uzaklışı 2,13 olmaktadır. Aynı şekilde yeni nesnenin de veri kümесinin merkezine olan uzaklığını 0,96 olarak hesapladığımız hatırlanırsa yeni nesnenin - veri kümesi geneli düşünüldüğünde - ikinci nesneye daha çok benzediği söylenebilir.

İKİ SONUÇLU (BINARY) DEĞİŞKENLER İÇİN YAKINLIK ÖLÇÜLERİ

İki sonuçlu (binary) değişkenler, ölçüm değerleri sınıflama yoluyla elde edilen nitel değişkenlerdir. Bu değişkenler sadece evet/hayır, var/yok, erkek/kadın, doğru/yanlış gibi değerler alırlar. İki sonuçlu değişkenler için benzerlik veya uzaklık ölçüm değerlerin hesaplanabilmesi için her bir nesne incelenen değişkenlere ilişkin aldığı değerlerden oluşan bir vektör şeklinde ifade edilir. İki sonuçlu değişkenler içeren gözlem çiftleri arasındaki yakınlığın belirlenmesinde Öklid, Karesel Öklid, Büyüklük Farkı (Size Difference), Örütü Farkı (Pattern Difference), Lance ve Williams Uzaklık Ölçüsü, Biçim Farkı (Shape Difference) ve Jaccard Benzerliği (Jaccard Similarity) gibi birçok benzerlik ya da uzaklık ölçülerinden yararlanılmaktadır. Bu ölçüler temel olarak eşleştirmeye dayanan ölçüler oluplarından, hesaplama yapmadan önce kontenjans ya da diğer adıyla çapraz sınıflama tablosunun oluşturulması gereklidir. İki yönlü sınıflama tablosu olarak da adlandırılan kontenjans tablosu, iki sonuçlu değişkenler içeren nesne çiftinin karşılıklı eşleşen değerlerinin tekrar sayılarından oluşan tablodur. Kontenjans tablosunda herhangi bir değişkenin varlığı “1” ya da “+” ile, yokluğu ise “0” ya da “-” ile gösterilir. Buna göre iki sonuçlu nesne çifti için düzenlenen kontenjans tablosu Tablo 4.5’te verilmiştir.

		i. nesne			
		Değişken	yok (-)	var (+)	Toplam
j. nesne	Değişken	yok (-)	a	b	a + b
	var (+)	c	d	c + d	
	Toplam	a + c	b + d	p = a + b + c + d	

Tablo 4.5
İki Sonuçu İki Nesne
İçin Kontenjans Tablosu

Tablo 4.5'te verilen kontenjans tablosunda;

a değeri: i ve j nesnelerinin her ikisinde de ilgilenilen değişkenin olmadığı yani yok olduğu durum (0-0 eşleşmesi) sayısını,

b değeri: ilgilenilen değişkenin i nesnesinde var olduğu ve j nesnesinde olmadığı durum (1-0 eşleşmesi) sayısını,

c değeri: ilgilenilen değişkenin i nesnesinde olmadığı ve j nesnesinde var olduğu durum (0-1 eşleşmesi) sayısını,

d değeri: i ve j nesnelerinin her ikisinde de ilgilenilen değişkenin var olduğu durum (1-1 eşleşmesi) sayısını,

p değeri: değişken sayısını göstermektedir.

Basit Eşleştirme Katsayısı ve Uzaklılığı

Basit eşleştirme katsayısı, p tane değişken açısından ilgilenilen nesnelerin her ikisinde de olmama (0-0) ve olma (1-1) durum sayılarının oranını gösteren bir benzerlik ölçüsüdür. Diğer bir anlatımla, tesadüfi olarak seçilen bir değişkende her iki nesnenin de aynı özelliğe sahip olma olasılığını veren bir katsayıdır. Basit eşleştirme katsayısı,

$$s_{ij} = \frac{a+d}{a+b+c+d} \quad (4.21)$$

eşitliği ile hesaplanır. Bu katsayı [0,1] arasında değerler almakta ve üst sınır olan 1 değeri nesnelerin birbirlerine tam benzer olduğunu ifade etmektedir. Elde edilen bu benzerlik ölçüsünden yola çıkılarak basit eşleştirme uzaklığı ise

$$d_{ij} = 1 - s_{ij} = \frac{b+c}{a+b+c+d} \quad (4.22)$$

eşitliği yardımıyla elde edilir.

Elma ve muz meyvelerini yuvarlaklık, tatlılık, mayhoşluk ve gevreklik özelliklerinin var olup olmama durumuna göre var(+) / yok(-) şeklinde değerlendiriniz ve bu iki nesneye ilişkin basit eşleştirme katsayısını ve basit eşleştirme uzaklığını elde ediniz.

ÖRNEK 8

	Yuvarlaklık	Tatlılık	Mayhoşluk	Gevreklik
Elma	var(+)	var(+)	var(+)	var(+)
Muz	yok(-)	var(+)	yok(-)	yok(-)

Tablo 4.6
Elma ve Muzun Dört
Değişken Açısından
Değerlendirilmesi

İlk olarak bu iki nesne için herhangi bir hesaplama yapmadan eldeki verilerin kontenjans tablosu halinde düzenlenmesi gereklidir. Buna göre, düzenlenen kontenjans tablosu Tablo 4.7'deki gibi olacaktır.

Tablo 4.7
Elma ve Muz Nesneleri
İçin Kontenjans Tablosu

		Elma		Toplam
		Yok (-)	Var (+)	
Muz	Yok(-)	a = 0	b = 3	3
	Var(+)	c = 0	d = 1	1
	Toplam	0	4	4

Tablo 4.7'de verilen kontenjans tablosu yardımıyla bir benzerlik ölçüsü olan basit eşleştirme katsayısı 4.21 eşitliği yardımıyla,

$$s_{ij} = \frac{a+d}{a+b+c+d} = \frac{0+1}{0+3+0+1} = \frac{1}{4} = 0,25$$

olarak elde edilir. Dolayısıyla elma ile muz arasında belirlenen özellikler açısından benzerlik ölçüsünün değeri 0,25'tir. Buradan da bu iki meyve arasındaki basit eşleştirme uzaklığının değeri ise 4.22 eşitliği yardımıyla,

$$d_{ij} = 1 - s_{ij} = 1 - \frac{1}{4} = \frac{3}{4} = 0,75$$

olarak elde edilir. Elde edilen bu sonuçlara göre elma ve muz meyvelerinin incelenen özelikler bakımından birbirlerine pek benzer olmadıkları söylenebilir.

Basit Eşleştirme Katsayısı ve Uzaklığı R Çözümü

R ile Basit Eşleştirme Katsayısı değerini hesaplayabilmek için *scrim* paketinde yer alan *smc()* fonksiyonundan yararlanılır. Dolayısıyla hesaplamları gerçekleştirebilmek için öncelikle *library(scrim)* komutu ile paketin R'de kullanıma hazır hale getirilmesi gerekir.

scrim paketi içerisinde yer alan *smc()* fonksiyonu yardımıyla iki sonuçlu değerler alan değişkenler arasındaki basit eşleştirme katsayısının değeri hesaplanır. *smc()* fonksiyonunun temel parametreleri, satırlarında nesnelerin ikili sonuçlarının girildiği veri matrisi *x* ve benzerlik veya uzaklık değerinden hangisinin hesaplanacağı seçimi için *dist* parametreleridir. Bu fonksiyon ile ilgili yardım için, *help("smc")* komutundan yararlanılabilir.

INTERNET



<https://cran.r-project.org/web/packages/scrim/>

smc() fonksiyonu yapısı gereği veri girişinde "0" değerine izin vermediğinden veri matrisinin değerleri, nesnelerde ilgilenilen özellik var ise "1", yok ise "2" olarak girilmiştir.

Örnek 8 için *smc()* fonksiyonu yardımıyla elma ve muzun dört farklı özellik için benzerlik ölçüm değerini veren basit eşleştirme katsayısı değerini elde etmek için girilmesi gereken komut dizisi ve hesaplama sonucu izleyen yapıya sahip olacaktır.

```
> library("scrim")
> x <- matrix(c(1,2,1,1,1,1,2,1,2), nrow=2)
> basit <- smc(x)
> basit
[,1] [,2]
[1,] 1.00 0.25
[2,] 0.25 1.00
```

Komut dizisinin en altında elde edilen "basit" değişkeninin köşegen haricindeki 0,25 değeri elma ve muz nesnelerinin basit eşleştirme katsayı değерidir. Basit eşleştirme uzaklık değerini elde edebilmek için *smc()* fonksiyonunun yazımı,

```
> uzaklik <- smc(x, dist=TRUE)
> uzaklik
[,1] [,2]
[1,] 0.00 0.75
[2,] 0.75 0.00
```

şeklinde olmalıdır. Burada elde edilen “uzaklık” değişkeninin köşegen haricindeki 0,75 değeri ise elma ve muz için basit eşleştirme uzaklığının değeridir. R aracılığı ile elde edilen gerek basit eşleştirme katsayısı gerekse basit eşleştirme uzaklıği değerlerinin daha önce- den Örnek 8'in çözümünde elde ettiğimiz değer ile aynı olduğu görülmektedir.

Binary Öklid ve Binary Karesel Öklid Uzaklığı

Binary Öklid uzaklığı, iki sonuçlu değişkenler arasındaki yakınlığın belirlenmesinde yaygın olarak kullanılan ve tutarlı bir ölçü olduğu kabul edilen bir uzaklık ölçüsüdür. Binary Öklid uzaklığı,

$$d_{ij} = \sqrt{b+c} \quad (4.23)$$

eşitliği yardımıyla hesaplanır.

Binary Karesel Öklid uzaklığı ise Binary Öklid uzaklığının karesi alınmak suretiyle elde edilir. Dolayısıyla Binary Karesel Öklid uzaklığı ise

$$d_{ij}^2 = b + c \quad (4.24)$$

eşitliği yardımıyla hesaplanır. Her iki uzaklık ölçüsü de iki nesnenin eşleşmeyen değişken sayıları üzerinden hesaplama yapılması mantığına dayanır. Hesaplanan ilgili uzaklıklar için elde edilecek değerler $[0, \infty)$ aralığında yer alacaktır.

Örnek 8'deki veriler için Binary Öklid ve Binary Karesel Öklid uzaklık değerlerini hesaplayalım.

ÖRNEK 9

Elma ve muzun dört farklı özelliği göz önünde bulundurularak Tablo 4.7'de düzenlenen kontenjans tablosundaki değerlere göre Binary Öklid uzaklığı 4.23 eşitliği yardımıyla,

$$d_{ij} = \sqrt{b+c} = \sqrt{3+0} = 1,73$$

olarak elde edilir. Elde edilen bu uzaklık değeri $[0,1]$ tanım aralığına uyacak şekilde dönüştürülecek olursa

$$d' = \frac{d}{1+d} = \frac{1,73}{1+1,73} = \frac{1,73}{2,73} = 0,63$$

değeri elde edilir. Binary Karesel Öklid uzaklığı ise 4.24 eşitliği yardımıyla

$$d_{ij}^2 = b+c = 3+0 = 3$$

olarak elde edilir. Benzer şekilde elde edilen Binary Karesel Öklid uzaklık değerinin de tanım aralığı $[0,1]$ olacak şekilde dönüşüm yapılırsa

$$d' = \frac{d}{1+d} = \frac{3}{1+3} = \frac{3}{4} = 0,67$$

sonucu elde edilir.

Binary Öklid ve Binary Karesel Öklid Uzaklığı R Çözümü

R ile Binary Öklid uzaklığı değerini hesaplayabilmek için *vegan* paketinde yer alan *vegdist()* fonksiyonundan yararlanılır. Dolayısıyla işlemleri yapabilmek için önce *library(vegan)* komutu ile paketin R'de kullanıma hazır hâle getirilmesi gereklidir.

vegan paketi içerisinde yer alan *vegdist()* fonksiyonu yardımıyla Binary Öklid uzaklığının değeri hesaplanır. *vegdist()* fonksiyonunun temel parametreleri, veri matrisini ifade eden *x*, hesaplanmak istenen uzaklık ölçüsü yöntemi seçimi için *method* ve veri tipini belirleyen *binary* parametreleridir. Bu fonksiyon ile ilgili yardım için, *help("vegdist")* komutundan yararlanılabilir.

INTERNET



<https://cran.r-project.org/web/packages/vegan/>

Örnek 8 için *vegdist()* fonksiyonu yardımıyla elma ve muzun ele alınan dört özelliği için Binary Öklid uzaklığı değerinin hesaplanmasıne ilişkin komut dizisi ve hesaplama sonucu izleyen biçimde ortaya çıkacaktır.

```
> library("vegan")
> x<-matrix(c(1,0,1,1,1,0,1,0), nrow=2)
> binary_oklid<-vegdist(x, method = "euclidean", binary
= TRUE)
> binary_oklid
 1
2 1.732051
> binary_karesel_oklid <- binary_oklid^2
> binary_karesel_oklid
 1
2 3
```

Komut dizisinin üçüncü satırında elde edilen “binary_oklid” değişkeninin değeri dört özellik açısından değerlendirilen elma ve muz nesneleri için hesaplanan Binary Öklid uzaklığının değeridir. Devamında elde edilen Binary Karesel Öklid uzaklık değeri ise “binary_karesel_oklid” değişkenine atanmıştır. R aracılığı ile elde edilen her iki uzaklık değerinin Örnek 9'un çözümünde elde ettiğimiz değer ile aynı olduğu görülmektedir.

Jaccard Benzerlik Katsayıısı ve Uzaklığı

Jaccard benzerlik katsayıısı özellikle ekolojik araştırmalarda belirli bir nesnenin farklı bölgelerde var olup olmadığıının belirlenmesinde kullanılmaktadır. İki nesnenin de araştırma bölgesi sınırları içerisinde var olmaması (0-0) durumu gözlem değerinin (a'nın) göz ardi edildiği durumları dikkate alarak hesaplanan bir benzerlik ölçüsüdür. Buna göre Jaccard benzerlik katsayıısı,

$$s_{ij} = \frac{d}{b+c+d} \quad (4.25)$$

eşitliği yardımıyla hesaplanır. Jaccard benzerlik katsayıısı [0,1] arasında değerler almaktadır. Jaccard benzerlik katsayıısı ile elde edilecek Jaccard uzaklık değeri ise

$$d_{ij} = 1 - s_{ij} = 1 - \frac{d}{b+c+d} \quad (4.26)$$

şeklinde elde edilir.

T.C. Gıda Tarım ve Hayvancılık Bakanlığı küçükbaş hayvanlarda rastlanan H.I. ve İ.I.T. hastalıklarının bölgесel etkileşimlerle bulaşabileceğini ve yerleşim yerlerine göre birbirlerine benzerlik gösterebileceğinden şüphelenmektedir. Bu amaç ile Eskişehir İl sınırları içerisinde birbirine komşu olan 15 küçükbaş hayvan çiftliğinde incelemeler yapılmıştır. Yapılan incelemeler sonucunda elde edilen veriler Tablo 4.8'de verilmiştir. Tablo 4.8'deki veriler için Jaccard benzerlik katsayısını ve uzaklığını hesaplayınız.

ÖRNEK 10

Çiftlik No	H.I.	i.i.t.
1	1 (var)	0 (yok)
2	1 (var)	1 (var)
3	0 (yok)	0 (yok)
4	0 (yok)	1 (var)
5	1 (var)	1 (var)
6	0 (yok)	1 (var)
7	1 (var)	0 (yok)
8	0 (yok)	0 (yok)
9	0 (yok)	0 (yok)
10	1 (var)	1 (var)
11	1 (var)	1 (var)
12	1 (var)	1 (var)
13	0 (yok)	0 (yok)
14	0 (yok)	0 (yok)
15	1 (var)	1 (var)

Tablo 4.8
Eskişehir İl
Sınırlarındaki 15
Küçükbaş Hayvan
Çiftliğinde H.I. ve İ.I.T.
Hastalıkları Tespit
Çalışması Sonuçları

Öncelikle yapılan inceleme sonuçlarının kontenjans tablosu halinde düzenlenmesi gereklidir. Tablo 4.8 verilerinden hareketle, ilgilenilen hastalıkların varlık ve yokluk sayılarını ifade edecek şekilde düzenlenecek kontenjans tablosu Tablo 4.9'da verilmiştir.

		H.I.		Toplam
		0 (yok)	1 (var)	
i.i.t.	0 (yok)	a = 5	b = 2	7
	1 (var)	c = 2	d = 6	8
Toplam		7	8	15

Tablo 4.9
H.I. ve İ.I.T.
Hastalıkları İçin
Kontenjans Tablosu

Tablo 4.9 kontenjans tablosuna göre Jaccard benzerlik katsayısı 4.25 eşitliği yardımıyla,

$$s_{ij} = \frac{d}{b+c+d} = \frac{6}{2+2+6} = \frac{6}{10} = 0,60$$

olarak elde edilir. Elde edilen Jaccard benzerlik katsayısı üzerinden Jaccard uzaklıği ise 4.26 eşitliği yardımıyla,

$$d_{ij} = 1 - s_{ij} = 1 - 0,60 = 0,40$$

olarak elde edilir.

Jaccard Benzerlik Katsayısı ve Uzaklığı R Çözümü

R ile Jaccard uzaklığı değerini hesaplayabilmek için **vegan** paketi içerisinde yer alan **vegdist()** fonksiyonundan yararlanılır.

Örnek 10 verileri için **vegdist()** fonksiyonu yardımıyla küçükbaş hayvanlarda rastlanan H.İ. ve İ.İ.T. hastalıklarının birbirlerine olan Jaccard uzaklığı için komut dizisi ve hesaplama sonucu izleyen biçimde ortaya çıkacaktır.

```
> library("vegan")
> x <- matrix c(1,0,1,1,0,0,0,1,1,1,0,1,1,0,0,0,0,0,1,1,1
,1,1,1,0,0,0,1,1), nrow=2)
> jaccard <- vegdist(x, method = "jaccard", binary = TRUE)
> jaccard
1
2 0.4
```

Komut dizisinin en altında elde edilen “jaccard” değişkeninin değeri Jaccard uzaklık değeridir. R aracılığı ile elde edilen Jaccard uzaklığı değerinin Örnek 10'un çözümünde elde ettiğimiz uzaklık değeri ile aynı olduğu görülmektedir.

Özet



Benzerlik ve uzaklık kavramlarını tanımlamak

Kısaca iki nesnenin birbirine benzeme derecesi veya iki nesnenin birbirinden farklılık derecesi olarak ifade edebileceğimiz benzerlik ve uzaklık kavramlarını temel olarak geliştirilen çeşitli ölçülerden birçok alanda yararlanılmaktadır. Özellikle nesne veya değişkenlerin sınıflandırılmasında veya kümelenmesinde bu ölçüler kullanılmaktadır. Benzerlik ve uzaklık ölçülerini temelinde yapılan tüm analizlerdeki ortak nokta birbirine benzer nitelikteki nesne ya da değişkenleri diğerlerinden ayırt etmektir.



Dönüşümleri uygulamak

Birçok alanda gereken analizleri yapabilmek ve sonuca ulaşabilmek için geliştirilen çeşitli paket programlar mevcuttur. Elbette ki farklı paket programlar farklı algoritmalar kullanabilmektedir. Dolayısıyla farklı paket programlarının girdileri de çıktıları da farklı olabilmektedir. Bunun yanı sıra farklı benzerlik ve uzaklık ölçülerini de farklı aralıklarda sonuçlar vermektedir. Tüm bu nedenlerden dolayı gerek benzerlik ve uzaklık ölçülerini birbiri cinsinden ifade edebilmek için gerekse ölçüm değerlerini belirli aralıklarda ifade edebilmek için bir takım dönüşümler kullanılmaktadır.



Başlıca benzerlik ve uzaklık ölçülerini hesaplamak

Benzerlik ve uzaklık ölçülerini temel olarak verinin tipine göre farklılaşmakta ve farklı veri türleri için farklı ölçüler hesaplanmaktadır. Verinin sahip olduğu özelilikler ve karşılaşılacak nesnenin özelliklerine bağlı olarak başlıca kullanılan benzerlik ve uzaklık ölçülerini de farklı algoritma ve hesaplama yöntemlerine sahip olmaktadır.

Kendimizi Sınayalım

- 1.** Yakınlık ölçümleri ile ilgili aşağıdaki ifadelerden hangisi **yanlıştır**?
 - a. İki nesnenin birbirine benzeme derecesinin sayısal ölçüsüne benzerlik denir.
 - b. Uzaklık ölçüleri ile iki nesne arasındaki farklılığın derecesi ölçülür.
 - c. Farklı aralıklarda elde edilmiş yakınlık ölçüm değerlerini belirli aralığa ölçeklendirmek için dönüşümlerden yararlanılır.
 - d. Birbirine çok benzeyen nesne çiftlerinde benzerlik ölçüm değeri küçüktür.
 - e. Uzaklık ve benzerlik kavramları birbirinin tam tersi kavamlardır.

- 2.** İki nesne sınıflayıcı nitelik açısından değerlendirildiğinde aşağıdakilerden hangisi doğrudur?
 - a. Benzerlik değeri uzaklık değerinden küçüktür.
 - b. Benzerlik değeri uzaklık değerinden büyüktür.
 - c. Uzaklık değeri 1'dir.
 - d. Benzerlik değeri 1'dir.
 - e. Benzerlik ve uzaklık değerleri toplamı 1'dir.

- 3.** $[20,80]$ kapalı aralığında hesaplanmış $s = 50$ benzerlik değerinin $[0,1]$ aralığındaki karşılığını bulunuz?
 - a. $s' = 0,25$
 - b. $s' = 0,40$
 - c. $s' = 0,50$
 - d. $s' = 0,60$
 - e. $s' = 0,75$

- 4.** $[0, \infty)$ aralığında değerler alan ve $d = 500$ olarak elde edilmiş uzaklık değerinin $[0,1]$ aralığına düşen karşılığı nedir?
 - a. $d' = 0,998$
 - b. $d' = 0,776$
 - c. $d' = 0,554$
 - d. $d' = 0,332$
 - e. $d' = 0,001$

- 5.** Aşağıdakilerden hangisi sürekli değişkenler için hesaplanan yakınlık ölçülerinden biri **değildir**?
 - a. Öklid uzaklığı
 - b. Basit eşleştirme uzaklığı
 - c. Mahalanobis uzaklığı
 - d. Açısal uzaklık
 - e. Karl Pearson uzaklığı

- 6.** X ve Y değişkenleri arasındaki Pearson korelasyon katsayısının değeri $r = 0,70$ ise bu değişkenler arasındaki korelasyon uzaklık değeri aşağıdakilerden hangisidir?
 - a. 0,30
 - b. 0,15
 - c. 0,42
 - d. 1,00
 - e. 0,67

- 7.** Jaccard uzaklıği için aşağıdaki ifadelerden hangisi **yanlıştır**?
 - a. $[0,1]$ kapalı aralığında değerler alır.
 - b. İki sonuçlu değişkenler için hesaplanır.
 - c. R yazılımında `vegan` paketi içerisindeki `vegdist()` fonksiyonu ile hesaplanabilir.
 - d. Kontenjans tablosunun değerleri kullanılarak hesaplanır.
 - e. L_λ norm olarak da bilinen genel bir uzaklık ölçüsüdür.

- 8.** R yazılımında `scirme` paketindeki `smc()` fonksiyonu hangi uzaklığın hesaplanması kullanılır?
 - a. Binary Öklid uzaklığı
 - b. Basit eşleştirme uzaklığı
 - c. Minkowski uzaklığı
 - d. Korelasyon uzaklığı
 - e. Manhattan uzaklığı

- 9.** Özellikle doküman ve çoklu ortam nesnelerinin kıyaslanması ve metin madenciliğinde kullanılan yakınlık ölçüsü aşağıdakilerden hangisidir?
 - a. Açısal benzerlik
 - b. Karesel Öklid uzaklığı
 - c. Karl Pearson uzaklığı
 - d. Jaccard uzaklığı
 - e. Binary Öklid uzaklığı

- 10.** Öklid uzaklığı ile ilgili aşağıdaki ifadelerden hangisi doğrudur?
 - a. Farklı ölçü birimlerine sahip değişkenler için hesaplanır.
 - b. Birimler arası farkların mutlak değeri alınmak suretiyle hesaplanır.
 - c. Öklid uzaklığı “sıfır” ile “sonsuz” arasında değerler alır.
 - d. Minkowski uzaklığının $\lambda = 1$ için özel halidir.
 - e. Sıralayıcı öbekle ölçülmüş veriler için hesaplanır.

Kendimizi Sınavalım Yanıt Anahtarı

- | | |
|-------|---|
| 1. d | Yanıtınız yanlış ise "Temel Tanım ve Kavramlar" konusunu yeniden gözden geçiriniz. |
| 2. e | Yanıtınız yanlış ise "Basit Nitelikler Arasındaki Yakınlık" konusunu yeniden gözden geçiriniz. |
| 3. c | Yanıtınız yanlış ise "Yakınlık Dönüşümleri" konusunu yeniden gözden geçiriniz. |
| 4. a | Yanıtınız yanlış ise "Yakınlık Dönüşümleri" konusunu yeniden gözden geçiriniz. |
| 5. b | Yanıtınız yanlış ise "Sürekli Değişkenler İçin Yakınlık Ölçüleri" konusunu yeniden gözden geçiriniz. |
| 6. b | Yanıtınız yanlış ise "Korelasyon Katsayısı ve Korelasyon Uzaklığı" konusunu yeniden gözden geçiriniz. |
| 7. e | Yanıtınız yanlış ise "Jaccard Benzerlik Katsayısı ve Uzaklığı" konusunu yeniden gözden geçiriniz. |
| 8. b | Yanıtınız yanlış ise "Basit Eşleştirme Katsayısı ve Uzaklığı" konusunu yeniden gözden geçiriniz. |
| 9. a | Yanıtınız yanlış ise "Açısal Benzerlik (Cosine Similarity)" konusunu yeniden gözden geçiriniz. |
| 10. c | Yanıtınız yanlış ise "Öklid ve Karesel Öklid Uzaklığı" konusunuyinden gözden geçiriniz. |

Sıra Sizde Yanıt Anahtarı

Sıra Sizde 1

Durum 1: A ve B nesnelerinin ortak özelliklerinin az olması, bu iki nesne arasındaki farklılıkların çok olduğunu dolayısıyla iki nesnenin benzerliğinin de az olduğunu ifade eder.

Durum 2: A ve B nesnelerinin ortak özelliklerinin çok olması, bu iki nesne arasındaki farklılıkların daha az olduğunu dolayısıyla iki nesnenin benzerliğinin ise fazla olduğunu ifade eder.

Durum 3: A ve B nesnelerinin ortak özelliklerinin aynı olması ise bu iki nesnenin aynı (özdeş) nesneler olduklarını dolayısıyla bu nesnelerin tamamıyla benzer olduklarını ifade eder.

Sıra Sizde 2

Öncelikle [10,100] aralığında elde edilmiş olan $d = 80$ uzaklık değerinin [0,1] aralığındaki değeri;

$$d' = \frac{d - enk(d)}{enb(d) - enk(d)} = \frac{80 - 10}{100 - 10} = \frac{70}{90} \approx 0,78$$

olarak elde edilir. İkinci olarak da bu uzaklık değerinden benzerlik değeri;

$$s = 1 - d' = 1 - 0,78 = 0,22$$

olarak elde edilir.

Sıra Sizde 3

Hatırlanacağı üzere Karesel Öklid uzaklığı, değişkenlere göre toplam uzaklığın karekökü alınmamış hali yani Öklid uzaklığının karesidir. Örnek 3 için Öklid uzaklık değerleri sırasıyla $d_{12} = 29,16$, $d_{13} = 36,40$ ve $d_{23} = 11,18$ olarak elde edilmiştir. Dolayısıyla bu değerlerin kareleri de bize Karesel Öklid uzaklıklarını verecektir. Buna göre Karesel Öklid uzaklık değerleri sırasıyla $d_{12}^2 = (29,16)^2 = 850,31$, $d_{13}^2 = (36,40)^2 = 1324,96$ ve $d_{23}^2 = (11,18)^2 = 124,99$ olarak elde edilir.

Yararlanılan ve Başvurulabilecek Kaynaklar

- Aggarwal, C.C. (2015). **Data Mining: The Textbook**. Springer, New York.
- Bandyopadhyay, S. & Saha, S. (2013). **Unsupervised Classification: Similarity Measures, Classical and Metaheuristic Approaches, and Applications**. Springer-Verlag Berlin Heidelberg.
- Cichosz, P. (2015). **Data Mining Algorithms: Explained Using R**. John Wiley & Sons, Chichester, UK.
- Everitt, B., Landau, S., Leese, M., & Stahl, D. (2011). **Cluster Analysis**. (5th ed.), John Wiley & Sons, Chichester, UK.
- Ergüt, Ö. (2011). **Uzaklık ve Benzerlik Ölçülerinin Kümeleme Sonuçlarına Etkisi**. Marmara Üniversitesi Sosyal Bilimler Enstitüsü Yüksek Lisans Tezi, İstanbul.
- Özdamar, K. (2013). **Paket Programlar ile İstatistiksel Veri Analizi**. Cilt 2, 9. Baskı, Nisan Kitabevi, Eskişehir.
- Tan, P.N., Steinbach, M., & Kumar, V. (2005). **Introduction to Data Mining**. Addison-Wesley, Boston, USA.
- R Core Team (2015). **R: A language and environment for statistical computing**. R Foundation for Statistical Computing, Vienna, Austria. URL: <http://www.R-project.org/>
- Fridolin Wild (2015). **lsa: Latent Semantic Analysis**. R package version 0.73.1. <http://CRAN.R-project.org/package=lsa>
- Holger Schwender and with a contribution of Arno Fritsch (2013). **scrime: Analysis of High-Dimensional Categorical Data such as SNP Data**. R package version 1.3.3. <http://CRAN.R-project.org/package=scrime>
- Jari Oksanen, F. Guillaume Blanchet, Roeland Kindt, Pierre Legendre, Peter R. Minchin, R. B. O'Hara, Gavin L. Simpson, Peter Solymos, M. Henry H. Stevens and Helene Wagner (2015). **vegan: Community Ecology Package**. R package version 2.3-0. <http://CRAN.R-project.org/package=vegan>

5

Amaçlarımız

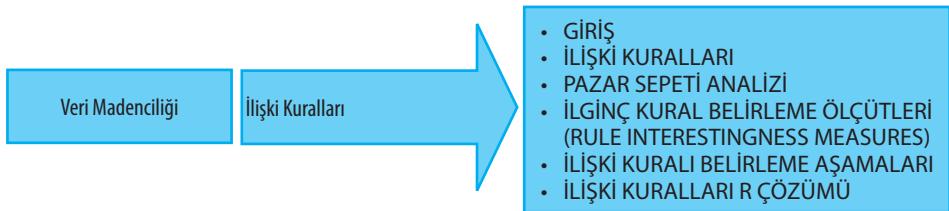
Bu üniteyi tamamladıktan sonra;

- 🕒 İlişki kurallarına ilişkin temel kavramları tanımlayabilecek,
- 🕒 Veri kümesi için güçlü ilişki kuralları elde edebilecek,
- 🕒 Apriori algoritmasının çalışma prensibini açıklayabilecek,
- 🕒 İlişki kurallarını yorumlayabilecek bilgi ve becerilere sahip olabileceksiniz.

Anahtar Kavramlar

- İlişki Kuralı
- Güçlü İlişki Kuralı
- Nesne Küme
- Pazar Sepeti Analizi
- Destek
- Güven
- Kaldırıcı
- Apriori Algoritması
- Destek Bazlı Budama

İçindekiler



İlişki Kuralları

GİRİŞ

Bilgisayar teknolojisinin gelişimine paralel olarak artan verinin büyülüğu her alanda çeşitli veri tabanlarının oluşmasına yol açmaktadır. Günümüzde birçok işletme rutin olarak büyük miktarda veri toplamakta ve depolamaktadır. Elde edilen bu verinin büyülüğu ve karmaşıklığı ilk bakışta gerekli olan bilgiye erişimi zor hale getirse de istenilen bilginin dışında daha birçok yararlı bilginin de keşfedilmesine sunmaktadır.

İLİŞKİ KURALLARI

İlişki kuralları, veri madenciliğinin tanımlayıcı modellerinden birisidir. Büyük veri küpleri içerisinde belirli veriler arasındaki ilişkileri bulan ve olayların birlikte gerçekleşme ihtimallerini geçmiş verileri analiz edip ortaya koyarak geleceğe yönelik çalışmalarını destekleyen veri madenciliği yöntemine *ilişki kuralları* denilmektedir. Genel olarak ilişki kuralları sayesinde büyük miktarlardaki veriler arasında ilginç birliktelik örüntüleri keşfederek karar verme, pazarlama ve iş yönetimi vb. gibi konularda birçok fayda sağlımaktadır. İlişki kuralları; ekonomi, eğitim, e-ticaret, pazarlama, iletişim ve sağlık gibi birçok sektörde geniş kullanıma sahip veri madenciliğinin özel bir uygulama alanıdır. İlişki kurallarının çeşitli sektörlerde kullanım amaçlarına birkaç örnek verilecek olursa,

- İletişim sektöründeki müşterilerin isteği bağlı olarak satın aldığı telesekreter, çağrı aktarma, ilave süre, internet hızı ve internet kotası vb. gibi ek hizmet kullanımı, hizmet paketleri oluşturmak amacıyla kullanılabilir.
- Finans sektöründeki kişisel hesaplar üzerindeki kullanılan krediler, yatırım yapılan ürünler vb. hareketler, müşterilerin yararlanmak isteyebileceği alternatif ürün ve hizmetlerin pazarlanmasıında kullanılabilir.
- Sigorta sektöründe alışılmışın dışında gelen sigorta talepleri bir dolandırıcılık girişiminin belirtisi olabilir. Dolayısıyla tedbir ve önlem almak amacıyla kullanılabilir.
- Sağlık sektöründe bir hastanın hastalık ve tedavi geçtiği ileride yaşaması muhtemel rahatsızlıkların belirlenmesinde ve tedbir alınmasında kullanılabilir.

Özellikle belirtmek gerekmektedir ki ilişki kuralları, günlük hayatı özellikle insanların beklenelerinin belirlenmesinde çoğu zaman başarısız olmaktadır. Örneğin, bireysel bankacılık alanında çapraz satış modellerinin belirlenmesinde ilişki kuralları iyi bir seçim değildir. Çünkü oluşturulan kurallar genellikle halihazırda uygulanan promosyon stratejileriyle birebir aynı çıkmaktadır. Bunun altında yatan neden ise zaten her müşteriye başlangıçta standart bir ürün hizmet paketinin sunulmasıdır. Farklılaşma ancak müşterilerin zaman içerisinde ilave ürün ve hizmet taleplerinin olması halinde gözlemlenebilir.

İlişki kuralları, aynı işlem içinde çoğunlukla beraber görülen nesneleri içeren kurallardır. Herhangi bir ürün alırken, bu ürünün yanında başka bir ürün ya da ürünlerin satın alınması, bu ürünler arasındaki bağlantıyı ifade eder. Bu tür bağlantıların ortaya çıkarılması ve bunun bir kural olarak değerlendirilmesi ise ilişki analizi ile mümkün olmaktadır. Literatürde bu türden çalışmalarla “*pazar sepeti analizi*” denilmektedir. Pazar sepeti analizi, müşterilerin alışveriş alışkanlıklarının veritabanındaki bilgiler aracılığı ile ortaya çıkartılması işlemidir. Müşterilerin alışveriş alışkanlıklarının ortaya çıkartılması, mağazalardaki ürünlerin yerleştirilmesine, mağaza alanının tasarlanması ve satış yapılmak üzere ürünlerin belirlenmesine yardımcı olur.

Pazar sepeti analizinde nesneler, müşteriler tarafından satın alınan ürünlerdir. Bir kaleme satılan ve içerisinde birçok nesneyi barındıran satın alma ise *işlem* veya *kayıt* olarak nitelendirilir. Dolayısıyla Pazar sepeti analizinde, bir işlemde alınan nesneler arasındaki ilişkiler incelenerek çeşitli ilişki kuralları oluşturulur. Oluşturulan bu kurallar aracılığı ile müşterilerin daha sonra yapacakları alışverişlerinde hangi mal ve hizmeti alma eğiliminde oldukları saptanarak daha fazla ürün satışını gerçekleştirebilmek için gerekli düzenlemeler yapılabilir.

İlişki kurallarının elde edilebilmesi için kullanılan birçok yöntem vardır. Büyük veri kümeleri içerisinde nesneler arasında ilişki bulmak için algoritma geliştirmek çok zor olmamasına karşın, buradaki asıl zorluk geliştirilen algoritma ile elde edilecek önemli ve önemsiz çok sayıda kural içerisinde işe yarayacak bilgiyi üreten ilişki kural(lar)ının seçilmesidir. Dolayısıyla ortaya çıkarılan ilişki kuralları içerisinde önemli veya ilginç olanları ayırt edebilmek için birtakım ölçütler ihtiyaç duyulur. Bunun için temel olarak kullanılan ölçütler destek ve güven ölçütleridir. Bir ilişki kuralı oluşturmak amacıyla yapılacak **ilişki analizinin amacı**, değerleri karar verici tarafından belirlenen destek ve güven değerlerini kısaca eşik değerlerini sağlayan kuralların elde edilmesidir.

PAZAR SEPETİ ANALİZİ

Pazar sepeti analizi, müşterilerin daha önceden yapmış oldukları alışverişlerinden oluşan veritabanından her bir alışverişinde birlikte almış olduğu ürünler arasındaki ilişkilerden yola çıkılarak müşterilerin alışveriş alışkanlıklarının belirlenmesidir. Bu sayede müşterilerin kişisel tercihlerinin belirlenmesi, birlikte satışa sunulacak ürünlerin belirlenmesi, ürün satış raflarının tasarlanması ve promosyon düzenlemeleri gibi satışa artırmaya yönelik çalışmalar daha doğru bir şekilde yapılabilmektedir.

Pazar sepeti analizinde müşterilerin alışverişlerinde aldıkları her bir ürün nesne, içerisinde birçok nesneyi yani ürünü barındıran her bir alışveriş ise işlem olarak ifade edilir. Dolayısıyla ilişki kurallarını matematiksel bir model olarak ifade edebilmek için,

$$\begin{aligned} I &= \{i_1, i_2, \dots, i_m\} \text{ nesneler (ürünler) kümesi} \\ D &= \{t_1, t_2, \dots, t_n\} \text{ işlemler (alışverişler) veritabanı} \end{aligned}$$

olarak tanımlansın. Yapılan tanımlamalara göre $t_i \subseteq I$ dir. Yani her bir işlem ya da alışveriş ifade eden t_p I nesneler kümesinin bir alt kümesidir. Veritabanında yer alan her bir alışveriş (t_i) ayrı bir numara ile ifade edilir ki bu numaralara Tid denir. Dolayısıyla D, işlemlerden oluşan veri kümesini yani tüm alışverişlerden oluşan veritabanını ifade eder.

Tanım (İlişki Kuralı): A ve B iki nesne seti olsun. Belirlenen destek eşik değeri s ve belirlenen güven eşik değeri c için $A \Rightarrow B$ şeklinde ifade edilen bir kural,

- i. A nesne setinin destek değeri *Destek* (A) $\geq s$
- ii. $A \Rightarrow B$ kuralının güven değeri *Güven* ($A \Rightarrow B$) $\geq c$

koşullarını sağlıyor ise $A \Rightarrow B$ kuralı bir *ilişki kuralı* olarak adlandırılır. Burada A *öncüil (antecedent)*, B ise *sonuç (consequent)* olarak adlandırılır ki $A \subseteq I$, $B \subseteq I$ ve $A \cap B = \emptyset$ 'dır. Yani A ve B nesne setleri, I nesneler kümesinin elemanlarından oluşan nesne setleridir ve bu nesne setlerinin ortak elemanları yoktur.

Özellikle pazarlama alanında satışları artırmabilmek amacıyla yapılan birçok çalışma bulunmaktadır. Örneğin bir marketten yapılan alışverişler üzerinden sıkılıkla birlikte alınan ürünleri belirleyebilmek için $A \Rightarrow B$ şeklinde bir ilişki kuralı oluşturulabilir. Bu amaçla $I = \{Süt, Ekmek, Yumurta, Şeker\}$ nesneler kümesi iken bu ürünleri içeren küçük bir işlemler veritabanı ise Tablo 5.1'deki gibi olsun.

Tid	Nesneler
1	Süt, Ekmek
2	Ekmek, Yumurta
3	Ekmek, Şeker
4	Süt, Ekmek, Yumurta
5	Ekmek, Yumurta

Tablo 5.1
Beş İşlemden Oluşan
Market Veritabanı
Örneği

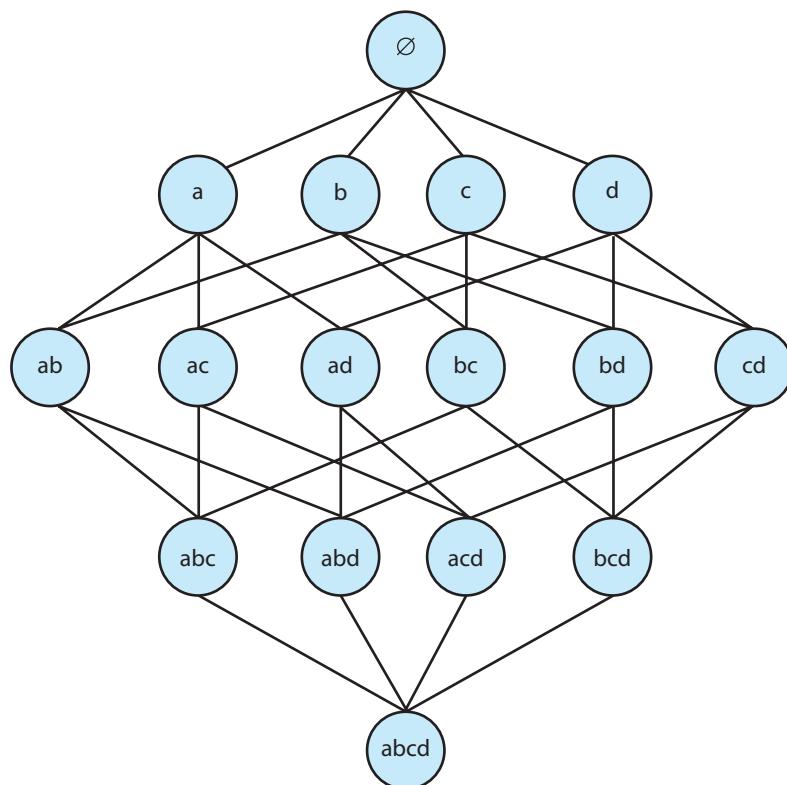
Bu marketten süt ve ekmek alan müşterilerin bunlarla birlikte çoğunlukla yumurta da aldıkları yönünde oluşturulacak bir ilişki kuralı $\{Süt, Ekmek\} \Rightarrow \{Yumurta\}$ şeklinde ifade edilir. Dolayısıyla böyle bir ilişkiye keşfeden market sahibi, süt ve ekmek alacak müşterilere yumurta fiyatında indirim yaparak veya satış reyonunda süt ve ekmeğin yanına yumurta da koymak suretiyle satışlarını artırmayı deneyebilir.

Gerek market örnek olayı gereksiz ilişki kuralının matematiksel tanımı incelendiğinde, bir ilişki kuralı oluşturmak için kullanılabilen nesne sayısının, I nesneler kümesinin birim sayısı ile sınırlı olduğu kolaylıkla anlaşılabilir. Dolayısıyla I nesneler kümesinin elemanları ile ilişki kuralı oluşturmak için kullanılabilen farklı birim sayılarına sahip öncül ve sonuç nesne setlerinin sayıları da sınırlıdır. Dolayısıyla incelenenek nesneler kümesinin sahip olduğu nesne sayısına göre oluşturulabilecek ilişki kuralı sayısı da değişmektedir.

İlişki kuralı elde etmek için ilk olarak kural oluşturmada kullanılacak nesne kümesi sayısının bilinmesi gereklidir. m adet nesne içeren bir I nesneler kümesinden elemanları birbirinden farklı oluşturulması mümkün tüm nesne setlerinin sayısı 2^m tanedir. Ancak bu nesne setlerinden bir tanesi boş kümedir ve boş küme ilişki kuralı belirlemek amacıyla kullanılamayacağından dolayı ilişki kuralı belirlemeye kullanılabilecek nesne seti sayısı $2^m - 1$ tane olur. Örneğin $m=4$ adet nesne ya da ürün içeren bir $I=\{a, b, c, d\}$ nesneler kümesinden farklı nesne sayılarına sahip, oluşturulması mümkün tüm nesne setlerinin sayısı $2^m = 2^4 = 16$ tane olur.

Şekil 5.1

$m=4$ Adet Nesne
İçeren $I=\{a,b,c,d\}$
Nesneler Kümesinden
Birbirinden Farklı
Oluşturulabilecek
Tüm Nesne Setleri



Şekil 5.1'den de görüldüğü üzere oluşturulan bu 16 nesne seti içerisinde var olan boş küme, ilişki kuralı oluşturmak amacıyla kullanılamaz. Dolayısıyla ilişki kuralı oluşturmak için kullanılabilen toplam nesne seti sayısı $2^m - 1 = 2^4 - 1 = 16 - 1 = 15$ tanedir.

İlişki kuralı oluştururken ikinci olarak, kural belirlemede kullanılabilen nesne setleri içerisindeki kaç tanesinin k tane nesne içeren nesne seti olduğunu bilinmesi gereklidir. m adet nesne içeren bir I nesneler kümesinden ilişki kuralı oluşturmada kullanılabilen k ($1 \leq k \leq m$) tane nesne içeren nesne kümelerin sayısı $C_k^m = \frac{m!}{k!(m-k)!}$ adet olacaktır.

Örneğin $m=4$ adet nesne ya da ürün içeren bir nesneler kümesinden $k=2$ nesne içeren nesne kümelerinin sayısı,

$$C_2^4 = \frac{4!}{2!(4-2)!} = \frac{4 \cdot 3 \cdot 2 \cdot 1}{2 \cdot 1 \cdot 2 \cdot 1} = 6$$

adet olacaktır.

Son olarak, oluşturulabilecek toplam ilişki kuralı sayısının bilinmesi gereklidir. m adet nesne içeren bir I nesneler kümesinden toplamda $3^m - 2^{m+1} + 1$ adet ilişki kuralı oluşturulabilir. Örneğin,

$m=5$ nesne içeren nesneler kümesinden $3^5 - 2^{5+1} + 1 = 180$ tane ilişki kuralı oluşturulabilirken,

$m=10$ nesne içeren nesneler kümesinden $3^{10} - 2^{10+1} + 1 = 57.002$ tane ilişki kuralı oluşturulabilir.

Bu iki sonuçtan anlaşılacağı üzere I nesneler kümesindeki nesne sayısı arttıkça oluşturulabilecek ilişki kuralı sayısı katlanarak artmaktadır. Dolayısıyla elde edilecek çok sayıda

ilişki kuralı içerisinde bir eleme yapmak gereklidir. Bu sebepten ötürü, oluşturulabilecek olası tüm kurallar içerisinde işe yarayacak bilgiyi üretmeyecek kural(lar)ı belirleyebilmek için, önem ve ilginin çeşitli ölçümleri üzerine kısıtlamalar konulabilir. Bu kısıtlamalar içerisinde en çok kullanılanlar ise belirlenecek destek ve güven eşik değerleridir. Oluşturabilecek tüm kurallar içerisinde bir eleme yapabilmek için her **bir kuralın** ne kadar **güçlü** olduğunun belirlenmesi gereklidir. Bunun için olası tüm kurallar için destek ve güven değerlerinin hesaplanması gereklidir.

Bir ilişki kuralının gücü, o kural için hesaplanacak destek ve güven değerleri ile ölçülebilir.

İLGİNÇ KURAL BELİRLEME ÖLÇÜTLERİ (RULE INTERESTINGNESS MEASURES)

İlgilenilen problemden ilişki kurallarını belirlemeye kullanılan nesneler kümesinin eleman sayısı arttıkça bu nesneler aracılığı ile oluşturulacak kural sayısı da katlanarak artmaktadır. Dolayısıyla bu kurallar içerisinde belirli ölçütler kullanmak suretiyle bilgi üretmek amacıyla kullanılmayacak, önemsiz kuralların elenmesi gereklidir. Bir alışveriş veritabanından oluşturulacak ilişki kuralları arasında işe yarayacak bilgiyi üretmek amacıyla kullanılacak ilişki kuralı *ilginç kural* olarak tanımlanabilir. Bir ilişki kuralının ilginç kural olarak değerlendirilebilmesi için,

- i. Daha önceden keşfedilmemiş
- ii. Eyleme dönük, bir başka ifadeyle uygulanabilir

olması gereklidir. Bir ilişki kuralının uygulanabilir olup olmadığı, ilgilenilen problemin amacı doğrultusunda konunun uzmanı olan karar verici tarafından verilen subjektif bir karardır. Dolayısıyla bir ilişki kuralının "*ilginç kural*" olarak değerlendirilmesi, problemin amacına ve karar vericinin tutumuna bağlı olarak değişebilmektedir. Elde edilen bir ilişki kuralı bir karar verici tarafından ilginç olarak değerlendirilirken, bir diğer karar verici tarafından ilginç olarak değerlendirilmeyebilir.

Bir ilişki kuralının "*ilginç*"lığı, kişiden kişiye değişiklik gösterebilen yani subjektif bir karar olmasına rağmen, bu kararın verilebilmesi için verilerden elde edilebilecek ilişki kurallarının bilimsel veya objektif ölçütler aracılığı ile de elenmesi beklenir. İlginç kuralların belirlenebilmesi amacıyla kullanılan başlıca objektif ölçütler destek ve güven ölçütleridir. İlişki kurallarının elenerek sayılarının azaltılmasında çoğu zaman destek ve güven temel ölçütleri yeterli olmasına karşın bazı durumlarda yapılan eleme sonucunda elde edilen kural sayısı da arzu edilenden fazla olabilir. Bu gibi durumlarda ilave ölçütler gereksinim duyulur. Bu amaçla geliştirilen birçok ölçüt mevcuttur. Bu ölçütler içerisinde en yaygın kullanılan öncül ve sonuç nesne setleri arasındaki korelasyonu hesaba katan kaldırıcı ölçütür.

Destek (Support)

İlginc olarak nitelendirilen ve bilgi üretmek amacıyla kullanılacak bir ilişki kuralının belirlenebilmesi için kullanılan ilk ölçüt, nesne setleri içerisinde eleme yapılmasını sağlayan destek değeridir. Karar verici tarafından belirlenmiş olan destek eşik değerine eşit veya bu değerin üzerinde destek değerine sahip nesne setleri üzerinden işlemler yürütülürken, bu değerin altında destek değerine sahip nesne setleri elenerek değerlendirme dışı bırakılır.

Bir A nesne setinin destek değeri, D işlemler veritabanında A nesne setini içeren işlem sayısının veritabanındaki tüm işlemlerin sayısına oranı şeklinde elde edilir ve

$$Destek(A) = \frac{|A|}{|D|} \quad (5.1)$$

eşitliği yardımıyla hesaplanır. Eşitlikte |A|, tüm işlemler içerisinde A nesne setini içeren işlem sayısını, |D| ise işlemler veritabanındaki tüm işlemlerin sayısını ifade eder. Aslında

Bir A nesne setinin destek değeri, aslında $P(A)$ 'dır. Yani A nesne setinin gözlenme olasılığıdır.

bir A nesne setinin destek değeri, A nesne setindeki nesnelerin veritabanındaki işlemler içerisindeki bulunma olaslığını ifade eder ve $P(A)$ şeklinde gösterilir. Destek değeri $[0,1]$ aralığında değer alır ve yüzde olarak yorumlanır.

Tablo 5.1'de daha önce verilen veritabanı için iki nesne içeren bir $A = \{\text{Süt}, \text{Ekmek}\}$ nesne setinin destek değeri 5.1 eşitliği yardımıyla,

$$\text{Destek}(A) = \text{Destek} \{ \text{Süt}, \text{Ekmek} \} = \frac{|\{\text{Süt}, \text{Ekmek}\}|}{|D|} = \frac{2}{5} = 0,40 (\%40)$$

olarak elde edilir. Veritabanında yer alan toplam 5 işlemin 2 tanesinde (1. ve 4. işlemlerde) süt ve ekmek nesneleri birlikte alınmış olduğundan, elde edilen bu destek değeri alışverişlerin %40'ında süt ve ekmeğin birlikte alınmış olduğunu ifade eder.

ÖRNEK 1

Tablo 5.1'de verilen market işlemlerine ilişkin veritabanını kullanarak, $I = \{\text{Süt}, \text{Ekmek}, \text{Yumurta}, \text{Şeker}\}$ şeklinde belirlenen nesneler kümesi için iki nesneden oluşan ($k=2$) nesne setlerinin destek değerlerini hesaplayarak sonuçları yorumlayınız.

Destek değerlerini hesaplayabilmek için öncelikle $m=4$ nesne içeren I nesneler kümesinden iki nesneden oluşan nesne setlerinin ($k=2$) belirlenmesi gereklidir. İki nesne içeren nesne seti sayısı

$$C_2^4 = \frac{4!}{2!(4-2)!} = \frac{4.3.2.1}{2.1.2.1} = 6$$

tane olacaktır. Bunlar sırasıyla, $\{\text{Süt}, \text{Ekmek}\}$, $\{\text{Süt}, \text{Yumurta}\}$, $\{\text{Süt}, \text{Şeker}\}$, $\{\text{Ekmek}, \text{Yumurta}\}$, $\{\text{Ekmek}, \text{Şeker}\}$ ve $\{\text{Yumurta}, \text{Şeker}\}$ nesne setleridir. Dolayısıyla Tablo 5.1'deki işlemler veritabanına göre bu nesne setlerinin destek değerleri ise,

$$\text{Destek} \{ \text{Süt}, \text{Ekmek} \} = \frac{2}{5} = 0,40$$

$$\text{Destek} \{ \text{Süt}, \text{Yumurta} \} = \frac{1}{5} = 0,20$$

$$\text{Destek} \{ \text{Süt}, \text{Şeker} \} = \frac{0}{5} = 0$$

$$\text{Destek} \{ \text{Ekmek}, \text{Yumurta} \} = \frac{3}{5} = 0,60$$

$$\text{Destek} \{ \text{Ekmek}, \text{Şeker} \} = \frac{1}{5} = 0,20$$

$$\text{Destek} \{ \text{Yumurta}, \text{Şeker} \} = \frac{0}{5} = 0$$

olarak elde edilir.

Hesaplanan destek değerlerine göre, marketten yapılan alışverişlerin hiçbirisi süt ve şeker nesneleri birlikte alınmamıştır. Bu durum yumurta ve şeker için de aynıdır. Çünkü her iki nesne seti için de hesaplanan destek değerleri sıfırdır.

Benzer şekilde bu marketten yapılan alışverişlerin %20'sinde ekmek ve şeker birlikte alınmışken, aynı durum süt ve yumurta için de geçerlidir.

Son olarak yapılan alışverişlerin %40'ında süt ve ekmek nesnelerinin birlikte alındığı, %60'ında ise ekmek ve yumurta nesnelerinin birlikte alındığı görülmektedir.

Tablo 5.1'de verilen beş işlemden oluşan market veritabanını kullanarak $I=\{Süt, Ekmek, Yumurta, Şeker\}$ şeklinde belirlenen nesneler kümesi için $k=1$, $k=3$ ve $k=4$ nesne içeren nesne setlerinin destek değerlerini ayrı ayrı hesaplayınız.



SIRA SİZDE

1

Bir nesne seti için destek değeri hesaplanabileceği gibi, benzer mantıkla $A \Rightarrow B$ şeklinde ifade edilen bir ilişki kuralı için de destek değeri hesaplanabilir. Bir ilişki kuralının destek değeri, D işlemler veritabanında A ve B nesne setlerini birlikte içeren işlem sayısının veritabanındaki tüm işlemlerin sayısına oranı şeklinde elde edilir ve

$$Destek(A \Rightarrow B) = \frac{|A \cup B|}{|D|} \dots \quad (5.2)$$

$A \Rightarrow B$ şeklindeki bir ilişki kuralının destek değeri, aslında $P(A \cup B)$ dir. Yani A ve B nesne setlerinin birlikte gözlenme olasılığıdır.

eşitliği ile hesaplanır. Eşitlikte $|A \cup B|$, tüm işlemler içerisinde hem A hem de B nesne setlerini birlikte içeren işlem sayısını ifade eder. Aslında **bir ilişki kuralının destek değeri**, o kuralın öncül (A) ve sonuç (B) kısmındaki nesne setlerinin birlikte gözlenme olasılığıdır ve $P(A \cup B)$ şeklinde ifade edilir.

Örneğin, Tablo 5.1'de verilen işlem veritabanı üzerinden belirlenecek $\{Süt, Ekmek\} \Rightarrow \{Yumurta\}$ şeklindeki bir ilişki kuralının destek değeri 5.2 eşitliği yardımıyla,

$$\begin{aligned} Destek(\{Süt, Ekmek\} \Rightarrow \{Yumurta\}) &= \frac{|\{Süt, Ekmek, Yumurta\}|}{|D|} \\ &= \frac{1}{5} = 0,20 (\%20) \end{aligned}$$

olarak elde edilir. Hesaplanan bu destek değeri, yapılan alışverişlerin %20'sinde süt, ekmek ve yumurtanın birlikte alınmış olduğunu ifade eder.

Destek Eşik Değeri

İlginc kural elde edebilmek için ilk eleme işlemi, hesaplanan destek değerleri üzerinden yapılır. Bu elemeyi yapabilmek için ise önceden bir destek eşik değerinin belirlenmiş olması gerekmektedir. Belirlenecek destek eşik değeri, tüm nesne setleri içerisinde bu destek eşik değerinden daha küçük destek değerine sahip nesne setlerinin elenmesini sağlar. Elenen nesne setleri, ilişki kuralı oluşturmak amacıyla kullanılmayacağından dolayı destek eşik değeri belirlenirken dikkatli olunması gerekmektedir.

Örnek 1'deki veriler için destek eşik değerinin 0,30 olarak belirlenmiş olduğunu varsayılmı. Bu durumda, bir nesne içeren nesne setleri içerisinde sadece $\{\text{Şeker}\}$ nesne setinin destek değeri 0,20 olduğu için ve destek eşik değerinden küçük olduğu için elenir. Benzer şekilde iki nesneden oluşan nesne setleri arasından ise $\{Süt, Şeker\}$, $\{Ekmek, Şeker\}$, $\{Süt, Yumurta\}$ ve $\{Yumurta, Şeker\}$ nesne setlerinin hesaplanan destek değerleri belirlenmiş olan 0,30 destek eşik değerinden daha küçük oldukları için elenerek değerlendirme dışı bırakılırlar. Son olarak üç ve dört nesneden oluşan nesne setleri içerisinde ise hiçbir nesne setinin bu destek eşik değerini aşamadığı görülür. Dolayısıyla tüm üç ve dört nesne içeren nesne setleri de elenmiş olur. Elenen bu nesne setleri ilişki kuralı oluşturmak için kullanılmazlar.

Belirlenecek destek eşik değerinin çok yüksek bir değer olması, ilginc kural elde edebilmek için ele alınacak nesne setlerinin sayısını aşırı derecede azaltacaktır. Dolayısıyla buna bağlı olarak oluşturulacak ilişki kuralı sayısı da az olacaktır. Sonuçta oluşturulan az

Belirlenen destek eşik değerine eşit veya daha büyük destek değerine sahip nesne setine **sık görülen nesne seti denir.**

saydakı ilişki kuralı ile bilgi üretmeyi sağlayacak ilginç kural ya oluşturulamayacak veya oluşturulsa dahi oluşturulan bu ilişki kuralı ile yararlı bilgi üretilemeyecek veya uygulanabilir olmayacağındır.

Aksine destek eşik değerinin çok küçük bir değer olarak belirlenmesi durumunda ise, aşırı fazla nesne setinin değerlendirilmesi gerekliliği durumu ile karşı karşıya kalınacaktır. Bu durum hem çok fazla işlem yapılmasını gerektirecek hem de ilginç olmayan birçok kuralın da ortaya çıkmasına neden olacaktır. Sonuçta kullanışlı bilgiye ulaşacaktır ilginç kural belirlenmesi güçleşecektir.

Özellikle büyük veri tabanlarında düşük destek eşik değeri belirlenerek elde edilecek birçok ilişki kuralı, karar verecek kişi için ilginç olma niteliği taşımamakta ve bilgi üretmek amacıyla kullanılamamaktadır. Bu durum, ilişki kurallarının belirlenmesinde yaşanan en büyük sorunlardan birisidir.

İlginç ilişki kuralı elde edebilmek için öncelikle nesne setlerinin destek değerleri hesaplanır. Belirlenen destek eşik değerine eşit ya da bu değerin üzerinde destek değerine sahip nesne setleri ilişki kuralları oluşturmada kullanılacak nesne setleridir. Destek eşik değerini geçen ve kural oluşturmada kullanılacak nesne setleri **sık görülen nesne setleri (frequent itemset)** olarak adlandırılır.

Güven (Confidence)

$A \Rightarrow B$ şeklindeki **bir ilişki kuralının güven değeri**, aslında A 'yı içeren işlemlerin aynı zamanda B 'yi de içermeye olasılığıdır yani $P(B | A)$ koşullu olasılığıdır. Yani A bilindiğinde B 'nin ortaya çıkma olasılığıdır.

İlginç ilişki kuralı elde edebilmek için kullanılan ikinci ölçüt, güven değeridir. Öncelikle karar verici tarafından belirlenmiş olan destek eşik değerine eşit ya da daha büyük destek değerine sahip nesne setleri yani sık görülen nesne setleri ile oluşturulması mümkün tüm ilişki kuralları oluşturulur. Karar verici tarafından belirlenmiş olan güven eşik değerine eşit ya da daha büyük güven değerine sahip ilişki kuralları ilginç kural elde etmek için değerlendirilmeye alınırken, bu değerin altında güven değerine sahip ilişki kuralları ise elenir, değerlendirilmez.

Sık görülen nesne setleri ile $A \Rightarrow B$ şeklinde oluşturulan bir ilişki kuralı için hesaplanacak güven değeri, D işlemler veritabanında A 'yı içeren ve aynı zamanda B 'yi de içeren işlemlerin sayısının sadece A 'yı içeren işlem sayısına oranıdır. Dolayısıyla $A \Rightarrow B$ şeklinde ifade edilen ilişki kuralı için güven değeri,

$$\text{Güven}(A \Rightarrow B) = \frac{\text{Destek}(A \cup B)}{\text{Destek}(A)} = \frac{|A \cup B|}{|A|} \quad (5.3)$$

eşitliği yardımıyla hesaplanır. Aslında **bir ilişki kuralının güven değeri**, o kuralın öncül(A) nesne setinin ortaya çıkması veya gözlenmesi durumunda sonuç (B) nesne setinin de ortaya çıkması, gözlenmesi olasılığıdır ve $P(B | A)$ şeklinde gösterilir. Güven değeri $[0,1]$ arasında değer alır ve yüzde olarak yorumlanır.

ÖRNEK 2

Örnek 1 verilerinden hareketle, $\{\text{Süt, Ekmek}\} \Rightarrow \{\text{Yumurta}\}$ şeklinde belirlenen bir ilişki kuralının güven değerini hesaplayınız ve sonucu yorumlayınız.

$\{\text{Süt, Ekmek}\} \Rightarrow \{\text{Yumurta}\}$ ilişki kuralının güven değeri 5.3 eşitliği yardımıyla,

$$\begin{aligned} \text{Güven}\left(\{Süt, Ekmek\} \Rightarrow \{Yumurta\}\right) &= \frac{\text{Destek}\{Süt, Ekmek, Yumurta\}}{\text{Destek}\{Süt, Ekmek\}} \\ &= \frac{1/5}{2/5} = \frac{0,20}{0,40} = 0,50 \end{aligned}$$

veya

$$\begin{aligned} \text{Güven}\left(\{Süt, Ekmek\} \Rightarrow \{Yumurta\}\right) &= \frac{|\{Süt, Ekmek, Yumurta\}|}{|\{Süt, Ekmek\}|} \\ &= \frac{1}{2} = 0,50 \end{aligned}$$

olarak hesaplanır.

Hesaplanan 0,50 güven değeri, bu ilişki kuralının işlemler veritabanında süt ve ekmeği içeren işlemlerin %50'si için geçerli, doğru bir kural olduğunu ifade etmektedir. Bir diğer anlatımla, tüm alışverişler içerisinde süt ve ekmeğin birlikte alındığı alışverişlerin yarısında yumurtanın da alınmış olduğunu gösterir.

Güven Eşik Değeri

İlginc kural belirlemeye ikinci eleme işlemi, sık görülen nesne setleri üzerinden belirlenecek ilişki kuralları için hesaplanan güven değerleri ile yapılır. Bir ilişki kuralı için hesaplanan güven değeri aslında bir koşullu olasılıktır ve ilgili kuralın doğruluğunun bir ölçüsüdür. Bu nedenle belirlenecek güven eşik değerinin eleme gücü yüksek olmalı yani olabildiğince büyük seçilmelidir. İlişki kuralı için hesaplanan güven değeri, karar verici tarafından belirlenen güven eşik değerinden büyük olan ilişki kurallarının her biri yararlı bilgi üretmek amacıyla kullanılabilecek kurallar olurken, hesaplanan güven değeri belirlenen eşik değerinden daha küçük olan ilişki kuralları elenir, değerlendirilmeye alınmaz. Genellikle oluşturulan kuralın güçlü destek ve yüksek güven oranına sahip olması istenir. Bu özellikle kurallar, **güçlü kurallar** olarak nitelendirilir. İlişki analizinin temel amacı, bu şekilde tanımlanan güçlü kuralları tespit etmektir.

Genel olarak uygulamaya konulmak üzere karar vericinin seçimi ile ilginc kural olarak belirlenecek ilişki kuralı ilişki analizi sonucunda elde edilen güçlü kurallar arasından belirlenir. Ancak ilgilenilen problemin yapısına ve amacına bağlı olarak seçilecek ilginc ilişiki kuralının güçlü bir ilişki kuralı olması gerekmek. Bunun tersi durum olarak matematiksel olarak objektif ölçütler ile güçlü bir kural olarak belirlenen bir ilişki kuralının da ilginc kural olması gibi bir zorunluluk söz konusu değildir.

Belirlenen destek ve güven eşik değerleri üzerinde destek ve güven değerine sahip ilişki kuralına **güçlü kural** denir.

Tablo 5.1'de verilen beş işlemden oluşan market veritabanını kullanarak $\{Yumurta\} \Rightarrow \{Ekmek\}$ şeklinde oluşturulacak bir ilişki kuralının destek ve güven değerlerini bulunuz, elde ettiğiniz değerleri yorumlayınız.



SIRA SİZDE

Kaldırıcı (Lift)

İlişki kuralı oluşturmak için kullanılan algoritmalarının hepsi ilişki kuralı oluşturmada destek ve güven eşik değerlerini kullanır. Belirlenen destek ve güven eşik değerleri, güçlü olmayan birçok kuralın gereksiz yere elde edilmesini engellemesine rağmen, bazı durumlarda değerlendirilmesi gereken güçlü kural sayısı yine de fazla olabilmektedir. Böyle durumlarda ortaya çıkan güçlü kurallar içerisinde bir seçim yapabilmek ya da güçlü ku-

ralları önem sırasına göre sıralamak ve problemin amacına en uygun ilişki kuralını belirleyebilmek için ilave kısıtlamalar kullanmak gerekmektedir. Bunlar içerisinde en çok kullanılan ölçüt ise, öncül(A) ve sonuç(B) nesne setleri arasındaki ilişkinin(korelasyonun) belirlenmesi temeline dayanarak hesaplanan *kaldıraç(lift)* değeridir.

$A \Rightarrow B$ şeklinde ifade edilen bir ilişki kuralı için kaldıraç değeri, A ve B nesne setlerinin istatistiksel olarak bağımsız oldukları varsayıımı altında, kuralın güven değerinin sonucun (B'nin) destek değerine oranı şeklinde elde edilir ve

$$Kaldıraç(A \Rightarrow B) = \frac{Güven(A \Rightarrow B)}{Destek(B)} = \frac{Destek(A \cup B)}{Destek(A).Destek(B)} \quad (5.4)$$

eşitliği yardımıyla hesaplanır. Oluşturulan güclü ilişki kuralının ilginç yani bilgi üretmede kullanılabilir bir kural olup olmadığıının bir ölçüsü olarak hesaplanan kaldıraç değeri $[0, \infty)$ arasında değer alır ve yüzde olarak ifade edilir. Hesaplanan kaldıraç değerinin,

- Kaldıraç ($A \Rightarrow B < 1$) olması, A ve B nesne setleri arasında ters yönlü (negatif) bir ilişki olduğunu,
- Kaldıraç ($A \Rightarrow B = 1$) olması, A ve B nesne setleri arasında ilişki olmadığını
- Kaldıraç ($A \Rightarrow B > 1$) olması, A ve B nesne setleri arasında aynı yönlü (pozitif) bir ilişki olduğunu

$A \Rightarrow B$ şeklindeki bir ilişki kuralının **kaldıraç değeri**, aslında A ve B nesne setlerinin birlikte gözlenme olasılığının A'nın ve B'nin ayrı ayrı gözlenme olasılıklarının çarpımına oranıdır yani

$$\frac{P(A \cup B)}{P(A).P(B)}$$

ifade eder. **Kaldıraç değeri**, öncül (A) nesne setinin gözlendiği durumlarda sonuç(B) nesne setinin olasılığındaki değişim hakkında bilgi verir. İlişki kuralları için hesaplanacak kaldıraç değerinin özellikle 1 değerinden büyük olması istenilen durumdur. Çünkü bir ilişki kuralında kaldıraç değerinin 1'den büyük olması, tüm işlemler içerisinde sadece B'nin gözlendiği işlemlerin sayısının, A'nın gözlendiği işlemler içerisinde B'nin de gözlendiği işlem sayısından daha az olduğu anlamına gelir. Dolayısıyla kaldıraç değeri ne kadar büyük olursa, ilişki kuralını oluşturulan nesne setleri arasındaki ilişki de o kadar güçlü olur. Ancak burada şunu da belirtmek gerekir ki, kaldıraç değeri güçlü ilişki kuraları arasından bir seçim yapabilmek için kullanışlı bir ölçüt olmasına karşın her zaman en iyi sonuç elde edilemeyebilir. Şöyledi ki güçlü destek ve düşük kaldıraç değerine sahip bir ilişki kuralı, düşük destek ve yüksek kaldıraç değerlerine sahip bir ilişki kuralından daha kullanışlı olabilir.

ÖRNEK 3

Bir alışveriş veritabanındaki veriler aracılığıyla, $\{Havuç\} \Rightarrow \{Turp\}$ ve $\{Havuç\} \Rightarrow \{Süt\}$ şeklinde iki güclü ilişki kuralı belirlenmiş ve bu kurallar için,

1. Kural için; $Güven(\{Havuç\} \Rightarrow \{Turp\}) = 0,85$ ve $Destek\{Turp\} = 0,50$
2. Kural için; $Güven(\{Havuç\} \Rightarrow \{Süt\}) = 0,85$ ve $Destek\{Süt\} = 0,70$

değerleri hesaplanmıştır. Kaldıraç ölçütı aracıyla elde edilmiş olan bu iki güclü kural dan hangisinin ilginç kural olarak belirlenmesi gerektiğini bulunuz.

Bu iki güclü ilişki kuralından hangisinin ilginç kural olarak seçilmesi gerektigine karar verebilmek için öncelikle kaldıraç değerlerinin hesaplanması ve yorumlanması gerekir. Buna göre;

$$Kaldıraç(\{Havuç\} \Rightarrow \{Turp\}) = \frac{Güven(\{Havuç\} \Rightarrow \{Turp\})}{Destek\{Turp\}} = \frac{0,85}{0,50} = 1,70$$

Yorumu: Havuç alındığında turbun da alınma olasılığı, sadece turbun alınma olasılığından %70 daha fazladır.

$$Kaldıraç\left(\{Havuç\} \Rightarrow \{Süt\}\right) = \frac{Güven\left(\{Havuç\} \Rightarrow \{Süt\}\right)}{Destek\{Süt\}} = \frac{0,85}{0,70} = 1,21$$

Yorumu: Havuç alındığında sütün de alınma olasılığı, sadece sütün alınma olasılığının %21 daha fazladır.

Her iki ilişki kuralının da güven değerleri aynı olmasına rağmen bu kurallar için hesaplanan kaldıraç değerleri birbirinden farklıdır. Bu güçlü ilişki kuralları için hesaplanan her iki kaldıraç değerlerinin de 1'den büyük olması hem havuç ile turp nesneleri arasında hem de havuç ile süt nesneleri arasında pozitif (aynı yönlü) ilişki olduğunu ifade eder. Ayrıca kaldıraç değerlerinin büyülüklüklerine bakarak ise havuç nesnesinin turp nesnesinin gözlenme sıklığı üzerindeki etkisinin, havuç nesnesinin süt nesnesinin gözlenme sıklığı üzerindeki etkisinden daha fazla olduğu sonucuna varılır. Sonuç olarak daha büyük kaldıraç değerine sahip $\{Havuç\} \Rightarrow \{Turp\}$ şeklinde oluşturulan güçlü ilişki kuralının bilgi üretmek amacıyla ilginç kural olarak seçilmesinin daha doğru olacağı kararına varılır.

İLİŞKİ KURALI BELİRLEME AŞAMALARI

Genel olarak, bir ilişki kuralı oluşturmak iki temel adımdan oluşan bir süreçtir.

1. Adım: *Sık Görülen Nesne Setlerinin Elde Edilmesi:* Karar verici tarafından belirlenen destek eşik değerine eşit ya da daha yüksek destek değerine sahip nesne setleri yani sık görülen nesne setleri elde edilir.

2. Adım: *Sık Görülen Nesne Setleri ile Güçlü İlişki Kuralının Elde Edilmesi:* Birinci adımda belirlenen en yüksek mertebeye sahip yani en fazla nesne içeren sık görülen nesne setinin elemanları kullanılarak ilişki kuralları oluşturulur. k adet nesne içeren bir sık görülen nesne seti L_k şeklinde gösterilir. L_k 'nın elemanları kullanılarak oluşturulacak toplam ilişki kuralı sayısı $2^k - 2$ tanedir. Örneğin 4 nesne içeren bir sık görülen nesne seti L_4 'den toplamda $2^4 - 2 = 14$ tane ilişki kuralı oluşturulur. Oluşturulan ilişki kuralları içerisindeki belirlenen güven eşik değerine eşit ya da daha yüksek güven değerine sahip ilişki kuralları güçlü ilişki kuralları olarak nitelendirilir ve bilgi üretmek amacıyla kullanılabilir.

İlişki kuralı oluşturma aşamalarından ilki olan sık görülen nesne setlerinin belirlenmesi adımı, ikinci adıma göre işlem yükü açısından çok daha karmaşıktır. Dolayısıyla ilişki kuralı oluşturmak amacıyla kullanılan algoritmaların performansını belirleyen adım da birinci adımdır. Etkin bir şekilde bir ilişki kuralı oluşturabilmek için zaman içerisinde AIS, SETM, Apriori, Eclat ve FP-Growth gibi birçok algoritma geliştirilmiştir. Geliştirilen bu algoritmalar arasında en temel ve en çok kullanılan algoritma Apriori algoritmasıdır.

Apriori Algoritması

İlişki kuralı oluşturabilmek için geliştirilen algoritmalar içerisinde en çok bilinen ve en sık kullanılan algoritmadır. Apriori algoritması, 1994 yılında Agrawal ve Srikant tarafından geliştirilmiştir. Algoritmanın ismi, sık görülen nesne kümelerin önsel bilgisini kullanmasından, diğer bir ifadeyle bilgileri bir önceki adımdan almasından dolayı bir önceki (prior) anlamına gelen "apriori" dir.

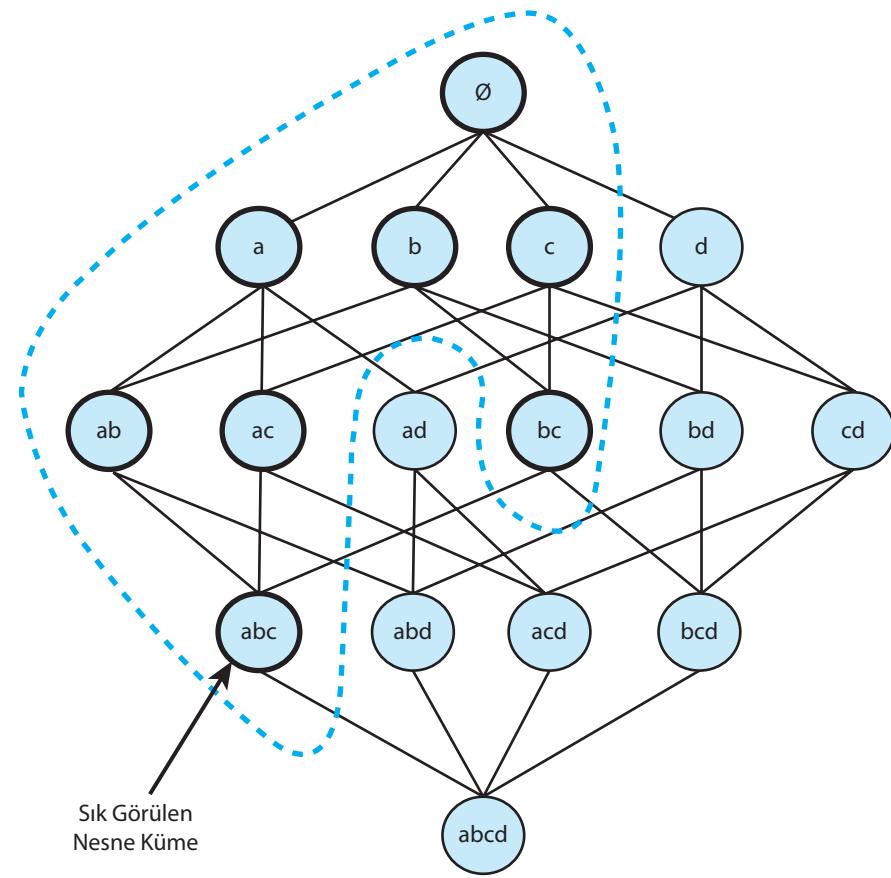
Apriori özelliği

Apriori algoritmasının temel yaklaşımı, "Eğer k nesneden oluşan nesne setleri kümesi en küçük destek kriterini sağlıyorsa, bu kümenin alt kümeleri de en küçük destek kriterini sağlar." şeklidindedir.

Örneğin; $I = \{a, b, c, d\}$ nesne kümesi için, şayet $\{a, b, c\}$ nesne kümesi bir sık görülen nesne kümesi ise, onun tüm alt kümeleri olan $\emptyset, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}$ ve $\{b, c\}$ kümeleri de sık görülen nesne kümeleridir. Bu özelliğe *apriori özelliği* adı verilir.

Şekil 5.2

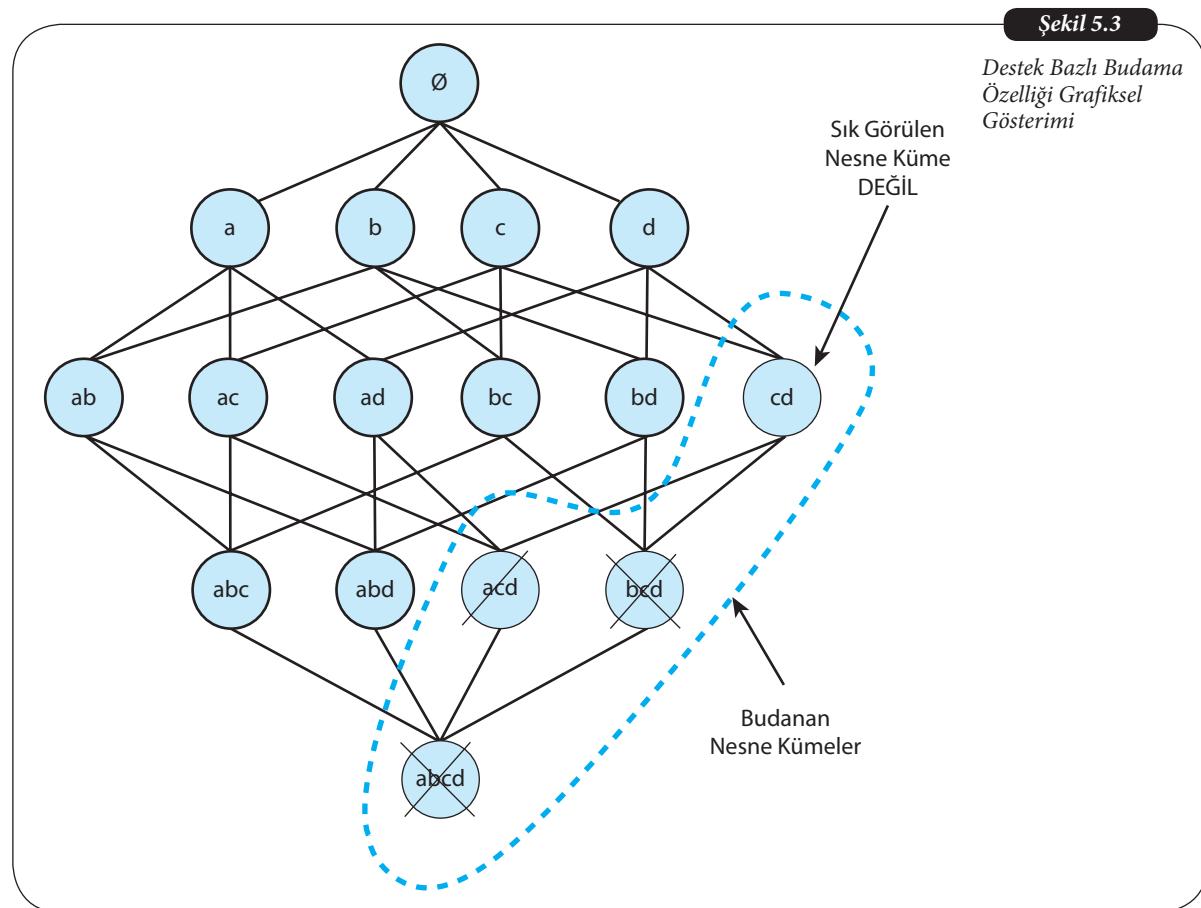
Apriori Özelliğinin
Grafiksel Gösterimi



Destek Bazlı Budama Özelliği

Apriori özelliğinin aksine, “Eğer bir alt küme sık görülen nesne kümesi değil ise, onun bütün üst kümeleri de sık görülen nesne kümesi değildir” temel yaklaşımına sahiptir. Böylece belirlenen destek eşik değerini geçemeyen az elemanlı kümelerin üst kümeleri de destek eşik değerini geçmeyeceği için değerlendirme dışı bırakılır. Bu yönteme *destek-bazlı budama (support based pruning)* denir.

Örneğin; $I=\{a,b,c,d\}$ nesne kümesi için, şayet $\{c, d\}$ nesne kümesi bir sık görülen nesne kümesi değil ise, bu kümeyi içeren tüm üst kümeleri olan $\{a, c, d\}$, $\{b, c, d\}$ ve $\{a, b, c, d\}$ kümeleri de sık görülen nesne kümeleri değildir.



$I=\{a,b,c,d\}$ şeklinde verilen dört nesne içeren nesne kümesi için, şayet $\{b\}$ ve $\{a, d\}$ nesne kümeleri sık görülen nesne kümeleri değil ise, destek bazlı budama özelliğine göre ilişki kuralı oluşturmak için kullanılabilen nesne seti sayısı kaçtır?



SIRA SİZDE

Apriori algoritması seviye mantığı (level-wise) arama olarak bilinen yinelemeli bir yaklaşım kullanır. Bu yaklaşımın k-1 öğeli nesne setleri ($k-1$) öğeli nesne setlerinin birleşimiyle oluşturulur. Böylece algoritma ile başlangıçta hesaplanan bilgiler daha sonraki yinelemelerde kullanıldığı için tekrar hesapların yapılması engellenmiş olur. Apriori algoritmasının işleyişinde ilk olarak bir nesne içeren nesne setleri arasından belirlenen destek eşik değerini geçen nesne setlerinden yani bir nesneli sık görülen nesne setleri kümesi belirlenir ve bu küme L_1 olarak adlandırılır. Daha sonra elde edilen L_1 kümesi, iki nesneli sık görülen nesne setleri kümesi olan L_2 'nin elde edilmesinde kullanılır. Benzer şekilde elde edilen L_2 kümesi ise L_3 kümesinin elde edilmesinde kullanılır. Bu şekilde daha fazla sık görülen nesne seti kümeleri bulunamayınca kadar yinelemeli bir şekilde algoritma ilerletilir.

Apriori algoritmasının, $k-1$ öğeli sık görülen nesne setleri kümesi L_{k-1} kullanılarak k öğeli sık görülen nesne setleri kümesi L_k 'nın elde edilmesi iki adımda gerçekleştirilir. Bunlar *birleştirme* ve *budama* adımlarıdır. Bu adımlar, Örnek 4'ün beşinci aşaması altında ayrıca incelenmiştir.

Apriori algoritmanın işleyışı ve uygulama adımları Örnek 4'ün çözümünde ayrıntılı olarak ele alınmıştır.

ÖRNEK 4

Bir marketten yapılan 4 adet alışverişe ait bilgiler Tablo 5.2'de verilmiştir. Market sahibi bu işlemler üzerinden bir ilişki kuralı oluşturmak istemektedir. Verilen bilgiler ile destek eşik değeri 0,50 ve giyen eşik değeri 0,75 olan güçlü ilişki kuralını adım adım elde ediniz.

Tablo 5.2
Dört Alışverişten
Oluşan İşlem Veritabanı
(D)

Tid	Nesneler
1	Makarna, Ayran, Et
2	Peynir, Ayran, Tavuk
3	Makarna, Peynir, Ayran, Tavuk
4	Peynir, Tavuk

Marketin D işlem veritabanına göre, marketten yapılan dört alışverişe alınan ürünlerin ilişkisel nesneler kümesi $I = \{Makarna, Peynir, Ayran, Et, Tavuk\}$ şeklinde oluşturulur.

Hatırlanacağı üzere bir ilişki kuralı iki adımda belirlenebiliyordu. Birinci adım sık görülen nesne setlerinin belirlenmesi, ikinci adım ise ilk adımada belirlenen sık görülen nesne setleri aracılığı ile güçlü ilişki kural(lar)ının elde edilmesiydi. Apriori algoritması ile ilişki kuralı oluşturma adımları çözüm üzerinde aşama aşama detaylandırılarak incelemiştir.

1. Adım: *Tüm Sık Görülen Nesne Setlerinin Elde Edilmesi:* Bu adımda amaç, apriori algoritması ile verilen 0,50 destek eşik değerine eşit veya daha büyük destek değerine sahip olan sık görülen nesne setleri kümelerinin elde edilmesidir.

- Aşama:** İlk aşamada I nesneler kümesindeki 1 adet nesne içeren nesne setleri belirleme ve belirlenen her bir nesne seti için destek değerleri hesaplanır.

Tablo 5.3
Oluşturulan 1 Ögeli
Nesne Setleri ve
Bunların Destek
Değerlerinin
Hesaplanması

1 Nesne İçeren Nesne Setleri	D Veritabanında Yer Aldığı İşlem Sayısı	Destek Değeri
Makarna	2	$2 / 4 = 0,50$
Peynir	3	$3 / 4 = 0,75$
Ayran	3	$3 / 4 = 0,75$
Et	1	$1 / 4 = 0,25$
Tavuk	3	$3 / 4 = 0,75$

- Aşama:** Hesaplanan destek değerleri içerisinde destek eşik değeri olarak verilen 0,50 değerinin üzerinde destek değerine sahip nesne setlerinden bir nesneli sık görülen nesne kümesi L_1 oluşturulur.

1 adet nesne içeren nesne setleri için hesaplanan destek değerleri, verilen destek eşik değeri 0,50 değeri ile karşılaştırıldığında sadece $\{Et\}$ nesne seti için hesaplanan 0,25 destek değerinin destek eşik değerinden küçük olduğu görülür. Dolayısıyla $\{Et\}$ nesne seti değerlendirme dışı bırakılır ve bir nesneli sık görülen nesne setleri kümesi L_1 ,

$$L_1 = \{\{Makarna\}, \{Peynir\}, \{Ayran\}, \{Tavuk\}\}$$

şeklinde elde edilir.

- Aşama:** L_1 sık görülen nesne setleri kümesi elemanlarının ikili kombinasyonları alınmak suretiyle birbirinden farklı tüm 2 adet nesne içeren nesne setleri oluşturulur. Ve oluşturulan bu nesne setlerinin destek değerleri hesaplanır.

2 Nesne İçeren Nesne Setleri	D Veritabanında Yer Aldığı İşlem Sayısı	Destek Değeri
Makarna, Peynir	1	$1 / 4 = 0,25$
Makarna, Ayran	2	$2 / 4 = 0,50$
Makarna, Tavuk	1	$1 / 4 = 0,25$
Peynir, Ayran	2	$2 / 4 = 0,50$
Peynir, Tavuk	3	$3 / 4 = 0,75$
Ayran, Tavuk	2	$2 / 4 = 0,50$

Tablo 5.4
 L_1 'in Elemanlarıyla Oluşturulan 2 Ögeli Nesne Setleri ve Bunların Destek Değerlerinin Hesaplanması

4. Aşama: 2 adet nesne içeren nesne setleri için hesaplanan destek değerleri içerisinde verilen destek eşik değeri 0,50 değerine eşit veya üzerinde destek değerine sahip nesne setlerinden iki nesneli sık görülen nesne setleri kümesi L_2 oluşturulur.

Tablo 5.4'ün son sütunundaki 2 adet nesne içeren nesne setleri için hesaplanan destek değerleri, verilen destek eşik değeri 0,50 değeri ile karşılaştırıldığında {Makarna, Peynir} ve {Makarna, Tavuk} nesne setleri değerlendirmeye dışı bırakılır. Buna göre iki nesneli sık görülen nesne setleri kümesi L_2 ,

$$L_2 = \{\{Makarna, Ayran\}, \{Peynir, Ayran\}, \{Peynir, Tavuk\}, \{Ayran, Tavuk\}\}$$

olarak elde edilir.

5. Aşama: Bu aşama giderek artan hesap yükünü azaltabilmek ve tekrar hesaplamlardan kaçınmak adına önceki aşamalarda elde edilen önsel bilgilerin değerlendirildiği aşamadır ve *birleştirme* ve *budama* adımlarından oluşur.

a. Birleştirme: L_2 sık görülen nesne setleri kümelerinin elemanlarının ikili kombinasyonları alınmak suretiyle 3 adet nesne içeren nesne setleri oluşturulur.

3 Adet Nesne İçeren Nesne Setleri
Makarna, Ayran, Peynir
Makarna, Ayran, Tavuk
Peynir, Ayran, Tavuk

Tablo 5.5
 L_2 'nin Elemanlarıyla Oluşturulması
Mümkün Tüm 3 Ögeli Nesne Setleri

b. Budama: *Apriori* özelliği gereğince oluşturulan ve 3 adet nesne içeren nesne setlerinin her biri ayrı ayrı ele alınır. Her bir 3 ögeli nesne setinin 2 ögeli alt kümelerinin L_2 'nin elemanı olup olmadıkları kontrol edilir. Şayet herhangi bir 3 nesneli nesne setinin 2 nesneli alt kümelerinin tümü L_2 'nin elemanı ise bu nesne seti için destek değeri hesaplanır. Aksi takdirde hesaplama yapmaya gerek yoktur.

Tablo 5.5'teki 3 adet nesne içeren nesne setlerinin ilki olan {Makarna, Ayran, Peynir} nesne setinin 2 nesneli alt kümeleri {Makarna, Ayran}, {Makarna, Peynir} ve {Ayran, Peynir}'dir. Bu alt kümelerden {Makarna, Peynir} ve {Ayran, Peynir} alt kümeleri L_2 'nin elemanı olmadığı için {Makarna, Ayran, Peynir} 3 ögeli nesne seti için destek değeri hesaplamaya gerek yoktur ve bu nesne seti elenerek değerlendirmeye dışı bırakılır. Benzer durum {Makarna, Ayran, Tavuk} nesne seti için de geçerlidir ve bu nesne seti için de destek değeri hesaplamaya gerek yoktur ve bu nesne seti de elenerek değerlendirmeye alınmaz. Sadece {Peynir, Ayran, Tavuk} nesne setinin tüm 2 nesneli alt kümeleri –ki bunlar {Peynir, Ayran}, {Peynir, Tavuk} ve {Ayran, Tavuk} kümeleridir – L_2 'nin elemanı oldukları için bu nesne seti için destek değeri hesaplanır.

Tablo 5.6
Budama Sonrası Elde Edilen 3 Ögeli Nesne Setleri ve Bunların Destek Değerlerinin Hesaplanması

3 Nesne İçeren Nesne Setleri	D Veritabanında Yer Aldığı İşlem Sayısı	Destek Değeri
Peynir, Ayran, Tavuk	2	2 / 4 = 0,50

6. Aşama: Budama sonrasında kalan 3 adet nesne içeren nesne setleri için hesaplanan destek değerleri verilen destek eşik değeri 0,50 değeri ile karşılaştırılır. Destek eşik değerine eşit veya üzerinde destek değerine sahip nesne setlerinden üç nesneli sık görülen nesne kümesi L_3 oluşturulur.

Tablo 5.6'da verilen ve budama sonucunda elde edilen 3 adet nesne içeren nesne seti bir tane olduğundan ve hesaplanan destek değeri destek eşik değeri olan 0,50 değerine eşit olduğu için 3 nesneli sık görülen nesne kümesi L_3 ,

$$L3 = \{\{Peynir, Ayran, Tavuk\}\}$$

şeklinde elde edilir.

7. Aşama: L_3 sık görülen nesne kümesinin sadece bir elemanı olduğu için 4 ve daha fazla nesneden oluşan nesne setleri oluşturulamaz.

Böylece algoritmanın ilk adımı tamamlanmış olur ve ikinci adıma geçilir.

2. Adım: *Sık Görülen Nesne Setlerinden Güçlü İlişki Kuralının Elde Edilmesi:* Bu adımda amaç, birinci adımda elde edilen en yüksek mertebe sahip sık görülen nesne setleri kümesinin elemanları kullanılarak güçlü ilişki kural(lar)ının oluşturulmasıdır.

Birinci adım sonunda elde edilen ve en çok nesne içeren sık görülen nesne kümesi,

$$L3 = \{\{Peynir, Ayran, Tavuk\}\}$$

şeklindedir ve ilişki kuralları bu küme elemanları ile oluşturulur.

DİKKAT



Her bir k ögeli sık görülen nesne setinden 2^{k-2} adet ilişki kuralı oluşturulabilir. Oluşturulan ilişki kuralları için güven değerleri hesaplanır. Belirlenen güven eşik değerine eşit veya üzerinde güven değerine sahip kurallar *güçlü kurallar* olarak nitelendirilir ve uygulamaya konulabilir.

Örnek 4'ün çözümünde birinci adım sonunda elde edilen sık görülen nesne setleri kümesi L_3 'ün bir tek elemanı olduğundan ve bu da 3 adet nesne içerdiginden yani $k=3$ olduğundan bu küme elemanları ile $2^{k-2}=2^3-2=6$ adet ilişki kuralı oluşturulabilir. Buna göre oluşturulacak ilk ilişki kuralı ve bunun güven değerinin hesaplanması şu şekildedir.

1. İlişki Kuralı: $\{Peynir\} \Rightarrow \{Ayran, Tavuk\}$

$$\begin{aligned} \text{Güven}\left(\{Peynir\} \Rightarrow \{Ayran, Tavuk\}\right) &= \frac{\text{Destek}\{\{Peynir, Ayran, Tavuk\}\}}{\text{Destek}\{\{Peynir\}\}} \\ &= \frac{2 / 4}{3 / 4} = \frac{0,50}{0,75} = 0,67 \end{aligned}$$

$\{Peynir\} \Rightarrow \{Ayran, Tavuk\}$ şeklinde oluşturulan ilk kural için hesaplanan 0,67 güven değeri D işlemler veritabanında peyniri içeren işlemlerin %67'sinin ayran ve tavuğu da içerdigini ifade eder. Ancak bu kuralın güven değeri verilen güven eşik değeri 0,75 değerlinden daha küçük bir değer olduğu için bu kural *güçlü bir kural değildir*.

Benzer şekilde oluşturulacak tüm ilişki kuralları ve bunların güven değerleri Tablo 5.7'de verilmiştir.

Kural No	İlişki Kuralı	Güven Değeri
1	{Peynir} \Rightarrow {Ayran, Tavuk}	(2/4) / (3/4) = 0,67
2	{Ayran} \Rightarrow {Peynir, Tavuk}	(2/4) / (3/4) = 0,67
3	{Tavuk} \Rightarrow {Peynir, Ayran}	(2/4) / (3/4) = 0,67
4	{Peynir, Ayran} \Rightarrow {Tavuk}	(2/4) / (2/4) = 1,00
5	{Peynir, Tavuk} \Rightarrow {Ayran}	(2/4) / (3/4) = 0,67
6	{Ayran, Tavuk} \Rightarrow {Peynir}	(2/4) / (2/4) = 1,00

Tablo 5.7
Oluşturulan İlişki Kuralları ve Bunların Güven Değerlerinin Hesaplanması

Elde edilen en yüksek mertebeden sık görülen nesne setleri kümesi L_3 'ün elemanlarından oluşturulabilecek tüm ilişki kuralları ve bunların hesaplanan güven değerlerine ilişkin düzenlenen Tablo 5.7 incelendiğinde yalnızca 4. ve 6. sıralarda yer alan sırasıyla {Peynir, Ayran} \Rightarrow {Tavuk} ve {Ayran} \Rightarrow {Tavuk, Peynir} şeklinde belirlenen ilişki kurallarının güven eşik değeri olan 0,75'in üzerinde güven değerlerine sahip oldukları görülür. Dolayısıyla bu iki kural *güçlü kurallardır* ve bilgi üretmek için kullanılabilir.

Sonuç olarak, bir marketten yapılan 4 adet alışverişten meydana gelen veritabanı üzerinden destek değeri en az 0,50 ve güven değeri en az 0,75 olan güçlü ilişki kural(lar) oluşturmak istendiğinde oluşturulan güçlü ilişki kuralları, hesaplanan destek ve güven değerleri ile bu değerlerin yorumları izleyen sekildedir.

İlişki Kuralı I: {Peynir, Ayran} \Rightarrow {Tavuk}

$$\text{Destek} \left\{ \text{Peynir, Ayran, Tavuk} \right\} = \frac{2}{4} = 0,50 (\%50)$$

Yorumu: Yapılan tüm alışverişlerin %50'sinde peynir, ayran ve tavuk birlikte alınmıştır.

$$\text{Güven} \left(\left\{ \text{Peynir, Ayran} \right\} \Rightarrow \left\{ \text{Tavuk} \right\} \right) = \frac{2/4}{2/4} = 1,00 (\%100)$$

Yorumu: Yapılan tüm alışverişler içerisinde peynir ve ayran alınan alışverişlerin %100'ünde yani tamamında bunların yanında tavuk da alınmıştır.

İlişki Kuralı II: {Ayran, Tavuk} \Rightarrow {Peynir}

$$\text{Destek} \left\{ \text{Ayran, Tavuk, Peynir} \right\} = \frac{2}{4} = 0,50 (\%50)$$

Yorumu: Yapılan tüm alışverişlerin %50'sinde ayran, tavuk ve peynir birlikte alınmıştır.

$$\text{Güven} \left(\left\{ \text{Ayran, Tavuk} \right\} \Rightarrow \left\{ \text{Peynir} \right\} \right) = \frac{2/4}{2/4} = 1,00 (\%100)$$

Yorumu: Yapılan tüm alışverişler içerisinde ayran ve tavuk alınan alışverişlerin %100'ünde yani tamamında bunların yanında peynir de alınmıştır.

Örnek 4'te verilen problemin çözümünde iki güçlü ilişki kuralı elde edilmiş ve elde edilen her iki kuralın da hesaplanan güven değerlerinin aynı olduğu görülmüştür. Dolayısıyla her iki ilişki kuralı da ilginç kural olmaya aday kurallardır ve uygulamaya konulmak üzere karar vericinin seçimine sunulabilir. Ancak belirlenen bu iki kuraldan hangisinin objektif bir ölçüt ile seçilmesinin daha doğru olacağına karar verebilmek adına her iki kuralın kaldırıcı değerlerinin hesaplanarak yorumlanması gereklidir.

İlişki Kuralı I: {Peynir, Ayran} \Rightarrow {Tavuk} için,

$$\begin{aligned} \text{Kaldıraç } \left(\left\{ \text{Peynir, Ayran} \right\} \Rightarrow \left\{ \text{Tavuk} \right\} \right) &= \frac{\text{Güven} \left(\left\{ \text{Peynir, Ayran} \right\} \Rightarrow \left\{ \text{Tavuk} \right\} \right)}{\text{Destek} \left\{ \text{Tavuk} \right\}} \\ &= \frac{1,00}{0,75} = 1,33 \end{aligned}$$

Yorumu: Peynir ve ayran alındığında Tavuğuñ da alınma olasılığı, sadece tavuğuñ alınma olasılığından %33 daha fazladır.

İlişki Kuralı II: {Ayran, Tavuk} \Rightarrow {Peynir} için,

$$\begin{aligned} \text{Kaldıraç } \left(\left\{ \text{Ayran, Tavuk} \right\} \Rightarrow \left\{ \text{Peynir} \right\} \right) &= \frac{\text{Güven} \left(\left\{ \text{Ayran, Tavuk} \right\} \Rightarrow \left\{ \text{Peynir} \right\} \right)}{\text{Destek} \left\{ \text{Peynir} \right\}} \\ &= \frac{1,00}{0,75} = 1,33 \end{aligned}$$

Yorumu: Ayran ve tavuk alındığında peynirin de alınma olasılığı, sadece peynirin alınma olasılığından %33 daha fazladır.

Her iki güçlü ilişki kuralının da hesaplanan kaldıraç değerleri 1 değerinden büyük olduğu için her ikisi için de öncül ve sonuç nesne setleri arasında pozitif bir ilişki olduğu söylenebilir. Ayrıca yine her iki güçlü ilişki kuralı için hesaplanan kaldıraç değerleri birbirine eşit olduğu için de her ikisinin de ilginç kurallar olarak uygulamaya konulabileceğini söylemek mümkündür. Elbette hangi kuralın uygulamaya konulacağı yönündeki son karar, karar vericiye ait olacaktır.

İLİŞKİ KURALLARI R ÇÖZÜMÜ

R ile ilişki kuralı oluşturabilmek için *arules* paketinin R'de kurulması ve hafızaya yüklenmesi gereklidir. *arules* paketi içerisinde yer alan *apriori()* fonksiyonu yardımıyla güçlü ilişki kuralları oluşturulur.

İNTERNET



<https://cran.r-project.org/web/packages/arules/>

apriori() fonksiyonunun temel parametreleri ilişki kurallarının oluşturulabilmesi için elde edilen tüm işlemleri (alışverişleri) barındıran veri değişkenini ifade eden *data* ve özellikle destek ve güven eşik değerleri vb. kısıtlamalara ilişkin eşik değerlerinin belirlendiği *parameter*'dır. Veri girişi standart veri girişlerinden herhangi birisi ile yapılabilir. Ancak girilen verinin *apriori()* fonksiyonu ile işlenebilmesi için işlemlerden oluşan veritabanı formatına dönüştürülmesi gereklidir. Veri dönüşümü için **help("transactions")** komutundan ve fonksiyon ile ilgili yardım için ise, **help("apriori")** komutundan yararlanılabilir.

Örnek 4 için *apriori()* fonksiyonu yardımıyla dört işlemden oluşan veritabanından destek eşik değeri 0,50 ve güven eşik değeri 0,75 olan güçlü ilişki kurallarının elde edilmesine ilişkin komut dizisi ve hesaplama sonucu izleyen biçimde ortaya çıkacaktır.

```

> library("arules")
> veri<-list (c ("Makarna", "Ayran", "Et"),
  c("Peynir","Ayran","Tavuk"), c("Makarna","Peynir","Ayran
  ","Tavuk"), c("Peynir","Tavuk"))
> islem <- as(veri, "transactions")
> kurallar <- apriori(islem, parameter = list(supp=0.50,
  conf=0.75, minlen=3))
> inspect(kurallar)
lhs          rhs      support   confidence      lift
1 {Ayran, Peynir} => {Tavuk}    0.5           1       1.333333
2 {Ayran, Tavuk} => {Peynir}   0.5           1       1.333333
  
```

Verilen komut dizisinin dördüncü satırındaki “`islem <- as(veri, "transactions")`” komutu, liste şeklinde girilmiş olan işlem verilerinin ***apriori()*** fonksiyonu ile işlenebilmesi için gereken veri dönüşümünün yapıldığı atama komutudur. Komut dizisinin en altında elde edilen “kurallar” değişkeni dört adet işlem içeren veritabanı üzerinden oluşturulan, destek değeri en az 0,50 ve güven değeri en az 0,75 olan güçlü ilişki kurallarını ve bu kuralların hesaplanan sırasıyla destek, güven ve kaldırıcı değerlerini vermektedir. R aracılığı ile elde edilen güçlü ilişki kuralları ve bu kuralların hesaplanan destek, güven ve kaldırıcı değerlerinin Örnek 4’ün çözümünde elde edilen sonuçlar ile aynı olduğu görülmektedir.

Yapılan analiz sonucunda {Çekiç,Çivî} => {Pense} şeklinde elde edilen güçlü ilişki kuralı için destek değeri 0,85 ve güven değeri 0,90 olarak elde edilmiştir. İlişki kuralını yorumlayınız.



SIRA SİZDE

4

Özet



İlişki kurallarına ilişkin temel kavramları tanımlamak

Veri madenciliğinde temel yöntemlerden birisi olan ilişki kuralları, toplanıp depolanarak çok büyük boyutlara ulaşabilen eş zamanlı verilerin analiz edilerek nesne ve/veya nesne setleri arasındaki ilginç birlikteklilik örüntülerinin keşfedilmesidir. Bu sayede birçok uygulama alanında daha verimli kararlar verilebilmek mümkün olur. Elbette ki eldeki veri yiğini içerisinde bir çok ilişki kuralı oluşturmak mümkündür. A ve B olarak tanımlanan iki nesne seti için genel olarak oluşturulacak bir ilişki kuralı $A \Rightarrow B$ şeklinde ifade edilir ki burada A nesne setine *öncül*, B nesne setine ise *sonuç* adı verilir. Bulunan ilişki kurallarının da objektif ölçütler aracılığı ile elenerek sayılarının en aza indirilmesi gereklidir. Bu amaçla kullanılan ölçütler içerisinde en yaygın kabul görenleri destek, güven ve kaldırıcı ölçütleridir.



Veri kümesi için güçlü ilişki kuralları elde etmek

Temelde eş zamanlı olarak elde edilen veri kümesinden oluşturulacak çok sayıda ilişki kuralı arasında seçilecek bir kuralın *güçlü* olarak nitelendirilebilmesi için ilişki kuralının amaç doğrultusunda karar verebilmeyi sağlayacak bilgiyi sunması gerekmektedir. Bu da ancak objektif ölçütler kullanılarak yapılabilir. Bir ilişki kuralının *güçlü kural* olarak kabul edilebilmesi için ilk etapta belirlenecek destek ve güven eşik değerlerini sağlaması gereklidir. Bilişim sektöründeki gelişmelere paralel olarak, günümüze kadar güçlü ilişki kuralları elde edebilmek için birçok algoritma geliştirilmiştir.



Apriori algoritmasının çalışma prensibini açıklamak

Apriori algoritması, bir ilişki kuralı elde etmek için veri setinin analiz ve çözümlenmesi esnasında kullanılan bir çözüm algoritmasıdır. 1994 yılında Agrawal ve Srikant tarafından geliştirilen algoritmanın temel amacı, iki temel adımdan oluşan ilişki kuralı elde etme adımlarından ilki olan ve işlem yükü açısından ikinci adıma göre daha fazla işlem gerektiren sık görülen nesne setlerinin mümkün olan en az işlem gerçekleştirerek belirlenmesidir. Bunun için seviye mantığı arama olarak bilinen yinelemeli bir yaklaşım kullanılır. Bu sayede algoritmanın ilk çalışmasında elde edilen bilgiler daha sonraki yinelemelerde de kullanılarak her bir yinelemede tekrar tekrar hesaplama yapılmasılarından kaçınılarak daha kısa zamanda ve daha az işlem yükü ile sık görülen nesne setleri belirlenebilir.



İlişki kurallarını yorumlamak

Ekonomiden eğitime, sağlıktan pazarlamaya geniş bir uygulama alanına sahip ilişki kuralları, çok büyük miktarlarda verinin bulunduğu veritabanı içerisinde eşanlı elde edilen nesne ve/veya nesne kümeleleri arasında ilk bakışta farkedilmeyen, karar verme, pazarlama ve iş yönetimi açısından önemli faydalara sağlayacak ilişkilerin keşfedilmesidir. $A \Rightarrow B$ şeklinde elde edilen bir ilişki kuralının bilgi üretmek amacıyla kullanılmıştır. Elde edilen ilişki kuralı, kural için hesaplanan destek ve güven değerleri üzerinden yorumlanır. Her iki değer açısından da en büyük olasılığa sahip kural karar almada bilgi üretmek amacıyla kullanılabilir. Şayet birden fazla güçlü ilişki kuralı elde edilmiş ise bunlar arasında seçim yapabilmek için kaldırıcı değerlerinin hesaplanması gereklidir. Hesaplanan kaldırıcı değeri, A öncül ve B sonuç nesne setleri arasındaki ilişkinin yönü ve kuvveti hakkında bir bilgi sağlar.

Kendimizi Sınayalım

- 1.** $A \Rightarrow B$ şeklinde ifade edilen bir ilişki kuralında A nesne seti ne olarak ifade edilir?
- İşlem
 - Öncül
 - Veritabanı
 - Kayıt
 - Sonuç
- 2.** 3 adet nesne içeren nesneler kümesinden ilişki kuralı oluşturmada kullanılabilecek farklı nesne sayılarına sahip oluşturulabilecek nesne seti sayısı kaçtır?
- 2
 - 4
 - 6
 - 7
 - 8
- 3.** Belirlenen destek eşik değerini geçen ve kural oluşturmadan kullanılacak nesne setlerine ne ad verilir?
- Sık görülen nesne seti
 - İlginç ilişki kuralı
 - Nesne seti
 - Güçlü ilişki kuralı
 - Nesneler kümesi
- 4.** Destek($\{Kahve, Şeker\} \Rightarrow \{Süt\}$)=0,75 olarak hesaplanmış ise aşağıdakilerden hangisi doğrudur?
- Kahve ve şeker alanların %75'i süt de almışlardır.
 - Kahve ve şekerin birlikte alındığı alışverişlerin %75'inde süt de alınmıştır.
 - Kahve ve şeker alanlar beraberinde süt de almışlardır.
 - Kahve ve şeker alma olasılığı, sadece süt alma olasılığından %75 fazladır.
 - Kahve ve şeker alanlar %25 olasılıkla süt de alırlar.
- 5.** Bir D alışveriş veritabanından $X \Rightarrow Y$ şeklinde oluşturulan bir ilişki kuralı için $|X|=20$, $|Y|=24$, $|X \cup Y|=12$ ve $|D|=50$ olarak elde edilmiştir. Buna göre bu ilişki kuralı için hesaplanacak güven değeri kaçtır?
- 0,24
 - 0,40
 - 0,50
 - 0,60
 - 0,83
- 6.** $I=\{a,b,c,d,e\}$ nesne kümesi için $\{a,c,d,e\}$ nesne kümesi sık görülen nesne küme olarak belirlenmiş ise aşağıdakilerden hangisi sık görülen nesne küme **olamaz**?
- {a,e}
 - {c,d}
 - {a,c,e}
 - {c,d,e}
 - {a,b,c,d,e}
- 7.** Herhangi bir kümenin sık görülen nesne küme olmaması durumunda onun bütün üst kümelerinin de sık görülen nesne küme olmaması ile açıklanabilen yöntem aşağıdakilerden hangisidir?
- Pazar sepeti analizi
 - Birleştirme adımı
 - Apriori algoritması
 - Yinelemeli yaklaşım
 - Destek bazlı budama
- 8.** $A \Rightarrow B$ şeklinde oluşturulan bir ilişki kuralının hesaplanan kaldırıcı değerinin 1 olması durumunda aşağıdaki ifadelerden hangisi doğrudur?
- Elde edilen kural güclü bir kural değildir.
 - A ve B nesne setlerinin tüm işlemler içerisinde gözlenme sayıları aynıdır.
 - A ve B nesne setleri arasında ilişki yoktur.
 - A nesne setini içeren işlemler aynı zamanda B nesne setini de içermektedir.
 - Elde edilen kuralın destek ve güven değerleri birbirine eşittir.
- 9.** 10 adet işlem içeren veritabanından $X \Rightarrow Y$ şeklinde oluşturulan bir ilişki kuralı için, Destek(X)=0,60, Destek(Y)=0,50 ve Destek($X \Rightarrow Y$)=0,30 olarak elde edilmiştir. Buna göre bu ilişki kuralı için aşağıdaki ifadelerden hangisi **yanlıştır**?
- $|X|=6$
 - $|Y|=4$
 - $|X \cup Y|=3$
 - $\text{Güven}(X \Rightarrow Y)=0,50$
 - $\text{Kaldırıcı}(X \Rightarrow Y)=1,00$
- 10.** R yazılımı aracılığı ile güclü ilişki kuralı elde edebilmek için hangi fonksiyon kullanılır?
- arules
 - transactions
 - apriori
 - data
 - parameter

Kendimizi Sınavalım Yanıt Anahtarları

1. b Yanınız yanlış ise “Pazar Sepeti Analizi” konusunu yeniden gözden geçiriniz.
2. d Yanınız yanlış ise “Pazar Sepeti Analizi” konusunu yeniden gözden geçiriniz.
3. a Yanınız yanlış ise “Destek (Support)” konusunu yeniden gözden geçiriniz.
4. a Yanınız yanlış ise “Destek (Support)” konusunu yeniden gözden geçiriniz.
5. d Yanınız yanlış ise “Güven (Confidence)” konusunu yeniden gözden geçiriniz.
6. e Yanınız yanlış ise “Apriori Algoritması” konusunu yeniden gözden geçiriniz.
7. e Yanınız yanlış ise “Apriori Algoritması” konusunu yeniden gözden geçiriniz.
8. c Yanınız yanlış ise “Kaldırıcı (Lift)” konusunu yeniden gözden geçiriniz.
9. b Yanınız yanlış ise “İlginç Kural Belirleme Ölçütleri” konusunu yeniden gözden geçiriniz.
10. c Yanınız yanlış ise “R Çözümü: İlişki Kuralları” konusunu yeniden gözden geçiriniz.

Sıra Sizde Yanıt Anahtarları

Sıra Sizde 1

Bir nesneden oluşan ($k=1$) nesne setleri ve destek değerleri,

$$\text{Destek} \left\{ \text{Süt} \right\} = \frac{2}{5} = 0,40 \quad \text{Destek} \left\{ \text{Ekmek} \right\} = \frac{5}{5} = 1,00$$

$$\text{Destek} \left\{ \text{Yumurta} \right\} = \frac{3}{5} = 0,60 \quad \text{Destek} \left\{ \text{Şeker} \right\} = \frac{1}{5} = 0,20$$

Üç nesneden oluşan ($k=3$) nesne setleri ve destek değerleri,

$$\text{Destek} \left\{ \text{Süt, Ekmek, Yumurta} \right\} = \frac{1}{5} = 0,20$$

$$\text{Destek} \left\{ \text{Süt, Ekmek, Şeker} \right\} = \frac{0}{5} = 0,00$$

$$\text{Destek} \left\{ \text{Ekmek, Yumurta, Şeker} \right\} = \frac{0}{5} = 0,00$$

Dört nesneden oluşan ($k=4$) nesne seti ve destek değeri,

$$\text{Destek} \left\{ \text{Süt, Ekmek, Yumurta, Şeker} \right\} = \frac{0}{5} = 0,00$$

olarak elde edilir.

Sıra Sizde 2

Tablo 5.1'deki market veritabanı üzerinden $\{\text{Yumurta}\} \Rightarrow \{\text{Ekmek}\}$ şeklinde belirlenen ilişki kuralının destek değeri,

$$\begin{aligned} \text{Destek} \left(\left\{ \text{Yumurta} \right\} \Rightarrow \left\{ \text{Ekmek} \right\} \right) &= \frac{|\{\text{Yumurta, Ekmek}\}|}{|D|} \\ &= \frac{3}{5} = 0,60 \end{aligned}$$

olarak elde edilir. Buna göre yapılan bu marketten yapılan alışverişlerin %60'ında yumurta ve ekmeğin birlikte alınmış olduğu söylenir.

Aynı kuralın güven değeri ise,

$$\begin{aligned} \text{Güven} \left(\left\{ \text{Yumurta} \right\} \Rightarrow \left\{ \text{Ekmek} \right\} \right) &= \frac{|\{\text{Yumurta, Ekmek}\}|}{|\{\text{Yumurta}\}|} \\ &= \frac{3}{3} = 1,00 \end{aligned}$$

olarak hesaplanır. Dolayısıyla elde edilen bu değere göre oluşturulan ilişki kuralının market veritabanında yumurtayı içeren alışverişlerin %100'ü yani tamamı için geçerli, doğru bir kural olduğunu söyleyor.

Sıra Sizde 3

$m=4$ adet nesne içeren bir $I=\{a,b,c,d\}$ nesneler kümesinden ilişki kuralı oluşturmada kullanılacak nese seti sayısı $2^{m-1}=2^4-1=15$ tanedir.

1 ögeli nesne setleri: $\{a\}, \{b\}, \{c\}, \{d\}$

2 ögeli nesne setleri: $\{a, b\}, \{a, c\}, \{a, d\}, \{b, c\}, \{b, d\}, \{c, d\}$

3 ögeli nesne setleri: $\{a, b, c\}, \{a, b, d\}, \{a, c, d\}, \{b, c, d\}$

4 ögeli nesne setleri: $\{a, b, c, d\}$

Tüm bu 15 nesne seti içerisinde $\{b\}$ ve $\{a, d\}$ nesne setlerinin sık görülen nesne setleri olmadığı bilindiğine göre destek bazlı budama özelliğine göre bu 15 nesne seti içerisinde $\{b\}$ nesnesini içeren nesne setleri ve $\{a, d\}$ nesnelerini içeren nesne setleri budanır ve değerlendirme dışı bırakılır. Dolayısıyla geriye ilişki kuralı oluşturmak için kullanabilecek $\{a\}, \{c\}, \{d\}, \{a, c\}$ ve $\{c, d\}$ olmak üzere 5 adet nesne seti kalır.

Sıra Sizde 4

{Çekici, Çivi} \Rightarrow {Pense} olarak elde edilen ilişki kuralının güçlü bir kural olması, analizi yapan (veya karar verici) tarafindan belirlenen destek ve güven eşik değerlerini sağladığını ifade eder.

Analiz sonucunda elde edilen destek değeri, kullanılan veritabanındaki tüm alışverişler içerisinde çekici, civi ve pense nesnelerinin birlikte alınma oranının %85 olduğunu, güven değeri ise çekici ve civinin birlikte alındığı alışverişler içerisinde pensenin de alınma oranının %90 olduğunu göstermektedir.

Yararlanılan ve Başvurulabilecek Kaynaklar

- Aggarwal, C.C. (2015). **Data Mining: The Textbook**, New York.
- Berry, M., Linoff, G. (2004). **Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management**, John Wiley & Sons.
- Han, J., Kamber, M., Pei, J. (2012). **Data Mining: Concepts and Techniques**, 3rd Edition, Morgan Kaufmann Publications, USA.
- Makhabel, B. (2015). **Learning Data Mining with R**, Packt Publishing Ltd., UK.
- Michael Hahsler, Christian Buchta, Bettina Gruen and Kurt Hornik (2015). **arules: Mining Association Rules and Frequent Itemsets**. R package version 1.2-1. <http://CRAN.R-project.org/package=arules>
- Michael Hahsler, Bettina Gruen and Kurt Hornik (2005), **arules - A Computational Environment for Mining Association Rules and Frequent Item Sets**. Journal of Statistical Software 14/15. URL: <http://www.jstatsoft.org/v14/i15/>
- Tan, P.N., Steinbach, M., & Kumar, V. (2005). **Introduction to Data Mining**. Addison-Wesley, Boston, USA.

6

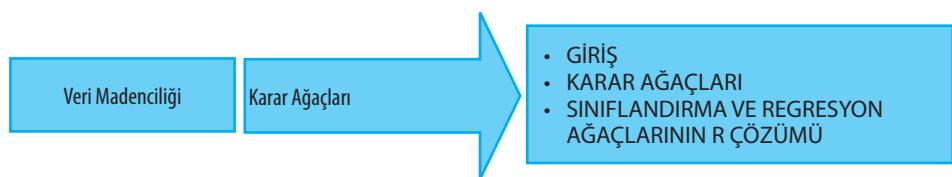
Amaçlarımız

- Bu üniteyi tamamladıktan sonra;
- 🕒 Karar verme, sınıflandırma ve kestirim kavramlarını açıklayabilecek,
 - 🕒 Karar ağaçlarını tanımlayabilecek,
 - 🕒 Sınıflandırma ve regresyon ağaçlarını R yazılımını kullanarak oluşturabilecek ve yorumlayabilecek bilgi ve becerilerine sahip olabileceksiniz.

Anahtar Kavramlar

- Karar Verme
- Karar Ağacı
- Ayırma Kriteri
- Entropi İndeksi
- Kazanç Ölçütü
- Gini İndeksi
- Sınıflandırma ve Regresyon Ağaçları

İçindekiler



Karar Ağaçları

GİRİŞ

Karar verme, karşılaşılan birden fazla seçenek içerisinde seçim yapma işlemidir. Karar verme bireyler kadar, işletmeler için de söz konusudur. Karar verme içgüdüsel olabileceği gibi, ihtiyaç duyulan gereksinimlerin çeşitli kriterlerine de bağlı olabilmektedir. Günüümüzün hızlı yaşam şartları, karşılaşılan seçeneklerin sayısını artırdığı gibi karar verme işleminin de hızlı bir şekilde yerine getirilmesini zorunlu hâle getirmektedir. Ancak, seçim sayısının çok olmasının ve kararların hızlı alınmasının, karar vericinin karar verme yeteneğini olumsuz olarak etkilememesi beklenir. Karar verme teknikleri en uygun kararın verilmesinde yardımcı olmak amacıyla geliştirilmiş tekniklerdir. Özellikle işletmelerin başarısının yöneticilerinin tutarlı kararlar verebilmesine bağlı olduğu düşünülebilir.

Karar probleminin zaman içerisinde doğuracağı sonuçlardan etkilenen sorumlu kişiye **karar verici** adı verilir. Karar verici, bir kişi olabileceği gibi bir grup veya bir kurum da olabilir. Karar verici için amaç, karar sürecinde önceden saptanan ve karar verici için belirgin özelliği olumlu olan sonuca ulaşmaktır. Belirlenen amaca ulaşmada etkin olan en az iki eylem biçimini söz konusu iken, bu eylem biçimlerinin seçiminde karar vericinin içinde yer alabileceği çeşitli koşullar veya şartlar etkili olabilmektedir. Bunun yanında, karar vericinin, çözümünü araştırdığı karar problemine ilişkin kapsam, çevre ve paydaşları çok iyi bir şekilde değerlendirmesi gerekmektedir.

Çeşitli eylem seçenekleri arasından uygun olanını belirleyen kararın etkin olması ve bu kararı uygulayacak olanların arasında mümkün olduğu kadar yüksek düzeyde kabul görmesi beklenir. Karar, kendi yargı birimlerine göre iyi olarak nitelendirilmelidir. İyi bir karar, benzer problemler ile karşı karşıya kalan iki farklı yöneticinin aynı seçenekler ve aynı koşullar altında aynı kararı vermesi ile özdeşleştirilebilir. Karar alıcının çevresindekilerin de, alınan kararı iyi olarak nitelendirmeleri beklenir. Karar vericinin etkin ve ras-yonel olması, problemin mali boyutunu iyi analiz etmiş olması, geleceğe dönük bir analiz yapmış olması gereklidir. Karar verme, genel olarak, bir problem çözümleme süreci olarak da adlandırılabilir.

Karar verme sürecinde, seçeneklerin, alınacak kararı etkileyen etmenlerin çokluğu ve hızlı karar verme gerekliliğinin getirdiği karmaşıklık, karar vericinin vereceği kararlarda olumsuz bir etkiye sahip olabilmektedir. Olası tüm seçeneklerin ve bunlara bağlı olarak elde edilecek tüm sonuçların rakamsal olarak takip edilmesi, pek çok karar vericinin daha fazla iş yüküyle karşılaşmasına neden olabilmektedir. Karar probleminin karar vericinin karşısına çıkabilecek tüm sonuçları ya da senaryoları gösteren grafiksel bir yardımcı araca

Karar verme, karar vericinin karşılaştığı bir problem çözümünde olumlu bir sonuca ulaşabilmek için, problemin sunduğu birden fazla olası seçenek içerisinde seçim yapması işlemidir.

İhtiyacı olabilir. Böyle bir grafik karar vericinin ilgili karar probleminde karşılaşabileceği tüm durumları göstereceğinden, karar verme süreci çok daha rahat sonuçlandırılabilir. Bu amaçla, karar vericiler gerçek bir ağaç andırıldığı için karar ağaçları olarak adlandırılan grafiksel gösterimi kullanabilirler. Kisaca *karar ağaçları*, karar vericinin içinde bulunduğu karar verme probleminde ortaya çıkabilecek tüm durumları ve karar vericinin karşılaşabileceği tüm senaryoları bir arada gösterebilen bir grafiksel yaklaşımdır. Karar ağaçlarının bazı avantajları,

- Açıklanmalarının kolay olması,
- İnsani karar almayı diğer yaklaşılara göre daha iyi yansıtması,
- Grafiksel olarak gösterilebilir olması,
- Uzman olmayan kişiler tarafından da kolaylıkla yorumlanabilir olması,
- Temsili değişkenlere ihtiyaç duymadan nitel değişkenleri de işleyebiliyor olmalarıdır.

Karar ağaçları basit karar verme problemlerinde kullanışlı olmakla beraber, karar sayısı ve kararları etkileyen etki sayısının artması ile daha karmaşık bir yapıya da sahip olabilirler.

Veri madenciliği uygulamalarında karar vericinin sıkılıkla karşılaştiği problemlerden bir tanesi de sınıflandırma problemidir. Çok basit bir sınıflandırma örneği olarak, bir bankanın müşterilerini, müşterilerin çeşitli niteliklerini (gelir durumu, statüsü, borç durumu vb.) temel olarak, kredi uygunluk durumu gibi bir nitel değişkenle göre riskli, risksiz olarak iki ayrı sınıfa gruplandırılmasının verilebilir. Sınıflandırma problemleri, veri madenciliğinde istatistiksel veya mantıksal yaklaşımı sahip yöntemler ile ele alınabilmektedir. Örneğin, diskriminant analizi sınıflandırma problemine matematiksel işlemler yardımıyla istatistiksel bir yaklaşım sağlar iken karar ağaçları, evet-hayır şeklinde değerlendirilen ifadeler ve karşılaştırma işlemleri yardımıyla mantıksal bir yaklaşım sağlar. Karar ağaçları, veri madenciliğinde karşılaşılan sınıflandırma problemlerinin çözümü için en sık başvurulan mantıksal yaklaşım yöntemidir.

Sınıflandırma, bir kaydı, önceden tanımlanmış çeşitli sınıflardan birine atayan bir modelin uygulanması işlemi olarak tanımlanabilir. Sınıflandırma yapabilmek için, girdi olarak nitelik değerlerinden oluşan örnek kayıtlığını ve karşılık gelen bir sınıf verilmelidir. Sınıflandırma modeli ise, mevcut olan nitelik değerleri ile yeni bir kaydın sınıfının **kestirimini** yapar ve sınıflayıcı olarak adlandırılır. Ünitenin izleyen kesiminde, karar ağaçları ve ilgili bazı kavramlar inceleneciktir.

DİKKAT



Sınıflandırma, belirli bir sınıfın sınıflandırılmamış bir birime atanması sürecini, sınıflayıcısı ise diğer nitelikler verilmiş iken bir birimin sınıfını kestiren modeli ifade eder.

SIRA SİZDE

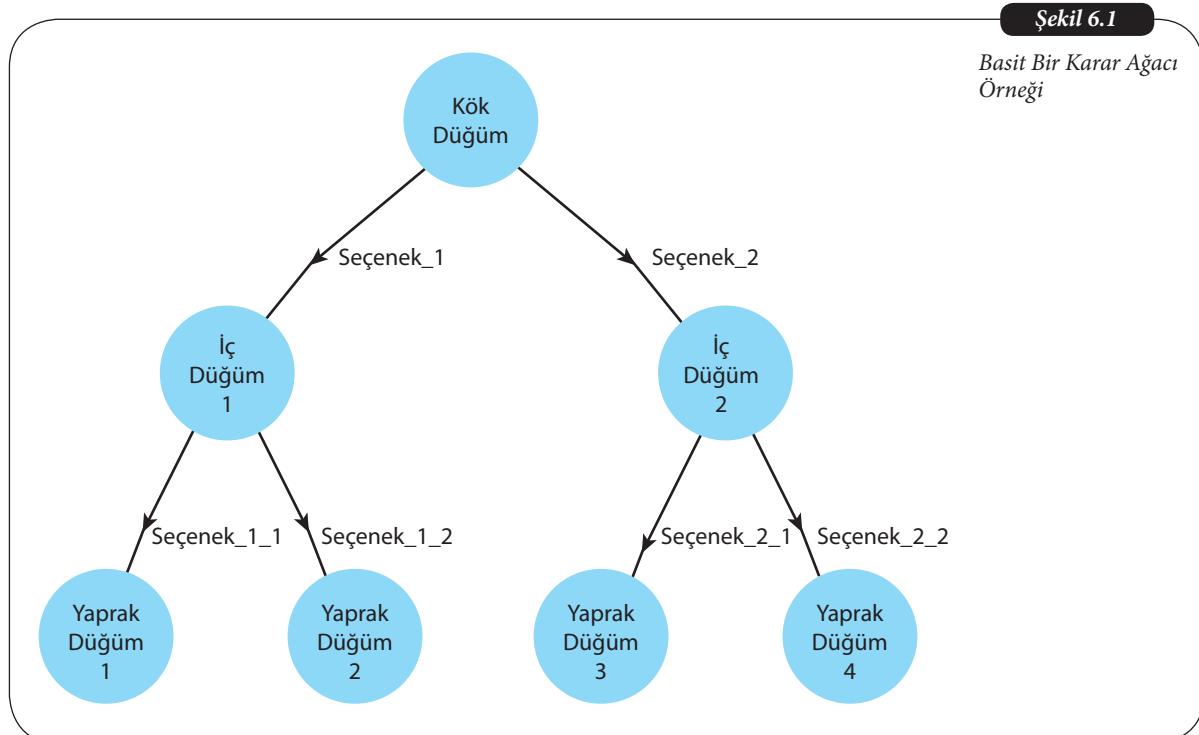
1



Bir karar verici olarak, yaşam alanınıza internet hizmeti almak istediğinizde, adsl, fiber optik ve gsm kablosuz mobil bağlantı seçenekleri ile karşı karşıya kalmaktasınız. Karar alternatifleriniz nelerdir? Karar verirken, seçiminizi etkileyebilecek etmenler var mı? Var ise, bu etmenleri açıklamaya çalışınız.

KARAR AĞAÇLARI

Sınıflandırma tekniklerinden birisi de karar ağacıdır. Karar ağacı ile ilgili bazı kavramların detaylı bir şekilde ele alınmasında büyük fayda bulunmaktadır. En basit anlamıyla karar ağacı, düğüm ve dal bileşenlerinden oluşan ve Şekil 6.1'de yer alan ağaca benzer bir yapıya sahip grafiksel bir tekniktir.



Problemde yer alan her bir nitelik için karar ağacında bir *düğüm* yer alır. Böylece niteliğin test edilmesi garanti altına alınır. Bir düğümden ayrılan *dallar* ise o düğümdeki testin tüm olası sonuçlarının her birine karşılık gelmektedir. Karar ağacının başlangıcını oluşturan ilk düğüm *kök düğüm* olarak adlandırılır. Karar ağacı bu düğümden başlayarak, problemin içerisindeki tüm karar seçeneklerini içerecek şekilde düğümlerin mantık sırasına göre eklenmesiyle tamamlanır. Son düğüm *yaprak düğüm*, diğer düğümler ise *İç düğüm* olarak adlandırılır. Yaprak düğümlerin her biri bir sınıfı temsil eder. Kimi sınıflandırma problemlerinde basit yapılı bir karar ağacı oluşturken, problemdeki nitelik sayısına bağlı olarak karar ağacı da karmaşık bir yapıya sahip olacaktır.

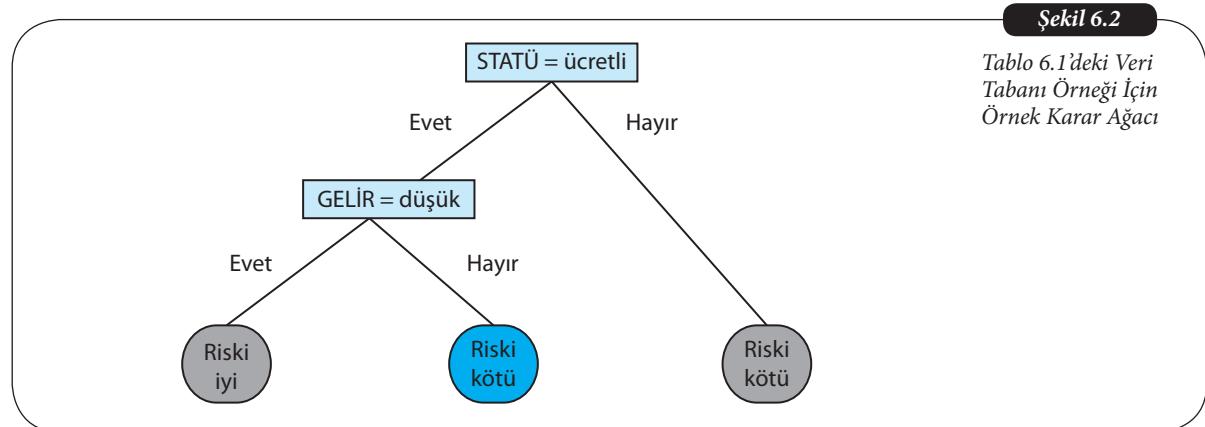
Örneğin bir banka, mevcut müşterilerini kredi risklerine göre sınıflamak ve yeni müşterilerini de bu sınıflandırmaya dahil etmek istesin. Bankanın müşteri veritabanı Tablo 6.1'dekine benzer bir yapıda olacaktır.

Kök ve iç düğüm bir karar ağacını başlatan ve büyütmen düğümler, **yaprak düğüm** ise dallanmayı sonlandıran düğümdür.

Tablo 6.1
Banka Müşteri Veri
Tabanı

MÜŞTERİ	BORÇ	GELİR	STATÜ	RİSK
1	yüksek	yüksek	işveren	kötü
2	yüksek	yüksek	ücretli	kötü
3	yüksek	düşük	ücretli	kötü
4	düşük	düşük	ücretli	iyi
5	düşük	düşük	işveren	kötü
6	düşük	yüksek	işveren	iyi
7	düşük	yüksek	ücretli	iyi
8	düşük	düşük	ücretli	iyi
9	düşük	düşük	işveren	kötü
10	düşük	yüksek	işveren	iyi
11	yüksek	yüksek	ücretli	kötü
12	düşük	yüksek	ücretli	kötü
13	düşük	yüksek	ücretli	kötü
14	yüksek	düşük	ücretli	iyi
15	düşük	düşük	ücretli	kötü
16	düşük	düşük	ücretli	kötü
17	düşük	düşük	işveren	kötü
18	düşük	düşük	işveren	kötü
19	düşük	düşük	işveren	kötü
20	düşük	düşük	işveren	iyi
21	düşük	yüksek	ücretli	iyi
22	düşük	yüksek	işveren	iyi
23	yüksek	yüksek	işveren	iyi
24	yüksek	yüksek	işveren	kötü
25	yüksek	yüksek	ücretli	iyi
26	düşük	düşük	ücretli	iyi
27	düşük	yüksek	ücretli	iyi
28	düşük	düşük	işveren	kötü
29	yüksek	yüksek	ücretli	kötü
30	düşük	düşük	ücretli	kötü
31	yüksek	düşük	işveren	kötü
32	yüksek	düşük	işveren	kötü
33	yüksek	düşük	işveren	iyi
34	yüksek	düşük	ücretli	iyi
35	düşük	yüksek	ücretli	iyi
36	düşük	yüksek	ücretli	iyi
37	düşük	yüksek	ücretli	kötü
38	düşük	yüksek	ücretli	kötü
39	düşük	yüksek	ücretli	kötü
40	yüksek	yüksek	ücretli	iyi
41	yüksek	düşük	ücretli	iyi
42	yüksek	düşük	ücretli	iyi
43	yüksek	düşük	ücretli	iyi
44	düşük	düşük	ücretli	iyi
45	yüksek	yüksek	işveren	iyi

Tablo 6.1'de yer alan veritabanı örneğinde, aslında bankanın, müşterilerinin kredi risklerini hangi kriterlere göre belirlediği de görülebilmektedir. Ancak, tablodan anlaşılmış gibi, bir karar vericinin bu tablo yardımıyla yeni müşterilerinin kredi riski sınıflandırılmasını yapması oldukça zordur. Tablo 6.1'de verilen müşteri veritabanından elde edilen borç, gelir ve statü niteliklerini içeren basit bir karar ağacı Şekil 6.2'de gösterilmektedir.



Şekil 6.2'de gösterilen karar ağacında, statü niteliğinin test edildiği ilk düğüm kök düğümdür. Kök düğümde, "statüsü ücretli mi?" sorusuna evet cevabı alındığında, gelir niteliğinin test edildiği düzgüme giden dal takip edilir. Bu düğümde de test edilen gelir niteliğinde "geliri düşük mü?" sorusuna da evet cevabı alındığında yaprak düzgüme ulaşmış olur. Dolayısı ile, statü ve gelir nitelikleri ilgili sıralamaya uyan tüm müşteriler ilgili yaprak düşüğünün temsil edildiği sınıfa dahil olur. Yaprak düğüm, karar ağacı modeli ile iletilen tüm bilgiyi içerir. Buna göre, örneğin statüsü ücretli, gelir düzeyi düşük olan tüm müşteriler kredi riski iyi sınıfına dahil edilmiş iken, statüsü ücretli olmayan tüm müşteriler, gelir düzeyi ne olursa olsun kredi riski kötü sınıfına dahil edilmiş olur. Bu modele göre, müşteriler 3 gruba sınıflandırılmıştır. Karar verici artık, yeni bir müşterisinden elde ettiği bilgileri kullanarak, oluşturduğu karar ağacı yardımıyla çok daha kolay bir şekilde müşterilerini sınıflayabilecektir. Karar ağacının matematiksel çözümünü elde edebilmek için çeşitli tanımlamaların yapılmasına ihtiyaç vardır.

$D = \{t_1, \dots, t_n\}$ veritabanını göstermek üzere, her $t_i, t_i = (t_{i1}, t_{i2}, \dots, t_{ih})$ ise ve $A = \{A_1, A_2, \dots, A_h\}$ niteliklerini gösteriyor iken, $C = \{C_1, C_2, \dots, C_m\}$ sınıf kümesi olarak tanımlanmış olsun. D müşteri veritabanı ile ilişkilendirilen karar ağacı izleyen özelliklere sahip olacaktır:

- Her bir düğüm A_i niteliği ile etiketlendirilecektir.
- Düğünden ayrılan her bir dal, ilgili düğüm ile ilişkili niteliğe uygulanabilen soru'nun yanıtlarıyla etiketlenecektir.
- Her bir yaprak düğüm C_i sınıfıyla etiketlenecektir.

Bu tanım kapsamında, daha önce ele aldığımız banka müşterileri veritabanı örneği karar ağacı tanımlamaları Tablo 6.2'deki gibi oluşturulur.

MÜŞTERİ	BORÇ (A_1)	GELİR (A_2)	STATÜ (A_3)	RISK (C)
1 (t_1)	yüksek	yüksek	işveren	kötü (C_1)
2 (t_2)	yüksek	yüksek	ücretli	Kötü (C_1)
...
44 (t_{44})	düşük	düşük	ücretli	iyi (C_2)
45 (t_{45})	yüksek	yüksek	işveren	iyi (C_2)

Tablo 6.2
Banka Müşteri Veri
Tabanı Üzerinde
Karar Ağacı Çözüm
Bileşenleri

Karar ağaçlarını, sınıflandırma probleminin çözümlenmesinde kullanırken iki adıma ihtiyaç duyulur. Bu adımlar,

1. Karar ağacının oluşturulması
2. Veritabanında yer alan her bir kaydın (t_i) sınıflandırmasının yapılması

şeklindedir. Karar ağıacı oluşturulduktan sonra, her bir kayıt bu karar ağıacının kök düğümünden başlayarak, geçtiği her düğümdeki sorunun yönlendirmesine göre bir yaprak düşüme ulaşır ve böylece sınıflandırma işlemi tamamlanmış olur. Bu süreçte karşılaşılabilen en önemli sorun, kök ve iç düğümlerde hangi niteliklerin yer alacağının tespit edilmesidir. Çünkü, sınırlı sayıda kayttan oluşan bir veri yiğini için olası tüm karar ağaçlarını oluşturmak ve bunların arasından en uygunu seçmek oldukça zor olacaktır. Bu nitelik, ayırmaya işlemini gerçekleştiren en iyi nitelik olacaktır ve ayırmaya kriteri olarak adlandırılır. Ayırmaya kriteri olarak öyle bir nitelik seçilmelidir ki diğer nitelikler ile karşılaştırıldığında en iyi ayırcı nitelik olmalıdır. Nitelik bir kez belirlendikten sonra, kök düğümün temsil ettiği niteliğin test sonuçlarının her biri için dallar oluşturulur. Örneğin, kök düğüm cinsiyet ise erkek seçeneği bir dalı kadın seçeneği ise diğer dalı oluşturacaktır. Kök düğümünden ayrılan dalların bağılılığı düğümlerin temsil edeceğii nitelikler de aynı kök düğümde olduğu gibi elde edilerek, karar ağıacının oluşturulmasına devam edilir. Bu ayırmaya işlemi, karar ağıacının büyümesi, örneğin bir düğümdeki kayıt sayısının bölünemeyecek kadar az olması veya ağaç derinliğinin araştırmacı tarafından belirlenen bir limite ulaşması gibi bir duruma kriterine ulaşılanada kadar devam edecektir. Ayırmaya ve duruma kriterleri, karar ağıacı oluşturmak için kullanılan algoritmalarla göre değişiklik göstermektedir.

Ayırmaya Kriterleri

Düğümün temsil ettiği, dolayısı ile ayırmaya işlemini en iyi şekilde gerçekleştirecek olan niteliğin seçilmesi, başka bir ifadeyle *ayırmaya kriterinin* belirlenmesi için çeşitli ölçüler geliştirilmiştir. Bu ölçüler, niteliğin veri tipine göre değişiklik göstermektedir. Nitel veri için Entropi İndeksi, Gini İndeksi, Sınıflandırma Hatası İndeksi ve Twoing ölçütleri kullanılır. Ek olarak Twoing ölçüsünün sıralı şekilde ölçülmüş değişkenlerin bulunduğu veri için Ordered Twoing bulunmaktadır. Nicel veriler için ise En Küçük Kareler Sapması yöntemi en sık kullanılan ölçütür. Bu ünitelerin izleyen kesiminde ilgili ölçütlerden Entropi İndeksi ve Gini İndeksi ayrıntılı olarak incelenmiştir.

Entropi İndeksi ile En İyi Ayırcı Niteliğin Seçilmesi

Entropi, bir veri yiğinındaki düzensizliğin, rassallığın miktarını ölçmek için kullanılan bir ölçütür. Veri yiğini içinde, örneğin bankanın oluşturduğu müşteri veritabanındaki müşterileri sınıflayan kredi riski niteliğinde, tek bir sınıf olması durumunda, entropinin 0 (sıfır) olması beklenir. Çünkü bir düzensizlikten veya rassallıktan söz edilemez. Bir başka deyişle, entropisi 0 olan bir grubun tam homojen bir grup, entropisi 1 olan grubun ise tam heterojen olduğu söylenebilir.

Bir veri yiğininin, sınıflayıcı niteliğinin alacağı değerler $C = \{C_1, C_2, \dots, C_m\}$ olmak üzere, m sınıfa ayırbilmesi için; ilgili sınıflar hakkında ortalama bilgiye ihtiyaç duyulmaktadır. T sınıf değerlerini içeren küme iken, P_T bu sınıfların olasılıklarını temsil etsin. Bu olasılık değerleri izleyen eşitlik yardımıyla hesaplanır.

$$P_T = \left\{ \frac{|C_1|}{|T|}, \frac{|C_2|}{|T|}, \dots, \frac{|C_m|}{|T|} \right\}$$

C_i sınıfının T kumesindeki sayısını tespit edebilmek amacıyla $|C_i|$ 'den yararlanılır. Bu durumda birinci sınıfın olasığı $P_1 = \frac{|C_1|}{|T|}$ olacaktır. T için entropi hesabı yapılabilmesi için izleyen eşitlik kullanılır.

$$H(T) = - \sum_{i=1}^m p_i \cdot \log_2 p_i$$

Entropi indeksi hesabını banka müşteri veritabanı örneği ile ele alabiliriz. Tablo 6.1'de verilen banka müşteri veritabanında yer alan müşterilerin kredi riskleri kümesi

$R = \{\text{kötü, kötü, kötü, kötü, iyi, iyi, kötü}\}$

olarak ortaya çıkmaktadır. C_1 iyi sonucunu temsil etmek üzere, C_1 için olasılık değeri

$$P_{iyi} = \frac{|C_1|}{|T|} = \frac{23}{45} = 0,51$$

ve C_2 kötü sonucunu temsil etmek üzere, C_2 için olasılık değeri,

$$P_{kötü} = \frac{|C_2|}{|T|} = \frac{22}{45} = 0,49$$

olarak hesaplanır. Bu hesaplama yardımıyla, Risk niteliğinin olasılık dağılımı

$$P_R = \left\{ \frac{|C_1|}{|T|}, \frac{|C_2|}{|T|} \right\} = \left\{ \frac{23}{45}, \frac{22}{45} \right\}$$

olacaktır. Dolayısıyla Risk'in entropisi ise;

$$\begin{aligned} H(\text{RISK}) &= - \sum_{i=1}^2 p_i \cdot \log_2(p_i) = -p_1 \log_2(p_1) - p_2 \log_2(p_2) \\ &= -(0,511 \cdot \log_2 0,511 + 0,489 \cdot \log_2 0,489) \\ &= 0,9996 \end{aligned}$$

değerini alacaktır

T hedef niteliği, hedef niteliği olmayan bir X niteliğine bağlı olarak T_1, T_2, \dots, T_m alt kümelerine ayrılmak istendiğinde, T 'nin bir elemanın sınıfını belirleyebilmek için bilgiye ihtiyaç duyulur. Bu bilgi, T_i 'nin bir elemanın sınıfının belirlenmesinde gerekli olan bilginin ağırlıklı ortalaması olarak kabul edilir. Hesaplanmanın gerçekleştirilemesi için izleyen eşitlik kullanılır.

$$H(X, T) = \sum_{i=1}^m \frac{|T_i|}{|T|} \cdot H(T_i)$$

T hedef niteliğini X niteliğine göre bölgerek elde edilen bilgiyi ölçmek için **kazanç ölçüyü**inden yararlanılır ve hesaplama için izleyen eşitlik kullanılır.

$$Kazanç(X, T) = H(T) - H(X, T)$$

En yüksek kazancı sağlayan nitelik, ayırcı nitelik olarak tanımlanır. Ancak, Entropi ve Gini indeksleri gibi indeksler belli değerleri çok sayıda bulunduran nitelikleri tercih etme eğilimindedirler. Dolayısıyla ayırmaya sayısı fazla olduğundan elde edilen sınıflar çok küçük sınıflar olacaktır. Bu durum araştırmacının güvenilir kestirimler yapmasını mümkün kılamayabilir. Benzer durumlarda kullanılan stratejilerden bir tanesi sadece ikili (binary) ayırmaya yapacak şekilde testler oluşturmak veya ayırmayı ne kadar iyi olduğunu belirlemek için kullanılan **kazanç oranı ölçütünü** kullanmaktadır. Bu ölçüt, T 'deki X 'in bölünme bilgisinden (split information) yararlanmaktadır. Kazanç oranı ölçütün hesaplanması izleyen eşitlik yardımıyla yürütülür.

$$Kazanç Oranı(X, T) = \frac{Kazanç(X, T)}{H(P_{X,T})}$$

eşitlikte, $(P_{X,T})$, X değerlerinin olasılık dağılımını temsil etmektedir ve

$$P_{X,T} = \left\{ \frac{|T_1|}{|T|}, \frac{|T_2|}{|T|}, \dots, \frac{|T_m|}{|T|} \right\}$$

eşitliği yardımıyla hesaplanır. Bölünme bilgisi $H(P_{X,T})$ ise,

$$H(P_{X,T}) = - \sum_{i=1}^m \frac{|T_i|}{|T|} \cdot \log_2 \left(\frac{|T_i|}{|T|} \right)$$

eşitliği yardımıyla hesaplanır.

Tablo 6.1'da yer alan banka müşterileri veritabanıörneğinde borç, gelir ve statü nitelikleri ele alındığında, hangi niteliğin en iyi ayırcı nitelik olduğu Entropi indeksi yardımıyla bulunmak istenebilir. Bu durumda borç niteliği için Entropi indeksi hesabında, öncelikle ilgili nitelikler arasında kontenjans tablosu hazırlanır. Kontenjans tablosu hazırlanırken, niteliğin aldığı her bir sonucun diğer niteliğin aldığı sonuçlar ile ilişkisi sayma yoluyla tespit edilir. Borç niteliği ile risk niteliği için Entropi indeksi hesabında kullanılacak kontenjans tablosu Tablo 6.3'de yer almaktadır.

Tablo 6.3
Borç Niteliği ile Risk
Niteliği Kontenjans
Tablosu

		Risk		
		İyi	Kötü	Toplam
Borç	Yüksek	10	8	18
	Kötü	13	14	27
	Toplam	23	22	45

İlgili kontenjans tablosu hazırlanıktan sonra borç niteliğinin tüm sonuçları için Entropi indeksi değerleri hesaplanır. İlgili değerler izleyen eşitliklerde yer almaktadır.

$$\begin{aligned} H(\text{BORÇ}_{\text{yüksek}}) &= - \sum_{i=1}^2 p_i \cdot \log_2 p_i \\ &= - \left(\frac{10}{18} \cdot \log_2 \frac{10}{18} + \frac{8}{18} \cdot \log_2 \frac{8}{18} \right) \\ &= 0,9911 \end{aligned}$$

$$\begin{aligned} H(\text{BORÇ}_{\text{düşük}}) &= - \sum_{i=1}^2 p_i \cdot \log_2 p_i \\ &= - \left(\frac{13}{27} \cdot \log_2 \frac{13}{27} + \frac{14}{27} \cdot \log_2 \frac{14}{27} \right) \\ &= 0,9990 \end{aligned}$$

İlgili Entropi indeksi değerleri hesaplandıktan sonra ağırlıklı ortalama değeri izleyen eşitlikteki gibi hesaplanır.

$$\begin{aligned} H(\text{BORÇ}) &= \sum_{i=1}^m \frac{|T_i|}{|T|} \cdot H(T_i) \\ &= \left(\frac{18}{45} \cdot 0,9911 + \frac{27}{45} \cdot 0,9990 \right) \\ &= 0,9958 \end{aligned}$$

Borç niteliği ile ayırma yapılması istendiğinde elde edilen kazanç,

$$\begin{aligned} \text{Kazanç(BORÇ, RİSK)} &= H(\text{RİSK}) - H(\text{BORÇ, RİSK}) \\ &= 0,9996 - 0,9958 = 0,0038 \end{aligned}$$

olacaktır.

Gelir niteliği ile risk niteliği için Entropi indeksi hesabında kullanılacak kontenjans tablosu Tablo 6.4'de yer almaktadır.

		Risk		
		İyi	Kötü	Toplam
Gelir	Yüksek	12	10	22
	Düşük	11	12	23
	Toplam	23	22	45

Tablo 6.4
Gelir Niteliği ile Risk Niteliği Kontenjans Tablosu

İlgili kontenjans tablosu hazırlandıktan sonra gelir niteliğinin tüm sonuçları için Entropi indeksi değerleri hesaplanır. İlgili değerler izleyen eşitliklerde yer almaktadır.

$$\begin{aligned} H(\text{GELİR}_{\text{yüksek}}) &= - \sum_{i=1}^2 p_i \cdot \log_2 p_i \\ &= - \left(\frac{12}{22} \cdot \log_2 \frac{12}{22} + \frac{10}{22} \cdot \log_2 \frac{10}{22} \right) \\ &= 0,9940 \end{aligned}$$

$$\begin{aligned} H(\text{GELİR}_{\text{düşük}}) &= - \sum_{i=1}^2 p_i \cdot \log_2 p_i \\ &= - \left(\frac{11}{23} \cdot \log_2 \frac{11}{23} + \frac{12}{23} \cdot \log_2 \frac{12}{23} \right) \\ &= 0,9986 \end{aligned}$$

İlgili Entropi indeksi değerleri hesaplandıktan sonra ağırlıklı ortalama değeri izleyen eşitlikteki gibi hesaplanır.

$$\begin{aligned} H(\text{GELİR}) &= \sum_{i=1}^m \frac{|T_i|}{|T|} \cdot H(T_i) \\ &= \left(\frac{22}{45} \cdot 0,9940 + \frac{23}{45} \cdot 0,9986 \right) \\ &= 0,9964 \end{aligned}$$

Gelir niteliği ile ayırma yapılması istendiğinde elde edilen kazanç,

$$\begin{aligned} \text{Kazanç(GELİR, RİSK)} &= H(\text{RİSK}) - H(\text{GELİR, RİSK}) \\ &= 0,9996 - 0,9964 = 0,0033 \end{aligned}$$

olacaktır.

Borç niteliği ile risk niteliği için Entropi indeksi hesabında kullanılacak kontenjans tablosu Tablo 6.5'de yer almaktadır.

Tablo 6.5
Statü Niteliği ile Risk
Niteliği Kontenjans
Tablosu

		Risk		
		İyi	Kötü	Toplam
Statü	İşveren	7	10	17
	Ücretli	16	12	28
	Toplam	23	22	45

İlgili kontenjans tablosu hazırlanıktan sonra gelir niteliğinin tüm sonuçları için Entropi indeksi değerleri hesaplanır. İlgili değerler izleyen eşitliklerde yer almaktadır.

$$\begin{aligned} H(\text{STATÜ}_{\text{işveren}}) &= - \sum_{i=1}^2 p_i \cdot \log_2 p_i \\ &= - \left(\frac{7}{17} \cdot \log_2 \frac{7}{17} + \frac{10}{17} \cdot \log_2 \frac{10}{17} \right) \\ &= 0,9774 \end{aligned}$$

$$\begin{aligned} H(\text{STATÜ}_{\text{ücretli}}) &= - \sum_{i=1}^2 p_i \cdot \log_2 p_i \\ &= - \left(\frac{16}{28} \cdot \log_2 \frac{16}{28} + \frac{12}{28} \cdot \log_2 \frac{12}{28} \right) \\ &= 0,9852 \end{aligned}$$

İlgili Entropi indeksi değerleri hesaplandıktan sonra ağırlıklı ortalama değeri izleyen eşitlikteki gibi hesaplanır.

$$\begin{aligned} H(\text{STATÜ}) &= \sum_{i=1}^m \frac{|T_i|}{|T|} \cdot H(T_i) \\ &= \left(\frac{17}{45} \cdot 0,9774 + \frac{28}{45} \cdot 0,9852 \right) \\ &= 0,9823 \end{aligned}$$

Statü niteliği ile ayırmaya yapılması istendiğinde elde edilen kazanç,

$$\begin{aligned} \text{Kazanç}(\text{STATÜ}, \text{RISK}) &= H(\text{RISK}) - H(\text{STATÜ}, \text{RISK}) \\ &= 0,9996 - 0,9823 = 0,0174 \end{aligned}$$

olacaktır.

Tüm nitelikler için elde edilen kazanç değerleri Tablo 6.6'da verilmiştir.

Tablo 6.6
Kazanç Ölçütü Tablosu

Nitelik	Kazanç Değerleri
BORÇ	0,0038
GELİR	0,0033
STATÜ	0,0174

Tablo 6.6'da verilen kazanç değerlerine göre banka müşterileri veritabanını sola ve sağa olmak üzere ikiye ayıracak olan kök düğümünü, en büyük kazanç değerine sahip olduğu için **statü niteliği** oluşturacaktır. Bu niteliğin, *ücretli* ve *işveren* olmak üzere iki sonucu mevcuttur, dolayısıyla düğümden ayrılan dallar "statüsü ücretli mi?" sorusunun cevabına göre ayrılacaktır. Daha önce Şekil 6.2'de gösterilen karar ağacında da, statü niteliğinin kök düğüm olduğu görülmektedir. Yukarıda yapılan işlemler, ilk ayırmaya işleminden sonra diğer dallar için de tekrarlanacaktır. İlk olarak, statüsü ücretli olan grubunu ayırmak için yeni ayırmaya işlemini en iyi şekilde gerçekleştirecek olan nitelik seçilecek, daha sonra ise ayırmaya işlemi gerçekleştirilecektir. Aynı ayırmaya işlemi, statüsü işveren olan grubu ayırmak için de tekrarlanacaktır. Bu iki dalın ayrılması için, müşteri veritabanında dikkate alınacak kısımlar Tablo 6.7 ve Tablo 6.8'de verildiği gibi olacaktır.

MÜŞTERİ	BORÇ	GELİR	STATÜ	RİSK
1	yüksek	yüksek	işveren	kötü
5	düşük	düşük	işveren	kötü
6	düşük	yüksek	işveren	iyi
9	düşük	düşük	işveren	kötü
10	düşük	yüksek	işveren	iyi
17	düşük	düşük	işveren	kötü
18	düşük	düşük	işveren	kötü
19	düşük	düşük	işveren	kötü
20	düşük	düşük	işveren	iyi
22	düşük	yüksek	işveren	iyi
23	yüksek	yüksek	işveren	iyi
24	yüksek	yüksek	işveren	kötü
28	düşük	düşük	işveren	kötü
31	yüksek	düşük	işveren	kötü
32	yüksek	düşük	işveren	kötü
33	yüksek	düşük	işveren	iyi
45	yüksek	yüksek	işveren	iyi

Tablo 6.7
Statüsü Niteliğinin Sağ Dalına Ayrılan Kayıtlar

MÜŞTERİ	BORÇ	GELİR	STATÜ	RİSK
2	yüksek	yüksek	ücretli	kötü
3	yüksek	düşük	ücretli	kötü
4	düşük	düşük	ücretli	iyi
7	düşük	yüksek	ücretli	iyi
8	düşük	düşük	ücretli	iyi
11	yüksek	yüksek	ücretli	kötü
12	düşük	yüksek	ücretli	kötü
13	düşük	yüksek	ücretli	kötü
14	yüksek	düşük	ücretli	iyi
15	düşük	düşük	ücretli	kötü
16	düşük	düşük	ücretli	kötü
21	düşük	yüksek	ücretli	iyi
25	yüksek	yüksek	ücretli	iyi
26	düşük	düşük	ücretli	iyi
27	düşük	yüksek	ücretli	iyi
29	yüksek	yüksek	ücretli	kötü
30	düşük	düşük	ücretli	kötü
34	yüksek	düşük	ücretli	iyi
35	düşük	yüksek	ücretli	iyi
36	düşük	yüksek	ücretli	iyi
37	düşük	yüksek	ücretli	kötü
38	düşük	yüksek	ücretli	kötü
39	düşük	yüksek	ücretli	kötü
40	yüksek	yüksek	ücretli	iyi
41	yüksek	düşük	ücretli	iyi
42	yüksek	düşük	ücretli	iyi
43	yüksek	düşük	ücretli	iyi
44	düşük	düşük	ücretli	iyi

Tablo 6.8
Statüsü Niteliğinin Sol Dalına Ayrılan Kayıtlar

Ek olarak, statü niteliği için kazanç oranı kriteri hesabı için izleyen eşitlikler kullanılır.

$$\text{Kazanç Oranı}(\text{Statü, Risk}) = \frac{\text{Kazanç}(\text{Statü, Risk})}{H(P_{\text{Statü,Risk}})}$$

$$P_{\text{Statü,Risk}} = \left\{ \frac{|T_1|}{|T|}, \frac{|T_2|}{|T|} \right\}$$

Ayırma bilgisini hesaplamak için kullanılacak $H(P_{\text{Statü,Risk}})$ ise,

$$\begin{aligned} H(P_{\text{Statü,Risk}}) &= - \sum_{i=1}^2 \frac{|T_i|}{|T|} \cdot \log_2 \left(\frac{|T_i|}{|T|} \right) = - \left[\frac{|T_1|}{|T|} \cdot \log_2 \left(\frac{|T_1|}{|T|} \right) + \frac{|T_2|}{|T|} \cdot \log_2 \left(\frac{|T_2|}{|T|} \right) \right] \\ &= - \left[\frac{17}{45} \cdot \log_2 \left(\frac{17}{45} \right) + \frac{28}{45} \cdot \log_2 \left(\frac{28}{45} \right) \right] = 0,28985 \end{aligned}$$

eşitliği yardımıyla hesaplanır. Statü niteliği için kazanç ölçütı, hatırlanacağı gibi daha önce 0,0174 olarak hesaplanmıştır. Dolayısıyla,

$$\text{Kazanç Oranı} = \frac{0,0174}{0,28985} = 0,0600$$

olarak hesaplanır.

Gini İndeksi ve Ayırıcı Niteliğin Belirlenmesi

Gini indeksi, ikili bölünmeye dayanan bir tekniktir. Bu indeksin hesaplanmasında nitelik değerlerinin sola ve sağa olmak üzere iki bölüme ayrılması işlemi yürütülür. Gini indeksi hesaplanması için izlenecek adımlar izleyen biçimde sıralanabilir;

1. Adım: Her nitelik değeri, sol ve sağ olmak üzere ikiye ayrılır, her bölüme karşılık gelen sınıf değerleri grupperlendirilir.

2. Adım: Her bir niteliğin sol ve sağ tarafta yer alan bölünmeleri için $Gini_{sol}$ ve $Gini_{sağ}$ değerleri hesaplanır. Bu adımda kullanmak üzere hesaplanacak olan Gini indeks değerlerinin tespit edilmesi için izleyen eşitlikler kullanılır.

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

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

Eşitliklerde, m sınıf sayısını, T bir düğümdeki örnekleri, $|T_{sol}|$ ve $|T_{sağ}|$ sol taraftaki ve sağ taraftaki örnek sayılarını, L_i ve R_i ise sol ve sağ tarafta yer alan i kategorisindeki örnek sayılarını ifade etmektedir.

3. Adım: Her bir j niteliği için, n düğümdeki örnek sayısı iken, Gini indeksinin ağırlıklı ortalaması izleyen eşitlik yardımıyla hesaplanır:

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

4. Adım: Her bir j niteliği için hesaplanan $Gini_j$ değerleri arasında en küçük olan seçilir, bölünme işlemi bu nitelik üzerinden gerçekleştirilir.

5. Adım: Bu adıma kadar yapılan tüm işlemler, karar ağacına yeni bir düğüm eklenemeye kadar tekrarlanır.

Daha önce incelediğimiz ve Tablo 6.1'de verilen banka müşterileri veritabanı örneğindeki borç, gelir ve statü nitelikleri ele alındığında, hangi niteliğin en iyi ayrıcı nitelik olduğu Gini indeksi yardımıyla da hesaplanabilir. Bu amaçla, öncelikle risk ile (borç, gelir, statü) nitelikleri arasındaki ilişkiyi göstermek üzere kontenjans tablosu Tablo 6.9'da görüldüğü gibi hazırlanır.

		Borç		Gelir		Statü	
		Yüksek	Düşük	Yüksek	Düşük	İşveren	Ücretli
Risk	Kötü	8	14	10	12	10	12
	İyi	10	13	12	11	7	16
	Toplam	18	27	22	23	17	28

Tablo 6.9
Nitelikler İçin
Kontenjans Tablosu

Tablo 6.9 incelediğinde, risk'in hedef veya sınıf niteliği olduğu görülmektedir. Risk ile borç niteliği arasındaki ilişkiye bakıldığında, borç niteliğinin iki sonucundan birincisi olan yüksek sonucu sol tarafa, diğer sonucu ise sağ tarafa ayrılacaktır. Borç niteliğinin sol ve sağ dalları için Gini indeksleri;

$$Gini_{Borç, \text{sol}} = 1 - \sum_{i=1}^2 \left(\frac{L_i}{|T_{\text{sol}}|} \right)^2 = 1 - \left[\left(\frac{8}{18} \right)^2 + \left(\frac{10}{18} \right)^2 \right] = 0,4938$$

$$Gini_{Borç, \text{sağ}} = 1 - \sum_{i=1}^2 \left(\frac{R_i}{|T_{\text{sağ}}|} \right)^2 = 1 - \left[\left(\frac{14}{27} \right)^2 + \left(\frac{13}{27} \right)^2 \right] = 0,4993$$

olacaktır. Borç niteliği için genel Gini indeksi değeri ise;

$$\begin{aligned} Gini_{Borç} &= \frac{1}{n} (|T_{\text{sol}}| \cdot Gini_{\text{sol}} + |T_{\text{sağ}}| \cdot Gini_{\text{sağ}}) \\ &= \frac{1}{45} (18 \cdot 0,4938 + 27 \cdot 0,4993) \\ &= 0,4971 \end{aligned}$$

olacaktır. Borç niteliğine benzer şekilde Gelir niteliğinin sol ve sağ dalları için Gini indeksleri;

$$Gini_{Gelir, \text{sol}} = 1 - \sum_{i=1}^2 \left(\frac{L_i}{|T_{\text{sol}}|} \right)^2 = 1 - \left[\left(\frac{10}{22} \right)^2 + \left(\frac{12}{22} \right)^2 \right] = 0,4959$$

$$Gini_{Gelir, \text{sağ}} = 1 - \sum_{i=1}^2 \left(\frac{R_i}{|T_{\text{sağ}}|} \right)^2 = 1 - \left[\left(\frac{12}{23} \right)^2 + \left(\frac{11}{23} \right)^2 \right] = 0,4991$$

olacaktır. Gelir niteliği için genel Gini indeksi;

$$\begin{aligned} Gini_{Gelir} &= \frac{1}{n} (|T_{\text{sol}}| \cdot Gini_{\text{sol}} + |T_{\text{sağ}}| \cdot Gini_{\text{sağ}}) \\ &= \frac{1}{45} (22 \cdot 0,4959 + 23 \cdot 0,4991) \\ &= 0,4975 \end{aligned}$$

olacaktır.

Son olarak, Statü niteliğinin sol ve sağ dalları için Gini indeksleri;

$$Gini_{\text{Statü, sol}} = 1 - \sum_{i=1}^2 \left(\frac{L_i}{|T_{\text{sol}}|} \right)^2 = 1 - \left[\left(\frac{10}{17} \right)^2 + \left(\frac{7}{17} \right)^2 \right] = 0,4844$$

$$Gini_{\text{Statü, sağ}} = 1 - \sum_{i=1}^2 \left(\frac{R_i}{|T_{\text{sag}}|} \right)^2 = 1 - \left[\left(\frac{12}{28} \right)^2 + \left(\frac{16}{28} \right)^2 \right] = 0,4898$$

olacaktır. Statü niteliği için genel Gini indeksi;

$$\begin{aligned} Gini_{\text{Statü}} &= \frac{1}{n} (|T_{\text{sol}}| \cdot Gini_{\text{sol}} + |T_{\text{sag}}| \cdot Gini_{\text{sag}}) \\ &= \frac{1}{45} (17 \cdot 0,4844 + 28 \cdot 0,4898) \\ &= 0,4878 \end{aligned}$$

değerini alacaktır.

Borç, Gelir ve Statü nitelikleri için hesaplanan Gini indeksleri Tablo 6.10'da bir araya getirilmiştir. Tablo 6.10'a göre, en düşük Gini indeksine sahip olan statü niteliği, kök düşüm için en iyi ayırcı nitelik olarak belirlenir.

Tablo 6.10
Niteliklere İlişkin Gini
İndeks Değerleri

Nitelik	Gini İndeksi
BORÇ	0,4971
GELİR	0,4975
STATÜ	0,4878

Karar Ağacı Oluşturma Algoritmaları

Sınıflandırma problemlerinde bir karar ağacının oluşturulması için farklı algoritmaların yararlanılabilir. Bu algoritmalar örnek olarak ID3, C4.5, CART, CHAID, QUEST, SLIQ, SPRINT ve MARS verilebilir. Bu algoritmalar, veri yiğinini *işleme şecline* ve kullanılan ayırma kriterine göre değişiklik göstermektedir. Üniteye izleyen kesiminde bu algoritmaların bazlarını kısaca ele alınacaktır.

ID3 algoritması en basit karar ağacı oluşturma algoritmasıdır. Ayırma kriteri olarak kazanç ölçütünden yararlanılmaktadır. Karar ağacının büyümesini durdurma kriteri ise tüm kayıtların tek bir sınıfa ait olması veya kazanç ölçütünün sıfırdan büyük olmaması durumudur. ID3 algoritmasında, karar ağacına herhangi bir budama işlemi uygulanmaz, ek olarak bu algoritma sayısal (ölçüm düzeyi nicel) nitelikleri ve kayıp veriyi işleyememektedir. 1983 yılında Ross Quinlan tarafından önerilmiştir.

C4.5 algoritması, ID3 algoritmasının geliştirilmiş hâlidir. Ayırma kriteri olarak kazanç oranından yararlanılmaktadır. Karar ağacının büyümesini durdurma kriteri, ayrılacak olan kayıtların sayısının belirli bir eşin altına düşmesi durumudur. C4.5 algoritmasında, karar ağacının büyümeye sahafından sonra, sınıflandırma hatasına dayanan budama işlemi uygulanmaktadır. C4.5 algoritması sayısal nitelikleri, ID3 algoritmasından farklı olarak da düzeltilmiş kazanç oranı ölçütünü kullanır, dolayısıyla kayıp veri içeren veri kümesi C4.5 algoritması ile işlenebilir. 1993 yılında Ross Quinlan tarafından önerilmiştir.

Kısaca CART olarak adlandırılan sınıflandırma ve regresyon ağaçları algoritması, ikili (binary) karar ağacı yapısından dolayı diğer algoritmaların farklılığı göstermektedir. Karar ağacındaki her bir düğüm yalnızca iki dala ayrılır. Ayırma kriteri için Entropi, Gini ve Twoing indekslerinden, karar ağacını budamak için ise maliyet-karmaşıklığı kriterinden faydalananır. CART algoritmasının önemli bir işlevi ise, yaprak düğümlerinde bir sınıf kestirimini yerine sayısal bir değer kestirimini içeren regresyon ağacının da oluşturulabilmesidir. Bu durumda, ayırma kriteri olarak en küçük kareler sapması kriterine başvurulmaktadır. 1984 yılında Breiman, Friedman, Olshen ve Stone tarafından önerilmiştir.

CHAID algoritması ilk olarak sayısal olmayan (ölçüm düzeyi sınıflayıcı) nitelikleri işleyebilecek şekilde geliştirilmiştir. CHAID algoritmasında, her girdi niteliği için, hedef niteliğe göre en az anlamlılıktaki farka sahip değer çiftleri bulur. CHAID algoritmasında, anlamlı olarak adlandırılan fark, istatistiksel bir testten elde edilen değeri ile ölçülür. Hedef, yani sınıf nitelik sürekli ise F testi, sınıflayıcı ise Pearson Ki-Kare testi, sıralayıcı ölçekle ölçülmüş ise maksimum benzerlik oranı testinden yararlanılmaktadır. CHAID algoritmasında, her seçilen değer çifti için elde edilen değerinin belli bir birleştirme eşik değerinden daha büyük olup olmadığı kontrol edilir ve olumlu sonuç alınan değerler doğru birleştirilir. Algoritma daha sonra da birleştirilecek potansiyel değer çiftleri aramaya devam eder. Bu arama işlemi, birleştirilecek anlamlı değer çiftlerinin bulunamamasına kadar devam eder. Bu işlem, her bir iç düğümü seçilen niteliğin homojen bir grubu yapacak biçimde, ayırma için en iyi niteliği seçmiş olur. Ayırma işlemi için kriter, en iyi ayırma niteliğinin düzeltilmiş değerinin belli bir ayırma eşik değerinden küçük olmasıdır. Ayrıca CHAID algoritması, en büyük ağaç derinliğine ulaşıldığında, bir ana düğüm olarak en küçük sayıda kayıta sahip olunması nedeniyle, düğümün daha fazla bölünmemesi ve alt düğüm olarak en küçük sayıda kayıta sahip olunması durumlarında da bu süreci durdurur. CHAID algoritmasında, oluşturulan karar ağacına budama uygulanmaz ve kayıp verinin söz konusu olması durumunda hepsini tek bir geçerli kategori olarak dikkate alarak işlem yürütülür. 1980 yılında Gordon V. Kass tarafından önerilmiştir.

QUEST algoritması, tek değişkenli ve doğrusal kombinasyon ayırmaları destekler. Her ayırma için (sıralayıcı veya sürekli niteliklerde) ANOVA F testi, Levene testi veya (sınıflayıcı niteliklerde) Pearson Ki-Kare testi kullanılarak, girdi niteliklerinin her biri ile hedef yani sınıf niteliğinin arasındaki birlaklık hesaplanır. Eğer hedef nitelik çok terimli (multinominal) ise 2-ortalamalar kümeleme tekniği ile iki süper sınıf oluşturur. Hedef nitelik ile en yüksek birlaklıği elde eden girdi nitelik en iyi ayırıcı nitelik olarak seçilir. Girdi niteliğinin optimal ayırma noktasını bulmak için Karesel Ayırma Analizi uygulanır. QUEST ihmäl edilebilir bir yanılığa sahiptir ve ikili (binary) karar ağaçları oluşturmada kullanılır. Oluşan ağaçları budamak için Ten-fold çapraz doğrulama metodu kullanılır. 1997 yılında Loh ve Shih tarafından önerilmiştir.

Karar Ağacı Budama Süreci ve Karar Ağacının Performansının Test Edilmesi

Budama bir ya da daha fazla dalı çıkartarak, karar ağacını daha basitleştirmek amacıyla, yaprak düğüm ile değiştirme işlemidir. Bu işlem, çıkartılmasına karar verilen dalın içerdığı kayıtların, bağlı olduğu üst düğüme dahil edilerek, düğümün yaprak düşüme dönüştürülmesine dayanır. Böylece, *kestirim hata oranının*, ortaya çıkan *aşırı uyum* (overfitting) sorununun giderilmesi, azaltılması ve sınıflandırma modelinin kalitesinin artırılması hedeflenir. Kisaca ifade etmek gerekirse, karar ağacının en iyi duruma getirilmesi işlemidir. Budama işlemi, gerekli görülmeli hâlinde, büyümesi önceden belirlenmiş olan durma kriterine göre sonlandırılmış karar ağacına uygulanabileceği gibi, durma kriterini daha esnek tanımlayarak ağacın olabildiğince büyümesi sağlanıktan sonra, en iyi duruma getirmek için de kullanılabilir. Budama, özellikle çok az sayıda kayıt bulunduran yaprak düğümlein kesilmesi bakımından önemlidir. Ancak, çok fazla budanmış bir karar ağacı ise, örnek uzayı hakkında yeterli bilgi sağlamayacaktır.

Budama süreci için çeşitli yöntemler geliştirilmiştir. Bu yöntemlerden bazıları maliyet karmaşıklığı (cost complexity), kötümser hata (pessimistic error), hata-karmaşıklığı (error complexity), kritik değer (critical value), azaltılmış hata (reduced error), en küçük-hata (minimum-error) budama yöntemleridir.

Çoğu teknikte olduğu gibi karar ağacı oluşturulurken de, veritabanının bir kısmı *modeli oluşturmak için kullanılırken*, kalan kısmı ise *oluşturulan modelin test edilebilmesi için ayrılr*. Veriyi ikiye ayırmadan amacı, kullanılan karar ağacı algoritmasının ortaya çıkardığı sınıflandırmanın test için saklanan veri ile tekrar denenerek, elde edilen sonuçlar arasında anlamlı bir farklılık olup olmadığı tespit edilmesidir. Bu tespit, elde edilen modelin performansını ölçen bir tespitidir. Bu amaca yönelik olarak kullanılan tekniklerden bazıları *hold-out* teknigi, *tekrarlı hold-out* (repeated hold-out) teknigi, çapraz-doğrulama (cross-validation) teknigi ve *bootstrap* teknigidir.

Hold-out teknigi, veritabanının, araştırmacının takdirinde olan bir oranda (yarı yarıya veya 1/3'e 2/3 gibi) iki ayrı gruba bölünerek, eğitim ve test verisi olarak ele alınmasına dayanır. Böylece sınıflandırmanın doğruluğu, eğitim verisi ile elde edilen karar ağacı modelinin test verisi üzerindeki doğruluğuna göre tahmin edilebilir.

Tekrarlı hold-out teknigi ise hold-out tekniginin çoklu tekrarına dayanmaktadır. Toplam doğruluk, her bir tekrrarda elde edilen model doğruluklarının aritmetik ortalaması olarak ifade edilir.

Çapraz-doğrulama yönteminde ise, veritabanı iki eşit gruba bölünür ve birinci grup eğitim verisi olurken ikinci grup test verisi olarak ele alınır. Daha sonra, grupların rolleri değiştirilir. Modelin hatası, bu iki denemenin hataları toplamına eşittir. 2-katlı çapraz doğrulama olarak da adlandırılan bu yöntem, k-katlı olarak genelleştirilebilir. Bu durumda, veritabanı eşit büyüklükte k tane gruba bölünür. Gruplardan bir tanesi test verisi olarak seçilirken, diğer gruplar eğitim verisi olarak ele alınır ve k grubun her birisi bir kez test verisi olacak şekilde bu işlem tekrarlanır. Toplam hata, tekrarların hataları toplamına eşit kabul edilir.

SIRA SİZDE

2

Tablo 6.8'de verilen Banka müşterileri veritabanını, kök düğümün ayırıcı niteliği olan statü niteliğinin ücretli kategorisi ile ayrılan kayıt grubunu en iyi ayıracak niteliği, Gini indeksini kullanarak belirleyiniz.

SINIFLANDIRMA VE REGRESYON AĞAÇLARININ R ÇÖZÜMÜ

Sınıflandırma ve regresyon ağaçları (CART), veri madenciliği sürecinde karşılaşılan sınıflandırma problemlerinde oldukça sık kullanılan bir yöntemdir. İkili (binary) karar ağaçları oluşturulduğu için diğer algoritmalarдан ayrılmaktadır. Karar ağacındaki her bir düğüm sadece iki dala ayırır. Ayırma kriteri için Entropi, Gini ve Twoing indekslerinden, karar ağacını budamak için ise maliyet-karmaşıklığı kriterinden yararlanmaktadır. CART algoritmasının önemli bir işlevi ise, yaprak düğümlerde bir sınıf kestirimini yerine sayısal bir değer kestirimini içeren regresyon ağacı da oluşturabilmesidir. Ünitenin izleyen kesimalinde, sınıflandırma probleminde CART sınıflandırma ve regresyon ağacı uygulaması R yazılımı ile yürütülecektir.

Öncelikle üzerinde çalışılan probleme ait verinin R ortamına aktarılması gerekmektedir. R'ye veri aktarmak için birkaç yöntem bulunmaktadır. R'ye veri aktarımı konusunu ele alırken örnek olay olarak Tablo 6.1'deki veri seti burada da kullanılacaktır.

İNTERNET



Ünite 2'den de hatırlanacağı gibi R yazılımı <https://cran.r-project.org/> linkinden ücretsiz olarak indirilebilir.

R'ye Veri Aktarma

R'ye veri aktarmanın birçok yöntemi mevcuttur. Bu yöntemlerden bazıları csv (comma separated values) türü dosya ile veri aktarımı, kopyala-yapıştır yöntemi ve veritabanı bağlantısı ile veri aktarım yöntemidir. Bu yöntemlere kısaca ünitenin izleyen kısmında yer verilmiştir.

Harun Sönmez (2006). R Yazılımı ile İstatistiksel Analiz, Kaan Kitabevi.



csv Dosyası ile R'ye Veri Aktarma

csv dosya türü, günümüz veri işleme (işlem tablosu, veritabanı vb) uygulamalarının tümünde standart olarak kullanılan bir dosya türüdür. Birçok yazılım csv türü dosya oluşturma ve işleme yeteneklerine sahiptir. Herhangi bir yazılım kullanılarak, elinizdeki veri setinin csv dosya türünde kaydedilmesi gereklidir. Bu esnada csv türü için (MS-DOS uyumlu, Windows uyumlu gibi) farklı alternatifler karşınıza çıkabilir. Bu nedenle, varsa değişken isimlerinde karakter sorunu çıkmaması için değişken isimlendirmelerinin uygun biçimde yapılmış olması işlemleri daha hızlandırmış olacaktır.

csv dosyası herhangi bir metin editörü ile bir kez açılarak, içeriğinin kontrol edilmesi yerinde olacaktır. Çünkü kullanılan işletim sistemi, uygulamaya özel sebepler ve işletim sisteminin yerel dil ayarları gibi çeşitli nedenlerden dolayı, csv dosyasında farklılıklar oluşabilir.

Şekil 6.3

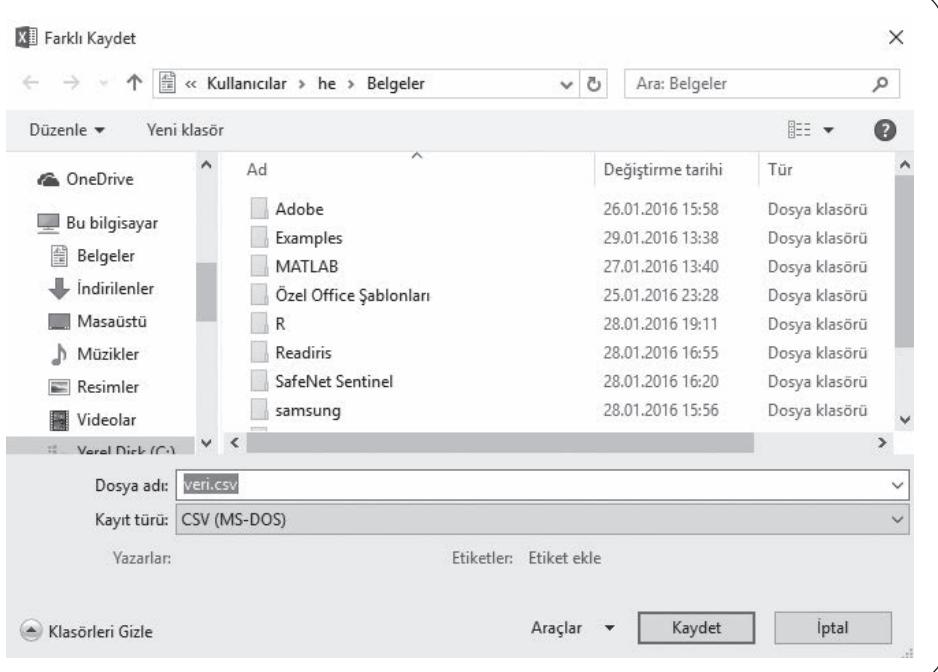
	A	B	C	D	E
1	MÜŞTERİ	BORÇ	GELİR	STATÜ	RISK
2	1	yüksek	yüksek	işveren	kötü
3	2	yüksek	yüksek	ücretli	kötü
4	3	yüksek	düşük	ücretli	kötü
5	4	düşük	düşük	ücretli	iyi
6	5	düşük	düşük	işveren	kötü
7	6	düşük	yüksek	işveren	iyi
8	7	düşük	yüksek	ücretli	iyi
9	8	düşük	düşük	ücretli	iyi
10	9	düşük	düşük	işveren	kötü
11	10	düşük	yüksek	işveren	iyi
12	11	yüksek	yüksek	ücretli	kötü
13	12	düşük	yüksek	ücretli	kötü
14	13	düşük	yüksek	ücretli	kötü

Banka Müşterileri
Veri Tabanı İçin İşlem
Tablosu Görünümü

Banka müşterileri veritabanı, işlem tablosu uygulamasına, Şekil 6.3'de de görüldüğü gibi değişken isimleri borç, gelir, statü ve risk olmak üzere ilk satırдан itibaren tanımlanmıştır. Veritabanını csv dosya türünde kaydedebilmek için ilgili uygulamanın Dosya menüsünden Kaydet seçeneği kullanılarak kayıt işlemi gerçekleştirilir. Şekil 6.4'te örnek bir işlem ekranı görülmektedir.

Şekil 6.4

Farklı Kaydet Seçeneği
ile csv Dosyasının
Kaydedilmesi



Şekil 6.5 “veri.csv” ismi ile kaydedilmiş bir dosyanın *Not Defteri* uygulamasındaki görselini içermektedir. Şekil 6.5’ten de görülebileceği gibi ilgili dosyanın ilk satırında değişken isimleri yer almaktadır. Ek olarak her bir veri noktası için değişkenlerin aldığı değerlerin de ‘,’ ile ayrıldığı görülmektedir. Dolayısı ile bu veri bir başka yazılımla kullanıldığında dosyanın oluşturulma yapısının hatırlanmasında fayda vardır. Aksi takdirde, değişkenler ve veri noktaları ilişkisinde istenmeyen kayıtlar ortaya çıkabilir.

Şekil 6.5

csv Dosyasının
Not Defteri
Uygulamasındaki
Görünümü

Dosya	Düzen	Birim	Görünüm	Yardım
MÜŞTERİ,BORÇ,GELİR,STATÜ,RİSK				
1,yüksek,yüksek,işveren,kötü				
2,yüksek,yüksek,ücretli,kötü				
3,yüksek,düşük,ücretli,kötü				
4,düşük,düşük,ücretli,iyi				
5,düşük,düşük,işveren,kötü				
6,düşük,yüksek,işveren,iyi				
7,düşük,yüksek,ücretli,iyi				
8,düşük,düşük,ücretli,iyi				
9,düşük,düşük,işveren,kötü				
10,düşük,yüksek,işveren,iyi				

Üzerinde çalışılması düşünülen müşteri bilgisini içeren veritabanı artık R’ye aktarılacak için hazır durumdadır. Aktarım için *read.csv()* fonksiyonundan yararlanılır. Bu fonksiyonun kullanımında gerekli olan parametreler *file*, *header*, *sep* ve *dec* parametreleridir. Diğer parametreler için R komut satırına *help(read.csv)* komutunu yazarak yardım alınabilir. Csv dosyamız “F:/” dosya yolunda “veri.csv” şeklinde kaydedildiği varsayılsrsa, izleyen komut satırları yardımıyla veri aktarım işlemi gerçekleştirilecektir.

```
>veri<-read.csv(file="F:/veri.csv",header=TRUE,sep="")
>veri[1:5,]
MÜŞTERİ BORÇ GELİR STATÜ RİSK
1    1 yüksek yüksek işveren kötü
2    2 yüksek yüksek ücretli kötü
3    3 yüksek düşük ücretli kötü
4    4 düşük düşük ücretli iyi
5    5 düşük düşük işveren kötü
```

file parametresi veri dosyasının kayıt ortamındaki konumunu ve dosya adını, *header* parametresi değişken isimlerinin ilk satırda olduğunu *True* değeri (yoksa *False*) atanarak, *sep* parametresi ise veri noktalarını ayıran simbolün ";" olduğunu belirtmektedir. *Dec* parametresinin kullanımı, sayısal bir nitelik olmadığından gerekmemektedir. Eğer sayısal bir nitelik söz konusu olursa, ondalıklı sayının ondalık ayıracını ifade eden simbol bu parametre ile verilmelidir ("." veya ";"; örneğin burada ondalık ayıracı ";" olsaydı, veri noktalarını ayıran simbol ";" olarak atanacaktı). Enter'a basıldığında "<" atama operatörü yardımıyla *veri* isimli değişkene tüm veri aktarılacaktır. *veri[1:5,]* komutu, ilk 5 kaydın görüntülenmesini sağlamaktadır, sadece değişken ismi yazıldığında tüm veri görüntülenecektir.

Kopyala-Yapıştır Komutu ile R'ye Veri Aktarma

Üzerinde çalışılan verinin küçük olması durumunda, veri aktarımı için tercih edilebilecek yöntem kopyala-yapıştır (Control+C/Control+V) yöntemidir. Bu yöntemde, öncelikle islemtablosuna girilmiş olan verinin ihtiyaç duyulan kısmının kopyalanması gereklidir. Şekil 6.6'da ilk 10 kayıt seçilmiştir. Seçme işleminden sonra işletim sisteminin uygun kopyala-ma tuş düzeni ile kopyalama işlemi gerçekleştirilir.

Şekil 6.6

	A	B	C	D	E
1	MÜŞTERİ	BORÇ	GELİR	STATÜ	RİSK
2	1	yüksek	yüksek	işveren	kötü
3	2	yüksek	yüksek	ücretli	kötü
4	3	yüksek	düşük	ücretli	kötü
5	4	düşük	düşük	ücretli	iyi
6	5	düşük	düşük	işveren	kötü
7	6	düşük	yüksek	işveren	iyi
8	7	düşük	yüksek	ücretli	iyi
9	8	düşük	düşük	ücretli	iyi
10	9	düşük	düşük	işveren	kötü
11	10	düşük	yüksek	işveren	iyi
12	11	yüksek	yüksek	ücretli	kötü
13	12	düşük	yüksek	ücretli	kötü

Banka Müşterileri
Veri Tabanında
Veri Kopyalama
Görünümü

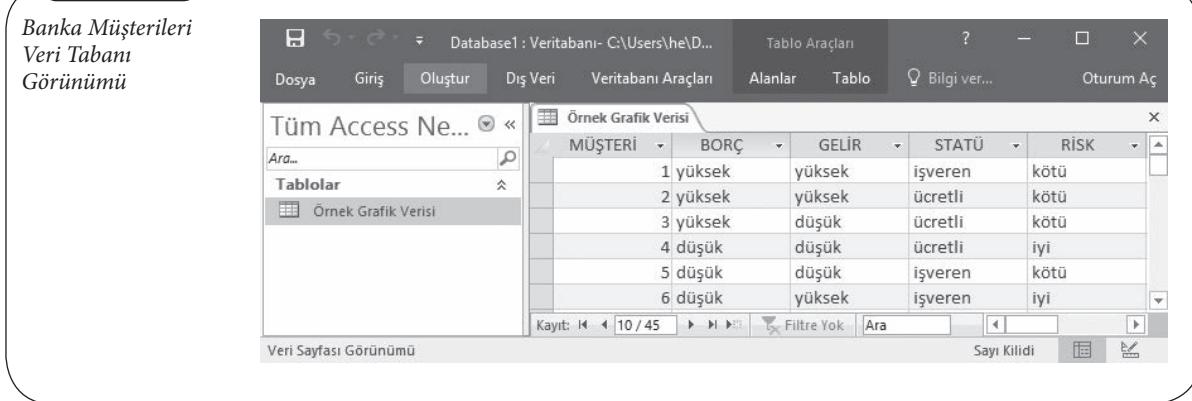
Kopyalama işleminden sonra R uygulamasına geçiş yapılır. R'de *read.table()* fonksiyonu bu aşamada faydalı olacaktır. Bu fonksiyon için kullanılan parametreler *read.csv* komutu ile aynıdır. Ancak, kopyala-yapıştır yardımıyla veri aktarılacağı için *file* parametresi veri yolu yerine "clipboard" ifadesine sahip olacaktır, *sep* parametresi sekme anlamına gelen "\t" ile kullanılacaktır. Komutun kullanımına dair detaylı bilgiler *help(read.table)* komutu ile elde edilebilir. İzleyen komut satırı ile veri aktarımı işlemi kopyala-yapıştır yöntemi ile tamamlanmış olur.

```
>veri1<-read.table(file="clipboard",header=TRUE,sep="/t")
>veri1
MÜŞTERİ BORÇ GELİR STATÜ RISK
1 1 yüksek yüksek işveren kötü
2 2 yüksek yüksek ücretli kötü
3 3 yüksek düşük ücretli kötü
4 4 düşük düşük ücretli iyi
5 5 düşük düşük işveren kötü
6 6 düşük yüksek işveren iyi
7 7 düşük yüksek ücretli iyi
8 8 düşük düşük ücretli iyi
9 9 düşük düşük işveren kötü
10 10 düşük yüksek işveren iyi
```

Veritabanı Erişimi ile R'ye Veri Aktarma

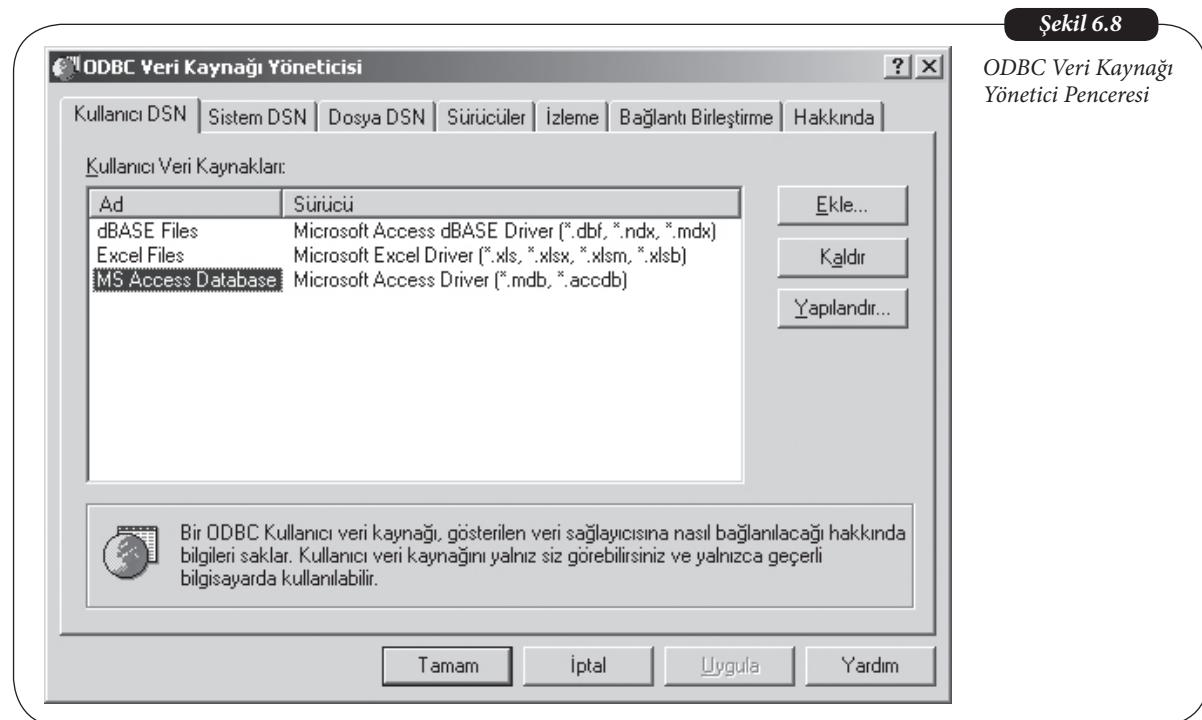
Bu bağlantı, Microsoft Windows işletim sisteminde yer alan **ODBC** (Open Database Connectivity) bağlantı türüdür. Bu bağlantı türü, farklı veritabanı sistemlerine standart teknikler ile bağlantı yapılmasını sağlar. Şekil 6.7'de, güncel veritabanı sistemlerinden bir tanesinin (MS Access) ekran görüntüsü verilmiştir.

Şekil 6.7



Microsoft Windows işletim sisteminde, var olan bağlantıları öğrenmek veya yoksa bir bağlantı yaratmak için **Denetim Masası -> Yönetimsel Araçlar -> Veri Kaynakları (ODBC)** menü hiyerarşisi izlenebilir. Şekil 6.8'de verilen ODBC veri kaynağı yönetici penceresinde, ODBC bağlantısının adı "MS Access Database" olarak tanımlanmıştır.

Şekil 6.8

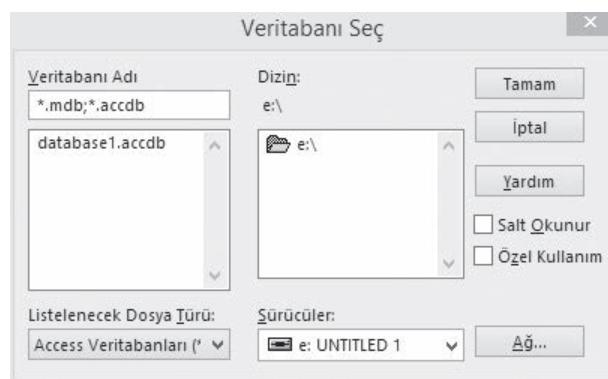


R yazılımının, veritabanından veri alabilmesi için bu bağlantı türü de kullanılabilir. Ancak, bu işlemin gerçekleştirilebilmesi için ihtiyaç duyulan **RODBC** paketinin kurulu olması gerekmektedir. Eğer kurulu değilse, *Paketler* menüsünden *Paket Kur* seçeneği seçilerek kurulur. Kurulum bittikten sonra, paketin hafızaya yüklenmesi için, yine aynı menüde bulunan *Paket Yükle* seçeneği yardımıyla veya *library(RODBC)* komutu yardımıyla **RODBC** paketi hafızaya yüklenebilir. Bu paketin, *odbcConnect()* ve *sqlFetch()* fonksiyonları yardımı ile bağlantı kurularak, veritabanındaki istenen bir veri tablosu okunabilir. *odbcConnect()* fonksiyonunun kullanımında gerekli olan parametreler *dsn*, *uid* ve *pwd* parametreleridir. Diğer parametreler için R komut satırına *help(odbcConnect)* komutunu yazarak yardım alınabilir. *dsn* parametresi sistemde kayıtlı olan veri kaynağının ismini, *uid* ve *pwd* parametreleri ise, veritabanına erişim için gerekli ise kullanıcı adı ve şifre değerlerini tanımlamaktadır. Bu parametrelerden *dsn*, Şekil 6.8'de de gösterilen "MS Access Database" değeri ile kullanılırken, diğer iki parametre ise, örnek veritabanına erişimi kullananıcı adı ve şifreye bağlı olmadığı için kullanılmayacaktır. Komut verildiğinde, R yazılımı, bağlantı kurulacak veritabanın dosyasının seçilmesini istediği, Şekil 6.9'da da gösterilen pencereyi açarak işlemin devam etmesini bekleyecektir. İzleyen komut satırı bahsedilen bu işlemleri gerçekleştirmektedir.

```
>library(RODBC)
>baglanti<-odbcConnect("MS Access Database")
```

Sekil 6.9

R Yazılımına Bağlantı
Yapılacak Veri
Tabanının Seçilmesi



Veri bağlantısı sağlandıktan sonra, `sqlFetch()` fonksiyonu yardımıyla istenen tabloya erişim sağlanır. `sqlFetch()` fonksiyonunun kullanımında gerekli olan parametreler `channel` ve `sqtable` parametreleridir. Diğer parametreler için R komut satırına `help(sqlFetch)` komutunu yazarak yardım alınabilir. `channel` parametresi, daha önce `odbcConnect()` fonksiyonu ile yapılan bağlantı, `sqtable` parametresi ise yapılan bağlantı üzerinden erişebilir olan tablonun ismini belirlemektedir. `channel` parametresi bir önceki komut satırında bağlantının nasıl yapıldığını gösteren “baglanti” değişkeni ile, `sqtable` parametresi ise Şekil 6.7’de gösterilen “Örnek Grafik Verisi” değerleri ile kullanılacaktır. İzleyen komut satırı, veri bağlantısı sağlanmış veritabanındaki “Örnek Grafik Verisi” isimli tablodan veri aktarımı işlemini gerçekleştirir.

```
>veri2<-sqlFetch(baglanti,"Örnek Grafik Verisi")
>veri2[1:5]
MÜŞTERİ BORÇ GELİR STATÜ RISK
1    1 yüksek yüksek işveren kötü
2    2 yüksek yüksek ücretli kötü
3    3 yüksek düşük ücretli kötü
4    4 düşük düşük ücretli iyi
5    5 düşük düşük işveren kötü
>close(baglanti)
```

Veri aktarımı tamamlandıktan sonra, başka veri aktarımı söz konusu değilse, sonlandırılacak bağlantının ismi verilmek şartı ile `close()` veya yapılmış tüm bağlantıları kapatmak için `closeAll()` fonksiyonu ile bağlantı(lar) sonlandırılabilir.

INTERNET



RODBC paketi hakkında detaylı bilgiye

<https://cran.r-project.org/web/packages/RODBC/index.html> linki aracılığı ile ulaşılabilir.

Sınıflandırma ve Regresyon Ağaçlarının rpart Paketi ile Çözümü

R ile sınıflandırma ve regresyon ağaçları oluşturabilmek için `rpart` paketinin R'de kurulu olması gerekmektedir. Eğer kurulu değilse, `Paketler` menüsünden `Paket Kur` seçeneği seçilerek kurulur. Kurulum bittikten sonra, paketin hafızaya yüklenmesi için, yine aynı menüde bulunan `Paket Yükle` seçeneği yardımıyla veya `library(rpart)` komutu yardımıyla `rpart` paketi hafızaya yüklenebilir.

rpart paketi hakkında detaylı bilgiye <https://cran.r-project.org/web/packages/rpart/> linki aracılığı ile ulaşılabilir.



INTERNET

rpart paketi içerisinde yer alan **rpart()** fonksiyonunda kullanılan parametreler sırasıyla, hedef niteliği de içeren herhangi bir etkileşimin söz konusu olmadığı ilişki formülünü ifade eden **formula**, formüldeki değişkenlerin çevrilebilmesi için gerekli olan veri yiğinını içeren değişkeni ifade eden **data** ve karar ağacının oluşturulma amacını ifade eden **method** parametreleridir. Bu fonksiyon ile ilgili yardım için **help(rpart)** komutundan yararlanılabilir. Daha önce de incelediğimiz örnek veritabanı için ilgili parametreler, sırasıyla **formula = RISK ~ BORÇ + GELİR + STATÜ**, **data=veri[,2:5]** ve sınıflandırma ağacı oluşturmak istendiğinden **method="class"** şeklinde kullanılacaktır. Burada veri değişkeninin ilk sütunu hariç tutulmuştur, çünkü ilk sütunda müşteri numarası yer almaktadır. İzleyen komut satırları sırasıyla **rpart** paketini erişilebilir hâle getirmekte ve **rpart()** fonksiyonu ile sınıflandırma ağacı modeli elde edilmektedir.

```
>library(rpart)
>agac<-rpart(formula=RISK~BORÇ+GELİR+STATÜ,data=veri[,2:5],method="class")
>agac
n= 45

node), split, n, loss, yval, (yprob)
  * denotes terminal node

1) root 45 22 iyi (0.5111111 0.4888889)
  2) STATÜ=ücretli 28 12 iyi (0.5714286 0.4285714)
    4) GELİR=düşük 13 4 iyi (0.6923077 0.3076923) *
    5) GELİR=yüksek 15 7 kötü (0.4666667 0.5333333) *
  3) STATÜ=işveren 17 7 kötü (0.4117647 0.5882353) *
```

Komut diziliminin en son satırında yer alan agac değişkeni bize elde edilen sonuçları gösterecektir. Sonuçlara göre, sırasıyla düğüm numarası (*node*), düğümü yaratan ayırcı niteliğin tanımı (*split*), düğümdeki kayıt sayısı (*n*), düğümdeki kayıp kayıt sayısı (*loss*), düğüm için yapılan sınıf kestirimi (*yval*) ve ilgili düğümde yer alan kayıtların sınıflayıcı nitelik değerlerinin olasılıkları (*yprob*) yer almaktadır. “*” ile işaretlenen düğümler yaprak düğümleri ifade etmektedir.

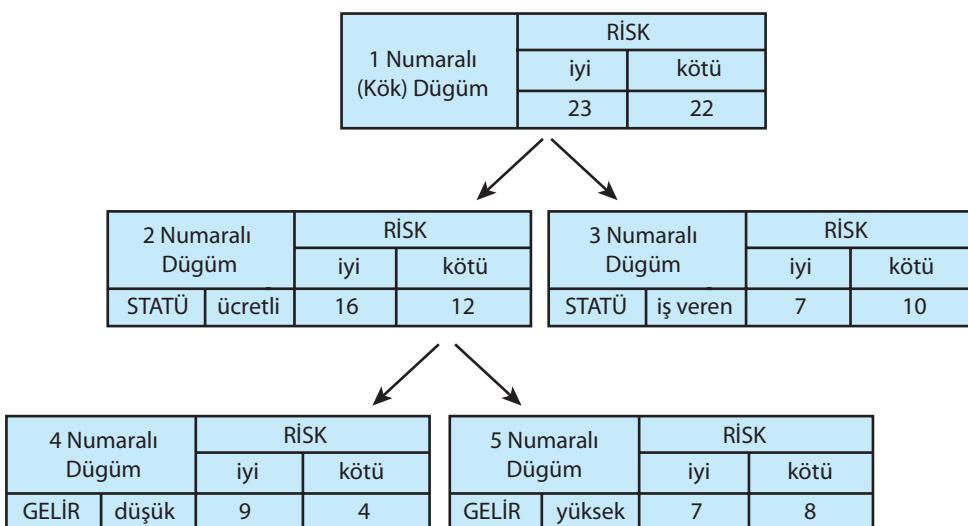
Elde edilen karar ağacında 1 numaralı kök düğümde (*root*) 45 kayıt yer almaktadır. Kök düğümde 22 kayıt kötü sınıfındadır, dolayısı ile kayıp olarak ortaya çıkmıştır. Kök düğümün risk sınıfı iyi olarak belirlenmiştir, çünkü düğümde baskın nitelik değeri, 23 kayıtta gözlemlenen iyi değeridir. Kök düğümünü en iyi şekilde ikiye ayıran nitelik olarak statü niteliği seçilmiştir. Sol dal ücretli statüsüne sahip kayıtları 2 numaralı düğüme, sağ dal ise işveren statüsüne sahip kayıtları 3 numaralı düğüme ayırmıştır. 2 numaralı düğüm 28 kayıt içermektedir ve 12 kayıtın risk sınıfı kötü olduğundan kayıp olarak ortaya çıkmıştır. 2 numaralı düğümün risk sınıfı iyi olarak belirlenmiştir, çünkü düğümde baskın nitelik değeri, 16 kayıtta gözlemlenen iyi değeridir. 3 numaralı düğüm 17 kayıt içermektedir ve 7 kayıtın risk sınıfı iyi olduğundan kayıp olarak ortaya çıkmıştır. 3 numaralı düğümün risk sınıfı kötü olarak belirlenmiştir. Burada, daha fazla ayrılamadığından 3 numaralı düğüm yaprak düğümdür ve “*” ile işaretlenmiştir. 2 numaralı düğüm ise bir iç düğümdür. 2 numaralı düğümü en iyi ayıran nitelik ise gelir niteliği olarak seçilmiştir ve sol dal ile düğümü düşük gelire sahip 13 kayıt 4 numaralı düğüme, sağ dal ile yüksek gelire sahip 15 kayıt 5 numaralı düğüme ayrılmıştır. 4 numaralı düğümdeki 13 kayıtta 4

tanesinin risk sınıfı kötü olduğu için kayıp olarak ortaya çıkmıştır ve düğümün risk sınıfı iyi olarak belirlenmiştir. Bu düğüm daha fazla ayrılamadığından bir yaprak düğümdür ve “*” ile işaretlenmiştir. Hatırlanacağı gibi 5 numaralı düğümde 15 kayıt bulunmaktadır, bu kayıtların 7 tanesinin risk sınıfı iyi olduğundan kayıp olarak ortaya çıkmıştır. 5 numaralı düğümün risk sınıfı kötü olarak belirlenmiştir ve daha fazla ayrılamadığı için yaprak düğümdür. Sınıflandırma ağacının sınıfladığı kayıtların sınıflayıcı niteliğe göre dağılımları Şekil 6.10'daki gibi özetlenebilir.

Yaprak düğümler incelendiğinde, kayıtların dağılımının, 5 numaralı düğüm hariç basık bir sınıf niteliğine sahip oldukları görülmektedir. Sınıflandırmanın amacı, bu dağılımların mümkün olduğunda ayırcı şekilde ortaya çıkmasını sağlamaktır. Başka bir ifadeyle, 4 numaralı düğümde, iyi kredi riskine sahip kayıt sayısı 9 iken, kötü kredi riskine sahip kayıt sayısı 4'tür; 3 numaralı düğümde de benzer bir durum söz konusudur. Ancak 5 numaralı yaprak düğümde, risk kategorileri neredeyse eşit sayıdadır. Ayırcı niteliğin önemi burada ortaya çıkmaktadır. Ayırcı nitelik, düğümde yer alan kayıtları mümkün olan en saf şekilde ayırabilmelidir.

Şekil 6.10

Elde Edilen Sınıflandırma Ağacındaki Kayıtların Sınıflayıcı Niteliğe Göre Dağılımı



R ile elde edilen sınıflandırma ağacı modeli biraz daha detaylı incelenmek istenirse **summary()** fonksiyonundan yararlanılır. İzleyen komut satırı banka müşteri veritabanı için elde edilen sınıflandırma ağacı modelini içeren *agac* değişkeninin içeriğini **summary(agac)** fonksiyonu yardımıyla görüntülemektedir.

```

>summary(agac)
Call:
rpart(formula = RİSK ~ BORÇ + GELİR + STATÜ, data = deneme[,2:5], method =
"class")
n= 45
  
```

```

CP nsplitt rel error xerror xstd
1 0.13636364 0 1.0000000 1.409091 0.1411614
2 0.04545455 1 0.8636364 1.409091 0.1411614
3 0.01000000 2 0.8181818 1.227273 0.1493789

Variable importance
GELİR STATÜ BORÇ
54      41      4

Node number 1: 45 observations, complexity param=0.1363636
predicted class=iyi expected loss=0.4888889 P(node) =1
    class counts: 23 22
    probabilities: 0.511 0.489
left son=2 (28 obs) right son=3 (17 obs)
Primary splits:
    STATÜ splits as RL, improve=0.5393091, (0 missing)
    BORÇ splits as RL, improve=0.1185185, (0 missing)
    GELİR splits as RL, improve=0.1015371, (0 missing)

Node number 2: 28 observations, complexity param=0.04545455
predicted class=iyi expected loss=0.4285714 P(node) =0.6222222
    class counts: 16 12
    probabilities: 0.571 0.429
left son=4 (13 obs) right son=5 (15 obs)
Primary splits:
    GELİR splits as LR, improve=0.7091575, (0 missing)
    BORÇ splits as RL, improve=0.1527884, (0 missing)
Surrogate splits:
    BORÇ splits as RL, agree=0.571, adj=0.077, (0 split)

Node number 3: 17 observations
predicted class=kötü expected loss=0.4117647 P(node) =0.3777778
    class counts: 7 10
    probabilities: 0.412 0.588

Node number 4: 13 observations
predicted class=iyi expected loss=0.3076923 P(node) =0.2888889
    class counts: 9 4
    probabilities: 0.692 0.308

Node number 5: 15 observations
predicted class=kötü expected loss=0.4666667 P(node) =0.3333333
    class counts: 7 8
    probabilities: 0.467 0.533

```

summary(agac) fonksiyonu yardımıyla detaylı olarak gösterilen sınıflandırma ağıacı modeline göre, fonksiyonun ürettiği ilk tablo maliyet karmaşıklık parametrelerini içerir ve sınıflandırma ağıacı için budama işlemi için gerekli görülmüşsa kullanılır. İlgili tabloda, her bir bölünme sayısına (*nsplits*), elde edilebilecek en küçük ağaçtan, en büyük ağaç'a karşılık gelen karmaşıklık parametresine (*complexity parameter*), bağıl hataya (*relative error*), çapraz doğrulama hatasına (*cross validation error*) ve çapraz doğrulama standart hatasına (*cross validation standard deviation*) yer verilmektedir. En uygun bölünme sayısını belirlemek için 1 standart hata kuralı kullanılabilir. Bu kuralda, en küçük çapraz doğrulama hatası

(*xerror*) ile ilgili standart hata (*xstd*) toplanır, bu toplamdan faydalananlarak en küçük çapraz doğrulama hatasına (*xerror*) sahip bölünme ilgili toplamdan küçük olan ve en az ayırmaya sahip olan küçük çapraz doğrulama hatasına en az ayırma sayısına sahip olan ağaç seçilir. Budama işlemi, seçilen bölünmeye ait karmaşıklık parametresi (*cp*) ile yapılacaktır.

Maliyet karmaşıklık tablosunu, elde edilen düğümlerin detaylı bilgileri takip etmektedir. 1 numaralı düğüm, diğer adıyla kök düğüm 45 gözlem içermektedir. Bu düğümün karmaşıklık parametresi 0,1363636 olarak elde edilmiştir. Sınıf sayıları ise iyi kredi riskine sahip 23, kötü kredi riskine sahip 22 banka müşterisi şeklindedir. Bu düğümde, kayıp oranı 22 değerinin 45 değerine oranı olan 0,489 olarak ortaya çıkmaktadır. Bu düğümdeki banka müşterilerinin ücretli statüsünde bulunan 28 tanesi sol düşüme, iş veren statüsünde bulunan 17 tanesi ise sağ düşüme ayrılmıştır. Bu ayırma 0,5393091 iyileşme (*improve*) değerine sahip olan gelir niteliği ile gerçekleştirilmıştır. Bu iyileşme değeri, düğümdeki birim sayısı (*n*) ile ayırma kriterindeki değişim, diğer bir ifadeyle kazanç ile çarpılması sonucunda elde edilmiştir. Bu değerin matematiksel büyülüğu çok önemli olmamakla birlikte göreceli büyülükleri ayırcı nitelik olarak seçilecek olan değişkenlerin ayırmada faydalalarının karşılaşmalıdır bir göstergesi konumundadır. Kök düğüm için ayırma işlemi statü niteliği yardımı ile yürütüldüğünde ayırma işlemine sağlanacak katkı miktarını belirlemek için izleyen eşitlik kullanılır.

$$\text{Improve} = 45 \cdot (Gini_{\text{Risk}} - Gini_{\text{Statü}})$$

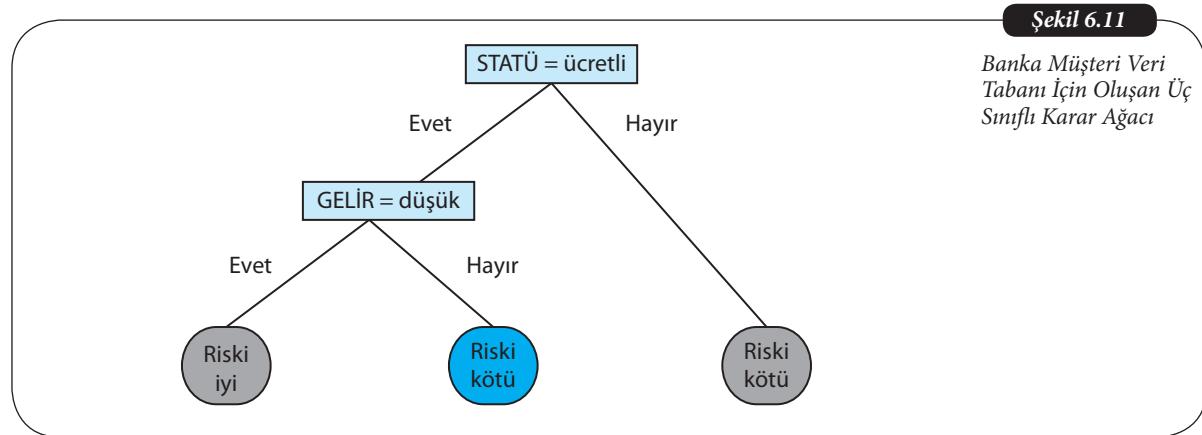
2 numaralı düğümde ise, kök düğümden sol tarafa ayrılan 28 banka müşterisi bulunmaktadır. 28 banka müşterisinin 16 tanesinin kredi riski iyi, 12 tanesinin kredi riski kötü olarak ortaya çıkmaktadır. Bu düğümde, kayıp oranı 12 değerinin 28 değerine oranı olan 0,428 olarak ortaya çıkmaktadır. Ek olarak bu düğümde ayırma yapabilecek nitelikler gelir ve borç nitelikleridir. Bu düğümü ayıran nitelik ise, 0,7091575 iyileşme değerine sahip olan gelir niteliği olacaktır. İyileşme değeri izleyen eşitlik yardımıyla hesaplanabilir.

$$\text{Improve} = 28 \cdot (Gini_{\text{Risk}} - Gini_{\text{Gelir}})$$

Gelir niteliği, 2 numaralı düğümü gelir düzeyi düşük olan 13 banka müşterisi sol tarafta, gelir düzeyi yüksek olan 15 banka müşterisi ise sağ tarafta yer alacak biçimde ayıracaktır. Borç niteliği ise, ayrıca vekil (*surrogate*) ayırcı olarak belirlenmiştir. Bu ünite kapsamında vekil ayırcı kavramı ele alınmamıştır; Ancak bir düğümdeki en iyi ayırmayı yapan niteliğe en yakın ayırmayı yapabilecek nitelik vekil ayırcı olarak tanımlanabilir. Burada amaç, sınıflandırma ağacı modeli oluşturulduktan sonra, uygulanması esnasında zaman içinde asıl ayırcı niteliğe sahip banka müşterileri ile karşılaşılmaması durumunda, asıl ayırcı niteliğin yerini alabilecek bir ayırcı niteliğin belirlenmesidir. Bu aşamada bir konuyu hatırlamakta fayda vardır. 2 numaralı düğümü ayıracak olan niteliğin belirlenmesinde önemli olan değer, iyileşme değeri değildir. Ünitenin önceki bölümünde, kök düğümü en iyi ayıran niteliğin belirlenebilmesi için oluşturulan ve Tablo 6.9'da verilen nitelikler için kontenjans tablosunun benzeri 2 numaralı düğüm için de oluşturulmalıdır. Bu kontenjans tablosu yardımıyla da 2 numaralı düğümü en iyi ayıracak olan niteliği belirleyebilmek için, gelir ve borç niteliklerine ilişkin Gini indekslerinin hesaplanması gerekmektedir. İyileşme değeri matematiksel büyülüğu çok önemli olmamakla birlikte, göreceli büyülükleri ayırcı nitelik olarak seçilecek olan değişkenlerin ayırmadaki faydalalarının karşılaşmalıdır bir göstergesi konumunda olan bir değerdir.

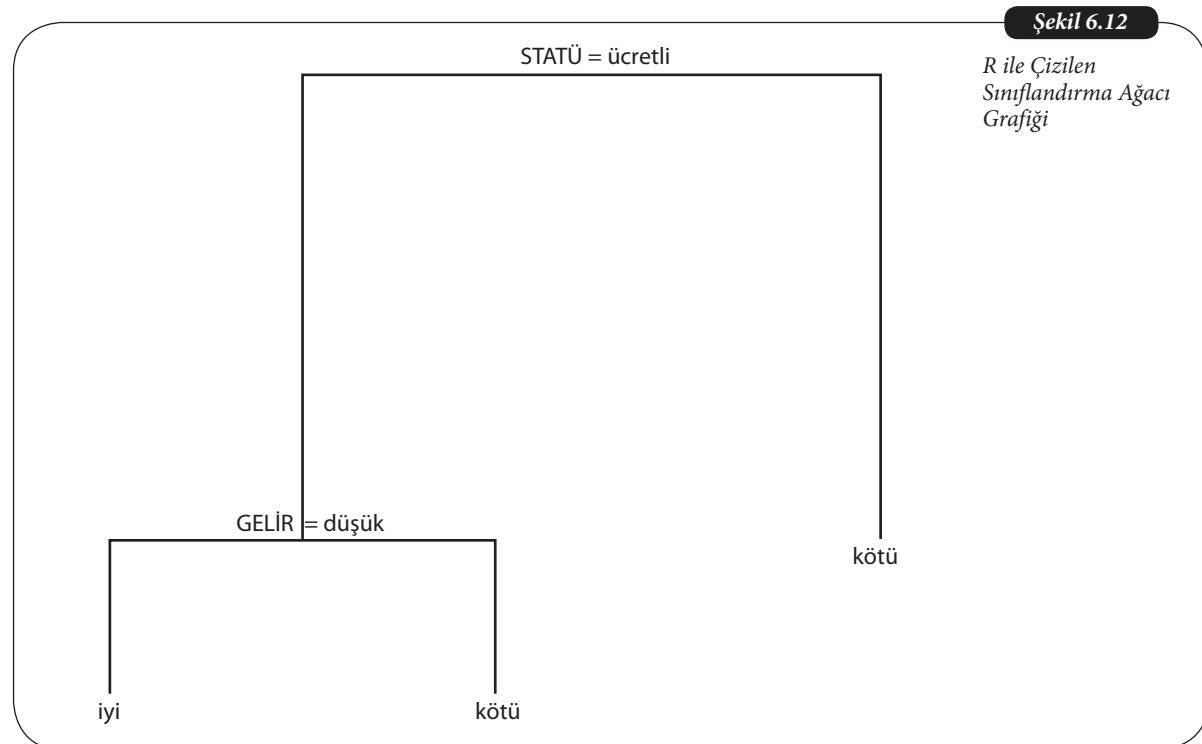
Oluşan sınıflandırma ağacı modeline göre, problemde üç adet sınıf bulunmaktadır. Banka için bu üç sınıfın ilki, ücretli ve geliri düşük müşterilerini kredi riski bakımından iyi, ikinci sınıfı ücretli ve geliri yüksek müşterileri ve üçüncü sınıf ise işveren olan müşterilerini

kredi riski bakımından kötü olarak sınıflayacaktır. İlgili sınıflandırma işlemi için karar ağacı herhangi bir grafik çizim programından yararlanılarak çizilebilir. Şu ana kadar elde ettiğimiz sonuçlardan faydalananarak, probleme ilişkin karar ağacı Şekil 6.11'de gösterilmiştir.



R yazılımı ile Şekil 6.11'de araştırmacı tarafından çizilen karar ağacı ***plot()*** fonksiyonu yardımıyla da çizilebilir. ***text()*** fonksiyonu ise düğüm ve ayırcı nitelik bilgilerini grafiğe ekleyen fonksiyondur. İzleyen komut satırları sırasıyla elde edilen sınıflandırma ağacının grafiğinin çizilmesini ve ayıra kriterlerinin grafik üzerinde gösterilmesini sağlar. ***plot()*** fonksiyonu yeni bir pencerede istenen karar ağacını çizecektir. Ancak Şekil 6.12'den de görüldüğü gibi oluşan karar ağacı Şekil 6.11 kadar görsel değildir. Bu durumda, grafigin sunumunun nasıl yapılacağı araştırmacının kendi kararına bağlıdır.

```
>plot(agac)
>text(agac)
```



Şekil 6.12'de görülen ve R yazılımı ile oluşturulan karar ağacının Şekil 6.11'deki çizebilmesi ve karar ağacının bazı ekstra bilgiler içerebilmesi için **rpart.plot** paketi kullanılabilir. Bu pakette yer alan **prp()** fonksiyonu bu amaç için kullanılabilir. **prp()** fonksiyonun kullanilan parametreleri grafiği çizilecek olan **rpart** fonksiyonu sonuçlarını içeren **x** ve düğümlerdeki ekstra gösterilecek bilgiyi belirleyen **extra** parametreleridir. **prp()** fonksiyonun ilk parametresi olan **x**, grafiği çizilecek olan sınıflandırma ağacı modelini ifade eder. Hatırlanacağı gibi, banka müşterileri veritabanından **rpart()** fonksiyonu yardımıyla elde edilen sınıflandırma ağacını içeren "agac" değişkeni **x** parametresi ile tanımlanacaktır. extra parametresi ise "4" olarak tanımlandığında kayıtların sınıf olasılıkları gösterilecektir. Benzer şekilde, kayıtların sınıfındaki yüzdesini ifade etmek için extra parametresinin değeri "100", her ikisinin birden görülmesi isteniyorsa extra parametresinin değeri "104" olarak kullanılır. **faclen** parametresi metin çıktılarının uzunluğunu düzenlemektedir ve "0" değeri, metin çıktıların kısaltılmayacağını belirtmektedir.

Diğer parametreler için R komut satırına **help(prp)** komutunu yazarak yardım alınabilir. İzleyen komut satırları sırasıyla **rpart.plot** paketini erişilebilir hâle getirmekte ve **prp()** fonksiyonu ile sınıflandırma ağacı modelinin çizdirilmesini sağlamaktadır. Şekil 6.13 izleyen komut satırlarının çalıştırılması sonucunda elde edilmiştir.

```
>library(rpart.plot)
>prp(agac,extra=104,faclen=0)
```

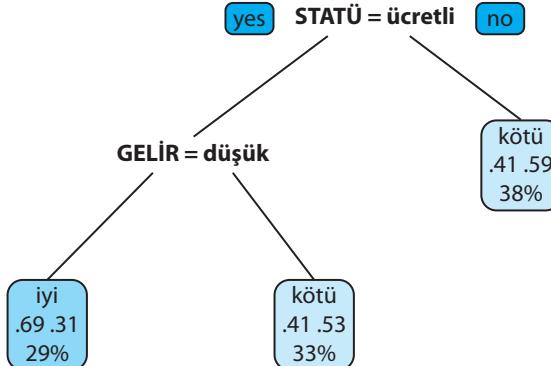
İNTERNET



rpart.plot paketi hakkında detaylı bilgiye <https://cran.r-project.org/web/packages/rpart.plot/index.html> linki aracılığı ile ulaşılabilir.

Şekil 6.13

prp Fonksiyonu ile
Sınıflandırma Ağacı
Grafiği



R yazılımında, karar ağacının grafiksel gösterimi için kullanılabilen diğer bir fonksiyon ise **rattle** paketi içinde yer alan **fancyRpartPlot()** fonksiyonudur. **fancyRpartPlot()** fonksiyonun kullanilan parametresi **model**'dır. Bu parametre, çizilecek olan karar ağacı modelini ifade eder ve daha önce elde edilen sınıflandırma ağacı modelini içeren "agac" değişkeni burada kullanılabilir. Fonksiyonun çalıştırılması sonucunda elde edilecek sınıflandırma ağacı Şekil 6.14'deki gibi olacaktır. Diğer parametreler için R komut satırına **help(fancyRpartPlot)** komutunu yazarak yardım alınabilir. İzleyen komut satırları sırasıyla **rattle** paketini erişilebilir hâle getirmekte ve **fancyRpartPlot()** fonksiyonu ile sınıflandırma ağacının çizdirilmesini sağlamaktadır.

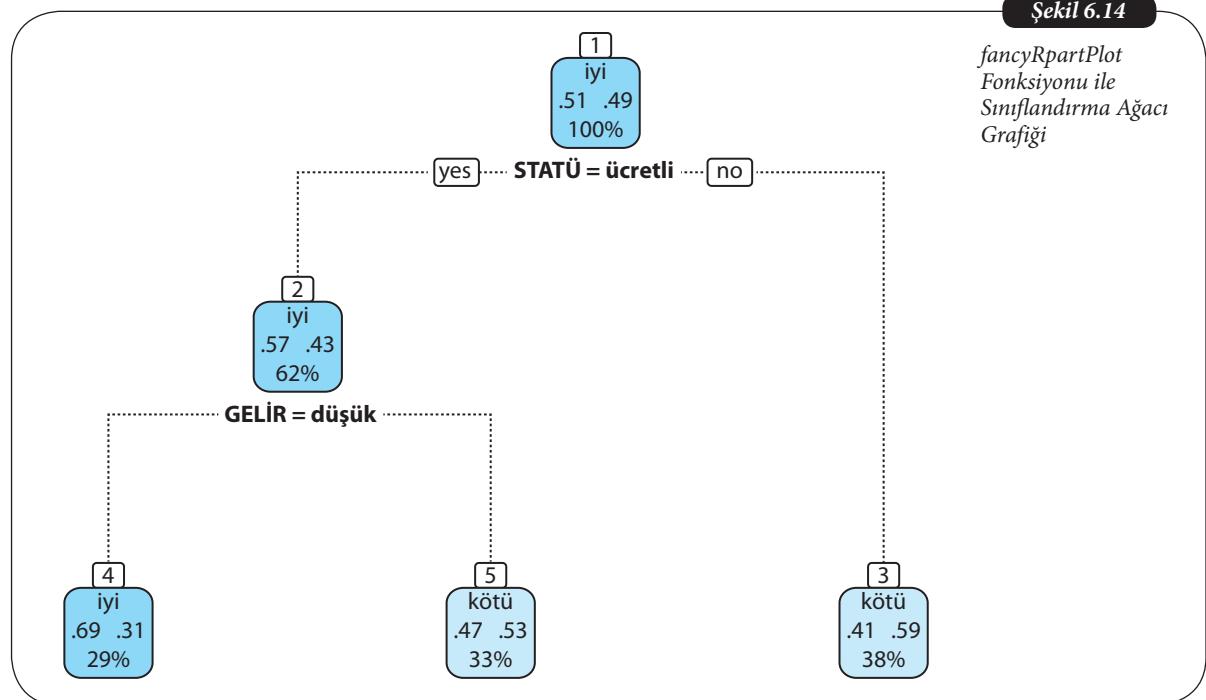
```
>library(rattle)
>fancyRpartPlot(agac)
```

rattle paketi hakkında detaylı bilgiye <https://cran.r-project.org/web/packages/rattle/index.html> linki aracılığı ile ulaşılabilir.



Sekil 6.14

fancyRpartPlot Fonksiyonu ile Siniflandırma Ağacı Grafiği



Tablo 6.1de verilen banka müşterileri veritabanı gelir niteliği nitelikle ölçülmüş bir değişkendir. Gelir değişkeni nicel nitelikle ölçülmüş bir değişken olduğunda ortaya çıkacak değişikliği görebilmek için; R yazılımı yardımıyla sınıflandırma ağacını tekrar oluşturunuz. Uygulama kolaylığı sağlama bakımından **Tablo 6.11**de gelir niteliği için Türk Lirası cinsinden örnek değerler verilmiştir.



Tablo 6.11
Gelir Niteliği İçin Nicel
Değerler (Bin TL)

Özet



Karar verme, sınıflandırma ve kestirim kavramlarını açıklamak

Karar verme karşılaşılan birden fazla seçenek içерisinden, içgüdüsel olarak veya ihtiyaç duyulan gerekşimlere göre seçim yapma işlemidir. Karar verme bir birey kadar, bir işletme için de söz konusudur. Günümüzün hızlı yaşam şartları, karşılaşılan seçeneklerin sayısını artırdığı gibi karar verme işleminin de hızlı bir şekilde yerine getirilmesini zorunlu hale getirmiştir ve karar sonuçlarını iyi sonuçlar olarak nitelendirilmesini gerektirmiştir. Bu hızlı yaşam şartları içinde doğru seçimlerin yapılmasında yardımcı olması için geliştirilmiş tekniklerden bir tanesi karar ağaçları olarak adlandırılır. Karar probleminin zaman içerisinde doğuracağı sonuçlardan etkilenen sorumlu kişiye veya kuruma olan karar verici için amaç, karar sürecinde önceden saptanan ve karar verici için belirgin özelliği olumlu olan sonuca ulaşmak olacaktır. Karar verme sürecinde, seçeneklerin, alınacak kararı etkileyen etmenlerin çokluğu ve hızlı karar verme gerekliliğinin getirdiği karmaşıklık karar vericinin vereceği kararlarda negatif bir etkiye sahip olabilmektedir. Karar ağaçları, karar probleminin karmaşıklığını, karar verme probleminde ortaya çıkabilecek tüm durumları ve karşılaşabileceği tüm senaryoları bir arada göstererek hafifleten ve uzman olmayan kişiler tarafından da kolaylıkla yorumlanabilir bir grafiksel yardımcı araçtır. Karar vericinin karşılaştığı problemlerden bir tanesi ise sınıflandırma problemidir. Sınıflandırma bir kayıtı, girdi olarak nitelik değerlerinden oluşan örnek kayıt yiğini ve karşılık gelen bir sınıf verilmesi şartıyla, önceden tanımlanmış çeşitli sınıflardan birine atayan bir modelin uydurulması işlemi olarak tanımlanabilir. Veri madenciliği metodolojisinde, diskriminant analizi gibi matematiksel işlemler yardımla istatistiksel olarak veya karar ağaçları gibi evet-hayır şeklinde değerlendirilen ifadeler ve karşılaştırma işlemleri yardımıyla mantıksal yaklaşımı sahip metotlar ile ele alınabilmektedir.



Karar ağaçlarını tanımlamak

Karar ağaçları, düğüm ve dal bileşenlerinden oluşan ve ağaca benzer bir yapıya sahip olan grafiks bir metottur. Bu yapıda, sınıflandırmayı sağlayan her bir nitelik bir **düğüm** tarafından temsil edilir ve niteliğin test edilmesini sağlar. Bir düğümden ayrılan **dallar** ise o düğümündeki testin tüm olası sonuçlarının her birine karşılık gelmektedir. İlk düğüm **kök düğüm** olarak adlandırılır ve karar ağıacı bu düğümden diğer düğümlere dallanarak büyümeye başlar. Son düğüm **yaprak düğüm**, diğer düğümler ise **İç düğüm** olarak adlandırılır. Yaprak düğümlerin her biri bir sınıfı temsil eder. Böylece, basit ama her zaman en basit olmayıpablek bir karar ağıacı oluşturulacaktır. Karar ağıacı, D veri tabanı, her $t_i \{A_i\}$ niteliklerinden oluşmuş ve C sınıf kümesi verilmiş iken, her bir düğüm A_i niteliği ile etiketlendirilmiş, düğümden ayrılan her bir dal ilgili düğüm ile ilişkili niteliğe uygulanabilen sorunun yanıtlarıyla etiketlenmiş ve her bir yaprak düğüm C_i sınıfıyla etiketlenmiş bir ağaçtır. Karar ağıacı oluşturulduktan sonra, her bir kayıt bu karar ağıacının kök düğümden başlayarak, geçtiği her düğümdeki sorunun yönlendirmesine göre bir yaprak düşüme ulaşarak, sınıflandırma yapılmış olur. Ağaç oluşturulurken kök ve iç düğümlerde hangi niteliklerin yer alacağı bulunmalıdır. Bu nitelik ayırmayı işlemini en iyi şekilde gerçekleştiren nitelik olacaktır ve ayırmayı kriteri olarak adlandırılır. Ayırıcı nitelik belirlendikten sonra, kök düğümün temsil ettiği niteliğin test sonuçlarının her biri için dallar oluşturulur. Kök düğümden ayrılan dalların bağlılığı düğümlerin temsil edeceğini nitelikler de aynı kök düğümde olduğu gibi elde edilebilir devam edilir. Bu ayırmayı, başka bir ifadeyle karar ağıacının büyümesi, örneğin bir düğümdeki kayıt sayısının bölünemeyecek kadar az olması veya ağaç derinliğinin araştırmacı tarafından belirlenen bir limite ulaşması gibi bir durma kriterine ulaşılana kadar devam edecektir. Ayırmayı kriterinin belirlenebilmesi için çeşitli yöntemler geliştirilmiştir. En sık kullanılan metotlardan bazıları nitelik veri için Entropi indeksi, sınıflandırma hatası Gini indeksi ve Twoign indeksi, nicel veriler için en küçük kareler sapması yöntemidir. Ayırmayı kriteri, ayırmayı gerçekleştiren nitelik olarak belirlenmektedir. Karar ağıacı ise, bu kriter yardımla nasıl oluşturulacağına bağlıdır. Bu amaçla, daha açık bir ifade ile karar ağıacı oluşturan çeşitli yöntemler de geliştirilmiştir. Bunlardan bazılı-

rı, ID3, C4.5, CART, CHAID, QUEST, SLIQ, SPRINT ve MARS yöntemleridir. Karar ağacı oluşturulduktan sonra, bir ya da daha fazla dalın çıkartılarak, karar ağacını daha basitleştirilmesini amaçlayan, yaprak düğüm ile değiştirme işlemi olan budama işlemi gerekebilir. Bununla birlikte karar ağacının ortaya çıkardığı sınıflandırmanın ne kadar iyi olduğunu ölçülmesi anlamına gelen performansının da ölçülmesi gerekebilir. Budama süreci için geliştirilmiş yöntemlerden bazıları maliyet karmaşıklık, kötümser hata, hata-karmaşıklık, kritik değer, azaltılmış hata, en küçük-hata budama yöntemleridir. Performans ölçümü için geliştirilen yöntemlerden bazıları ise hold-out metodu, tekrarlı hold-out metodu, çapraz-doğrulama metodu ve bootstrap metodudur.



Sınıflandırma ve regresyon ağaçlarını R yazılımını kullanarak oluşturmak ve yorumlamak

Sınıflandırma ve regresyon ağaçları (CART), veri madenciliği sürecinde karşılaşılan sınıflandırma problemlerinde sık kullanılan bir metottur. İkili (binary) karar ağaçları oluşturmaları ve yaprak düğümlerde bir sınıf kestirimini yerine sayısal bir değer kestirimini içeren regresyon ağacı da oluşturabilmesi bakımından diğer karar ağacı algoritmalarından ayrılmaktadır. CART algoritmasını uygulayabilen yazılımlardan bir tanesi de R yazılımıdır. R yazılımı, CART algoritmasını *rpart* paketi yardımıyla uygulamaktadır. Bu pakette yer alan *rpart()* fonksiyonu, temel düzeyde bir sınıflandırma veya regresyon ağacını *formula*, *data* ve *method* parametrelerini kullanarak, varsayılan olarak Gini indeksi yardımıyla oluşturmaktadır. *rpart()* fonksiyonu çıktısında, sırasıyla düğüm numarasına (*node*), düğümü yaratıcı ayırcı niteliğin tanımına (*split*), düğümdeki kayıt sayısına (*n*), düğümdeki kayıp kayıt sayısına (*loss*), düğüm için yapılan sınıf kestirimine (*yval*) ve ilgili düğümde yer alan kayıtların sınıflayıcı nitelik değerlerinin olasılıklarına (*yprob*) yer vermektedir. “*” ile işaretlediği düğümler ise yaprak düğümleri ifade etmektedir. Ayrıca, *rpart* paketinde yer alan *plot()* ve *text()* fonksiyonları yardımıyla CART ağacının standart bir grafiği elde edilebilir. Özellikle grafiksel bir yaklaşım olma yönü ile öne çıkan karar ağaçlarının, burada sınıflandırma ve regresyon ağaçları ele alındıktan, daha okunaklı ve bilgi sağlayacak şekilde grafiğe dökülmesini sağlayan diğer grafik fonksiyonları ise *rpart.plot* paketinde yer alan *prp()* fonksiyonu ve *rattle* paketinde yer alan *fancyRPartPlot()* fonksiyonudur.

Kendimizi Sınayalım

- 1.** Birden fazla seçenek içerisinde seçim yapma işlemine ne ad verilir?
 - a. Karar verme
 - b. Alternatif karar
 - c. Doğal durum
 - d. Sonuç
 - e. Karar verici

- 2.** Aşağıdakilerden hangisi, Açıköğretim Fakültesi Yönetim Bilişim Sistemleri Bölümünde verilecek olan derslerin belirlenmesi kararının sonuçlarından etkilenen bir karar verici olamaz?
 - a. Rektör
 - b. Öğrenci
 - c. Bölüm başkanı
 - d. Fakülte yönetim kurulu
 - e. Ana bilim dalı başkanı

- 3.** Aşağıdakilerden hangisi karar ağacı bileşenlerinden biri **değildir**?
 - a. Kök düğüm
 - b. İç düğüm
 - c. Yaprak düğüm
 - d. Dal
 - e. Ayırıcı nitelik

- 4.** Aşağıdakilerden hangisi bir karar ağacını sonlandıran düğümdür?
 - a. Yaprak düğüm
 - b. İç düğüm
 - c. Dal
 - d. Durma kriteri
 - e. Kök düğüm

- 5.** Aşağıdakilerden hangisi bir karar ağacını başlatan düğümdür?
 - a. Yaprak düğüm
 - b. İç düğüm
 - c. Dal
 - d. Durma kriteri
 - e. Kök düğüm

- 6.** Aşağıdakilerden hangisi nitel verilerde kullanılan ayırma kriteri belirleme metotlarından biri **değildir**?
 - a. Entropi indeksi
 - b. Gini indeksi
 - c. Sınıflandırma hatası indeksi
 - d. Twoing indeksi
 - e. En Küçük Kareler Sapması yöntemi

- 7.** Aşağıdakilerden hangisi karar ağacı oluşturma algoritmalarından biri **değildir**?
 - a. ID3
 - b. C4.5
 - c. CART
 - d. Kazanç oranı
 - e. QUEST

- 8.** Aşağıdakilerden hangisi karar ağacının budanmasında kullanılan yöntemlerden biri **değildir**?
 - a. Maliyet karmaşıklık yöntemi
 - b. Kötümser hata yöntemi
 - c. Hata-karmaşıklık yöntemi
 - d. Azaltılmış hata yöntemi
 - e. Hold-out yöntemi

- 9.** Aşağıdakilerden hangisi elde edilmiş bir karar ağacının performansını ölçen yöntemlerden biridir?
 - a. Maliyet karmaşıklık yöntemi
 - b. Kötümser hata yöntemi
 - c. Hata-karmaşıklık yöntemi
 - d. Kritik değer yöntemi
 - e. Çapraz-doğrulama yöntemi

- 10.** Bir karar ağacı düğümünde yer alan birimleri en iyi şekilde ayıracak niteliğin belirlenmesi amacıyla 5 nitelik için Gini indeksi hesaplanmıştır. Bu düğümü ayırmak için aşağıdaki niteliklerden hangisi seçilecektir?
 - a. $Gini_{Cinsiyet} = 0,4569$
 - b. $Gini_{Ders Çalışma Süresi} = 0,4669$
 - c. $Gini_{Çalışmaya Başlama Süresi} = 0,4700$
 - d. $Gini_{Gece Yatma Saati} = 0,4801$
 - e. $Gini_{Çalışma Odası Sıcaklık} = 0,4900$

Yaşamın İçinden

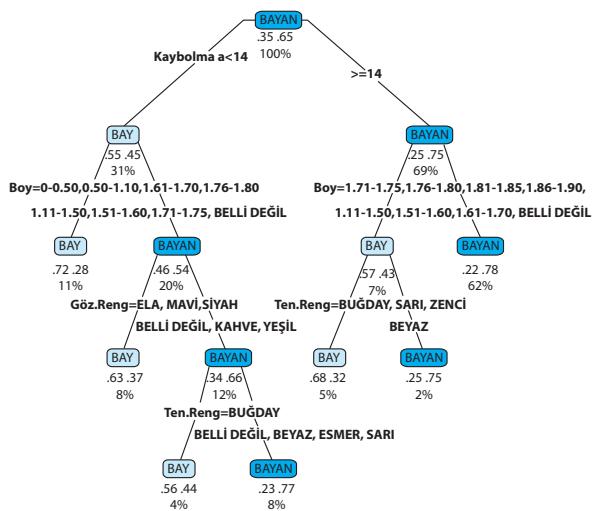
“

Bilindiği gibi dünya üzerinde olduğu kadar ülkemizde de yaşanan en önemli sorunlardan bir tanesi kayıp çocuk sorundur. Bu sorun, toplumların yaşamakta olduğu hızlı değişim sürecinden de etkilenmekte ve giderek büyüyen bir sorun halini almaktadır. Son yıllarda gözlemlenen bu artış, toplumlarda da tedirginliğe yol açmaktadır. Kayıp çocuk, “veli, vasi veya yakınları tarafından nerede olduğu veya akibeti bilinmeyen, vesayeti veya koruması altında bulunduğu kurumu izinsiz terk eden veya izinli ayrılsa bile kuruma geri dönmeyen ve hakkında polise kayıp müracaati yapılmış 18 yaşını tamamlamamış kişi” olarak tanımlanmaktadır. Bir çocuğun kaybolma nedenleri olarak macera yaşama, zengin veya ünlü olma hayali gibi çeşitli nedenlerden dolayı kendi isteğiyle kaçma; yoksulluk, aile içi şiddet, istismar, kötü ebeveynlik, çocuk işçilik, dilencilik, ideolojik veya suç işleme amaçları için kullanma; cinsel sömürü, zengin ülkelerde evlat edindirme gibi birçok neden sayılabilir. Bu önemli konuda, Başbakanlık İnsan Hakları Başkanlığı tarafından 2008 yılında “Kayıp Çocuklar Raporu” ve İçişleri Bakanlığı tarafından 2009 yılında “Kayıp Çocuk Rehberi” hazırlanmıştır. TBMM içerisinde kurulan Meclis Araştırma Komisyonu ise 2010 yılında, Kayıp Çocuk Sorunu ve İlgili Risk Faktörlerinin İncelenmesi amacıyla saha araştırması gerçekleştirmiştir. Emniyet Genel Müdürlüğü Asayiş Dairesi Başkanlığı internet sitesinde ise kayıp çocukların albümü yayınlanmaktadır. Burada, kayıp durumındaki çocukların tanımlanmasını sağlayabilecek çeşitli özellikleri verilmektedir. Bu özellikler, çocuğun doğduğu yer, kaybolduğu yer, cinsiyeti, göz rengi, saç rengi gibi nitel özellikler yanında yaşı, boyu, kilosu gibi çeşitli özelliklerdir. Bunların yanında diğer bazı özel bilgiler de kayıt altına alınabilmektedir. Bu değişkenler yardımıyla, kaybolan çocukların kaybolma sebeplerine göre sınıflandırılması ve elde edilen sınıflandırmaya göre, kaybolma vakası yaşanmadan önce önem alınarak, vaka sayısının azaltılmasına çalışılabilir.

Örneğin, İstanbul Emniyet Müdürlüğü Asayiş Dairesi Başkanlığı İnternet Sitesinden 2010 yılında kayıt altına alınmış 448 kayıp çocuğun *cinsiyeti, doğum tarihi ve kaybolma tarihi* özelliklerinden derlenmiş *kayıp yaşı, boyu, kilosu, ten rengi, göz rengi ve saç rengi* özelliklerine göre sınıflandırma ve regresyon ağacı ile sınıflandırılması mümkündür.

Bu sınıflandırma sonucunda, kayıp durumundaki çocukların 14 yaşından küçük ve büyük olmak üzere iki sinifa ayrılmaktadır. Daha sonraki sınıflandırmalar ise, sırasıyla *boy, göz rengi* ve *ten rengi* değişkenleri ile yapılmaktadır. Sınıflandırma sonunda, elde edilen sınıflardan bir tanesi *kaybolma yaşı 14'ten büyük, boyu 1.11-1.50, 1.51-1.60, 1.61-1.70 veya belli değil* olan kayıp bir çocuğun *kız çocuğu* olacağı şeklinde ortaya çıkmaktadır. 448 Kayıp çocuğun %62'si bu sınıfa dahil olmaktadır ve bu sınıfta yer alan bir çocuğun kız çocuğu olma olasılığı 0.78'dir. Sınıflandırma ve Regresyon Ağacı Şekil 6.15'deki gibidir.

Şekil 6.15: Kayıp Çocuklar İçin Sınıflandırma ve Regresyon Ağacı



Kaynak: Levent TERLEMEZ (2015). *Sınıflandırma Ağacı Yaklaşımının R ile Çözümlenmesi: Kayıp Çocuk Profil Örneği*, İktisadi Yenilik Dergisi, Siirt Üniversitesi İktisadi ve İdari Bilimler Fakültesi, Cilt:3, Sayı:1.

”

Kendimizi Sınavalım Yanıt Anahtarı

1. a Yanınız yanlış ise “Giriş” konusunu yeniden gözden geçiriniz.
2. b Yanınız yanlış ise “Giriş” konusunu yeniden gözden geçiriniz.
3. e Yanınız yanlış ise “Karar Ağaçları” konusunu yeniden gözden geçiriniz.
4. a Yanınız yanlış ise “Karar Ağaçları” konusunu yeniden gözden geçiriniz.
5. e Yanınız yanlış ise “Karar Ağaçları” konusunu yeniden gözden geçiriniz.
6. e Yanınız yanlış ise “Ayırma Kriteri” konusunu yeniden gözden geçiriniz.
7. d Yanınız yanlış ise “Karar Ağacı Oluşturma Algoritmaları” konusunu yeniden gözden geçiriniz.
8. e Yanınız yanlış ise “Karar Ağacı Budama Süreci ve Performansının Test Edilmesi” konusunu yeniden gözden geçiriniz.
9. e Yanınız yanlış ise “Karar Ağacı Budama Süreci ve Performansının Test Edilmesi” konusunu yeniden gözden geçiriniz.
10. a Yanınız yanlış ise “Karar Ağaçları” konusunu yeniden gözden geçiriniz.

Sıra Sizde Yanıt Anahtarı

Sıra Sizde 1

Bir karar verici olarak karar alternatifleriniz adsl bağlantı, fiber optik bağlantı ve kablosuz mobil bağlantı seçenekleridir. Bu karar alternatiflerinin üzerinde birakacağı genel memnuniyette farklı etkileri olabilir. Burada dikkat edilmesi gereken nokta, her karar alternatifini etkileyebilecek ve memnuniyetin üzerinde doğrudan etki edecek faktörlerin, başka bir ifade ile doğal durumlarının bulunmasının gerekliliğidir. Vermeniz gereken kararı etkileyen faktörler, örneğin abonelik ücreti, bağlantı hızı, müşteri hizmetlerinin geri dönüş hızı, ortaya çıkabilecek arızaların giderilme hızı, temel hizmet yanında sunulan yan hizmetler örnek verilebilir. Ancak bir çok faktörün doğal durumlarının oluşturulması, bu problem için çok zor olacaktır.

Sıra Sizde 2

Tablo 6.8’de verilen, kök düğümden statü ayırcı niteliğinin ücretli kategorisi ile ayrılan kayıt grubunu en iyi ayıracak niteliği, Gini indeksini kullanarak belirlemek için borç ve gelir nitelikleri için Gini indeksinin hesaplanması gerekmektedir. Hesaplama için dağılım tablosu, Tablo 6.12’deki gibi elde edilir.

Tablo 6.12: Ücretli Statüsündeki Müşterilerin Borç ve Gelir Niteliklerinin Dağılım Tablosu

Risk		Borç		Gelir	
		Yüksek	Düşük	Yüksek	Düşük
		Kötü	4	8	8
		İyi	7	9	7
		Toplam	11	17	15
					13

Borç niteliğinin sol ve sağ dalları için Gini indeksleri;

$$\text{Gini}_{\text{Borç, sol}} = 1 - \sum_{i=1}^2 \left(\frac{L_i}{|T_{\text{sol}}|} \right)^2 = 1 - \left[\left(\frac{4}{11} \right)^2 + \left(\frac{7}{11} \right)^2 \right] = 0.4628$$

$$\text{Gini}_{\text{Borç, sağ}} = 1 - \sum_{i=1}^2 \left(\frac{R_i}{|T_{\text{sağ}}|} \right)^2 = 1 - \left[\left(\frac{8}{17} \right)^2 + \left(\frac{9}{17} \right)^2 \right] = 0.4983$$

Borç niteliği için Gini indeksi;

$$\begin{aligned} \text{Gini}_{\text{Borç}} &= \frac{1}{n} (|T_{\text{sol}}| \cdot \text{Gini}_{\text{sol}} + |T_{\text{sağ}}| \cdot \text{Gini}_{\text{sağ}}) \\ &= \frac{1}{28} (11 \cdot 0.4628 + 17 \cdot 0.4983) \\ &= 0.4846 \end{aligned}$$

Şeklinde elde edilir.

Gelir niteliğinin sol ve sağ dalları için Gini indeksleri;

$$\text{Gini}_{\text{Gelir, sol}} = 1 - \sum_{i=1}^2 \left(\frac{L_i}{|T_{\text{sol}}|} \right)^2 = 1 - \left[\left(\frac{8}{15} \right)^2 + \left(\frac{7}{15} \right)^2 \right] = 0.4978$$

$$\text{Gini}_{\text{Gelir, sağ}} = 1 - \sum_{i=1}^2 \left(\frac{R_i}{|T_{\text{sağ}}|} \right)^2 = 1 - \left[\left(\frac{4}{13} \right)^2 + \left(\frac{9}{13} \right)^2 \right] = 0.4260$$

Gelir niteliği için Gini indeksi;

$$\begin{aligned} \text{Gini}_{\text{Gelir}} &= \frac{1}{n} (|T_{\text{sol}}| \cdot \text{Gini}_{\text{sol}} + |T_{\text{sağ}}| \cdot \text{Gini}_{\text{sağ}}) \\ &= \frac{1}{28} (15 \cdot 0.4978 + 13 \cdot 0.4260) \\ &= 0.4645 \end{aligned}$$

Şeklinde elde edilir. Bu sonuçlara göre,

Tablo 6.13: Niteliklerin Gini İndeks Değerleri

Nitelik	Gini İndeksi
Borç	0,4843
Gelir	0,4645

Borç ve gelir için hesaplanan Gini indeksleri Tablo 6.13’deki gibi olacaktır. Elde edilen Gini indeksi değerlerine göre, en düşük Gini indeksine sahip olan gelir niteliği ücretli statüsündeki müşterilerin gruplandığı düğüm için en iyi ayırcı nitelik olarak elde edilir. Dolayısı ile bu düğümde yer alan ücretli statüsündeki müşteriler, geliri yüksek ve düşük olan müşteriler olarak ayrılacaktır.

Sıra Sizde 3

Tablo 6.1'de verilen banka müşterileri veri tabanında yer alan gelir niteliği "yüksek" ve "düşük" olmak üzere iki düzeye sahip nitel bir ölçekle ölçülmüş bir niteliktir. Bu nitelik, Tablo 6.11'de verilen nicel değerler ile Şekil 6.16'daki gibi değiştirilerek, banka müşterilerinin sınıflandırması yapılmak istendiğinde sırasıyla Şekil 6.17 ve Şekil 6.18'deki gibi bir karar (sınıflandırma) ağıacı çıktı ve grafiği elde edilir. Elde edilen karar ağıacı sınıflandırmasına göre, geliri 20 Bin Türk Lirasından az olan banka müşterileri ile geliri 20 Bin Türk Lirasından az ve borcu yüksek olan banka müşterileri kredi riski iyi olarak sınıflandırılmışlardır. Geliri 20 Bin Türk Lirasından az ve borcu düşük olan banka müşterileri ise kredi riski kötü olarak sınıflandırılmıştır. Sınıflandırma ağıacı biraz daha detaylı incelendiğinde ise geliri 20 Bin Türk Lirasından az olan banka müşterileri (2 no'lu düğüm) ile geliri 20 Bin Türk Lirasından az ve borcu düşük olan banka müşterilerinin (7 no'lu düğüm) sınıflandırması geliri 20 Bin Türk Lirasından az ve borcu yüksek olan banka müşterilerine (6 no'lu düğüm) göre daha iyi sınıflandırılmıştır. Çünkü kredi riski iyi olarak nitelendirilen banka müşterilerinin sayısı, kötü olarak nitelendirilen banka müşterilerinin sayısından daha yüksektir. Yani mümkün olan en az sayıda kredi riski kötü olarak nitelendirilen müşteri bu sınıflara, yani yaprak düğümlere dahil olmuştur.

Şekil 6.16: Nicel Ölçekle Ölçülmüş Gelir Niteliği ile Banka Müşterisi Veri Tabanı

```
> veri1[1:5,]
  MÜŞTERİ   BORÇ   GELİR   STATÜ RİSK
1      1 yüksek    38 işveren kötü
2      2 yüksek    32 Ücretli kötü
3      3 yüksek    13 Ücretli kötü
4      4 düşük     58 Ücretli iyi
5      5 düşük     22 işveren kötü
>
```

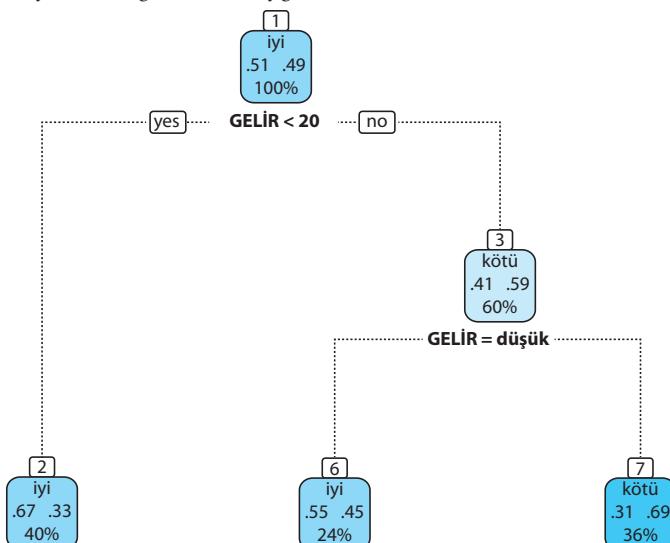
Şekil 6.17: Nicel Ölçekle Ölçülmüş Gelir Niteliği ile Banka Müşteri Veri Tabanı Kullanılarak Elde Edilen Sınıflandırma Ağacı Modeli.

```
> agac1<-rpart(formula=RİSK~BORÇ+GELİR+STATÜ,data=veri1[,2:5],method="class")
> agac1
n= 45

node), split, n, loss, yval, (yprob)
 * denotes terminal node

1) root 45 22 iyi (0.5111111 0.4888889)
  2) GELİR< 20.5 18 6 iyi (0.6666667 0.3333333) *
  3) GELİR>=20.5 27 11 kötü (0.4074074 0.5925926)
    6) BORÇ=yüksek 11 5 iyi (0.5454545 0.4545455) *
    7) BORÇ=düşük 16 5 kötü (0.3125000 0.6875000) *
>
```

Şekil 6.18: fancyRpartPlot Fonksiyonu ile Nicel Ölçekle Ölçülmüş Gelir Niteliği ile Banka Müşteri Veri Tabanı Kullanılarak Elde Edilen Sınıflandırma Ağacı Modeli Grafiği.



Yararlanılan ve Başvurulabilecek Kaynaklar

- Akpınar, H. (2014). **Data – Veri Madenciliği, Veri Analizi**, İstanbul: Papatya Yay. Eğitim.
- Dunham, H. M. (2002). **Data Mining – Introductory and Advanced Topics**, New Jersey: Prentice Hall International.
- Guidici, P. (2003). **Applied Data Mining – Statistical Methods for Business and Industry**, John Wiley and Sons Inc.
- James, G., Witten, D., Hastie, T. ve Tibshirani, R. (2013), **An Introduction to Statistical Learning with Applications in R**, Springer.
- Kantardzic, M. (2003). **Data Mining – Concepts, Models, Methods, and Algorithms**, IEEE Press.
- Maimon, O., Rokach, L. (2010). **Data Mining and Knowledge Discovery Handbook**, Springer.
- Milborrow, S. (2015). **rpart.plot: Plot ‘rpart’ Models: An Enhanced Version of ‘plot.rpart’**, R package version 1.5.3, <http://CRAN.R-project.org/package=rpart.plot>.
- Mingers, J. (1989). **An Empirical Comparison of Pruning Methods for Decision Tree Induction**, Machine Learning, 4, 227-243, Kluwer Academic Publishers.
- Özkan, Y. (2008). **Veri Madenciliği Yöntemleri**, İstanbul: Papatya Yay. Eğitim.
- R Core Team (2015). **R: A language and environment for statistical computing**, R Foundation for Statistical Computing, Vienna, Austria, <https://www.R-project.org/>.
- Ripley, B., Lapsley, M. (2015). **RODBC: ODBC Database Access**, R package version 1.3-12, <http://CRAN.R-project.org/package=RODBC>.
- Silahtaroğlu, G. (2013). **Veri Madenciliği – Kavram ve Algoritmaları**, İstanbul: Papatya Yay. Eğitim.
- Şıklar, E., Özdemir, A. (2013). **İstatistik II**, Açıköğretim Fakültesi Yayıncılık No: 1764, Eskişehir.
- Tan, P., Steinbach, M., Kumar, V. (2006). **Introduction to Data Mining**, Pearson Education, Inc.
- Therneau, T. M., Atkinson, E. J. (2015). **An Introduction to Recursive Partitioning Using the RPART Routines**, Mayo Foundation, <https://cran.r-project.org/web/packages/rpart/vignettes/longintro.pdf>.
- Therneau, T., Atkinson, B., Ripley, B. (2015). **rpart: Recursive Partitioning and Regression Trees**. R package version 4.1-10, <http://CRAN.R-project.org/package=rpart>.
- Williams, G. J. (2011). **Data Mining with Rattle and R: The Art of Excavation Data for Knowledge Discovery, Use R!**, Springer.

7

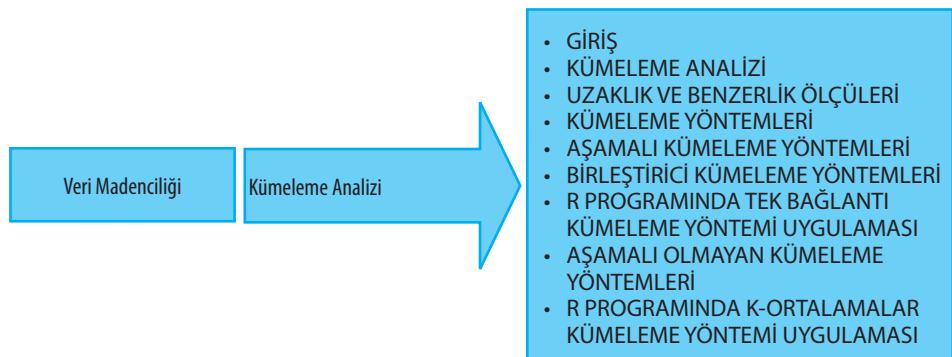
Amaçlarımız

- Bu üniteyi tamamladıktan sonra;
- 🕒 Kümeleme analizinin kullanım alanlarını tanımlayabilecek,
 - 🕒 Çok değişkenli analizler ile kümeleme analizi arasındaki ilişkileri karşılaştıracı ile,
 - 🕒 Benzerlik ya da uzaklık matrislerini belirleyebilecek,
 - 🕒 Aşamalı ve Aşamalı olmayan kümeleme yöntemlerini açıklayabilecek,
 - 🕒 R paket programında kümeleme analizini uygulayabilecek ve sonuçlarını yorumlayabilecek bilgi ve becerilerine sahip olabileceksiniz.

Anahtar Kavramlar

- Aşamalı Kümeleme Yöntemleri
- Aşamalı Olmayan Kümeleme Yöntemleri
- Birleştirici Kümeleme
- Ayırıcı Kümeleme

İçindekiler



Kümeleme Analizi

GİRİŞ

Büyük miktardaki veri yapıları üzerinde istatistiksel çözümlemeler yapmak ve veriyi çözümleyip kullanılabılır bilgiye ulaşabilmek için veri madenciliği yöntemi ortaya çıkmıştır. Veri madenciliği yöntemi bir sorgulama işlemi veya istatistik programlarıyla yapılmış bir analiz değildir. Veri madenciliği yöntemi çok büyük veri setleri ile ilgilenir. Bilişim teknolojisindeki gelişmeler dünyada gerçekleşen birçok faaliyetin elektronik olarak kayıt altına alınmasını, bu kayıtların kolayca saklanabilmesini, güncellenebilmesini ve gerektiğinde erişilebilmesini hem kolaylaştırmakta, hem de bu işlemlerin her geçen gün daha ucuza mal edilmesini sağlamaktadır. Fakat, bu büyük veri setleri yardımıyla önemli kararlar alabilmek ve anlamlı tahminler yapabilmek için konu üzerinde çalışan uzmanlarca analiz edilmesi gerekmektedir. Veri sayısının büyüğüğe bağlı olarak ve istenilen amaçlara göre bazı özel analiz algoritmaları geliştirilmiştir.

Veri Madenciliği Yöntemleri sınıflandırma, kümeleme ve birliktelik kuralları olarak ele alınabilir.

Sınıflandırma, veri madenciliğinde sıkılıkla kullanılmaktadır. Üzerinde çalışılan veritabanının bir kısmı eğitim seti olarak ele alınır ve buradan hareketle sınıflandırma kuralları oluşturulur. Bu kurallar yardımıyla yeni bir durum ortaya çıktığında nasıl karar vereileceği belirlenir. Veri madenciliği yönteminin sınıflandırma grubu içerisinde en sık kullandığı teknik “karar ağaçları”dır. Aynı zamanda lojistik regresyon, diskriminant analizi, sinir ağları ve fuzzy setleri de sıkılıkla kullanılmaktadır. İnsanlar yüzyıllardır verileri sınıflandırdıkları, kategorize ettikleri ve derecelendirdikleri için sınıflandırma, işlemi hem veri madenciliğinin temeli olarak hem de veri hazırlama aracı olarak kullanılmaktadır.

Kümeleme, veri setinde bulunan gözlemlerin ya da değişkenlerin kendi aralarındaki benzerlikleri göz önünde bulundurularak gruplandırılması işlemidir. Kümeleme yöntemlerinin çoğu veri arasındaki uzaklıkları kullanır. Uygulamada çok sayıda kümeleme yöntemi kullanılmaktadır. Bu yöntemler, değişkenler arasındaki benzerliklerden ya da farklılıklardan yararlanarak bir veri setini alt kümelere ayırmak için kullanılmaktadır. Kümeleme analizinin amacı, gruplanmamış verileri benzerliklerine göre sınıflandırmak ve araştırmacıya özetleyici bilgiler elde etmede yardımcı olmaktadır.

Birliktelik Kuralları, veri seti içerisinde yer alan kayıtların birbiriley olan ilişkilerini inceleyerek, hangi olayların eş zamanlı olarak birlikte gerçekleşebileceklerini ortaya koymaya çalışan yöntemler veri madenciliği yöntemleridir. Özellikle pazarlama alanında uygulanmaktadır (Pazar sepet analizleri). Bu yöntemler birlikte olma kurallarını belirli olasılıklarla ortaya koymaktadır.

KÜMELEME ANALİZİ

Çok değişkenli istatistik yöntemleri arasında yer alan, çok sayıda ve karmaşık yapıdaki veri setinde verileri gruplandırmak ve oluşan grupları karşılaştırmak amacıyla kullanılan kümeleme analizi, uygulaması kolay ve sonuçlarının anlaşılır olması nedeniyle sıkça başvurulan bir yöntemdir. Veri madenciliğinin bir alt türü olan bu yöntemde veriler uzaklık ve benzerliklerine göre kümelere ayrılmakta, kümeler arasındaki farklılıklar ve bu farklılıkların nedenleri üzerinde durulmaktadır. Analiz sonucunda birbirine en çok benzeyen birimler aynı kümelerde toplanmaktadır. Bu tür kümeler kendi içinde homojen farklı kümelerle de heterojen bir yapıya sahip olurlar.

Kümeleme analizinin temel amacı, hangi kümeye ait olduğu bilinmeyen bir grup verinin, sınıflandırılarak anlamlandırılmasıdır. Dolayısıyla kümeleme analizi birimleri ya da değişkenleri temel özelliklerine göre sınıflandırmak için kullanılmaktadır. Kisaca kümeleme analizinin genel amacının benzer olanları farklı olandan ayırmak olduğu ifade edilebilir.

Kümeleme analizi, çok değişkenli ham veri setindeki gözlemlerin sahip oldukları özellikler bakımından doğal grup yapılarını belirlemeyi, homojen alt gruplara ayırmayı sağlayan istatistiksel yöntemler topluluğudur. Sağlık, ziraat, biyoloji, psikoloji, sosyoloji, arkeoloji gibi gözlemlerin sınıflandırılmasına ihtiyaç duyulan pek çok bilim dalının faydalandığı uygulamalarda sıklıkla kullanılan bir yöntemdir. Bu yöntemde, veri setinde oluşturulan grupta, grup içi değişim en az, gruplar arası değişim en fazla olacağı bir yapı ortaya koymak amaçlanmaktadır. Araştırmaya ve tanımlamaya yönelik bir yöntemdir. Kümeleme analizinde temel amaç grupların elde edilmesi, yani veri yapısında var olan durumun belirlenmesidir. Kümeleme analizinden sadece gözlemleri grupplandırmak için değil, değişkenleri grupplandırmak için de yararlanılmaktadır.

Kümeleme analizi, diğer çok değişkenli analiz yöntemi olan diskriminant analizinde olduğu gibi *tahmin amaçlı kullanılmamakta* ve faktör analizinde olduğu gibi de *varsayımları bulunmamaktadır*.

Kümeleme analizi uygulama aşamasında temel varsayımlar gerektirmemektedir. Ancak analizde kullanılacak olan değişkenlerin seçimiinde hassas davranışılması gerekmektedir. Değişkenler arasındaki çoklu bağlantıya ve aşırı gözlemlere dikkat edilmelidir.

Bu yöntem yardımıyla ve mevcut paket programlar ile doğru ya da yanlış bir biçimde; her zaman kümeler oluşturulabilir. Ancak bu kümeler, kullanılan veri setine ait tek ve değişmeyen bir sonuç olmayacağındır. Çünkü çözümlemeler, tercih edilen değişkenlere ve kümeye üyeliğinin nasıl tanımlandığına bağlı olarak değişmektedir.

Kümeleme analizi genellikle dört aşamada uygulanmaktadır. Bunlar; veri matrisinin oluşturulması, benzerlik veya uzaklık matrislerinin hesaplanması, kümelemede esas alınacak yöntemlerin belirlenmesi ve elde edilen sonuçların yorumlanmasıdır.

Veri matrisinin oluşturulması, analizin *ilk* aşamasıdır. Yani öncelikle doğal sınıflamaları hakkında kesin bilgilerin bulunmadığı anakütlerlerden alınan n sayıda birimin incelenen p sayıda değişkene ilişkin gözlem sonucu değerleri elde edilir. Böylece veri matrisi oluşturulmuş olur. Araştırmada değişkenlerin seçimi aşamasında kullanılacak veri matrisine konunun kuramsal ve uygulamaya yönelik yanları dikkate alınmalıdır, konu ile ilgili güncel literatür taranmalıdır. Kümeleme analizi diğer çok değişkenli istatistiksel yöntemler gibi değişken seçimi aşamasında yapılan hataları ortadan kaldırma yeteneğine sahip değildir. Bu nedenden dolayı araştırmacıının literatürü denetleyerek ayırt edici özelliklere sahip değişkenleri analize dahil etmesi, ayırt edici özelliklere sahip olmayan değişkenleri ise çalışmadan çıkartması yararlı olacaktır. Ayrıca, veri yapısında genel eğilimin dışına çıkan ve veri yapısında bulunmaması gereken aşırı yüksek ya da düşük gözlemlerin de çalışmanın bu aşamasında belirlenmesi gereklidir.

Veri setinde bulunan değişkenlere ait ortalama ve varyanslar birbirinden çok farklı olduğunda, ortalaması veya varyansı büyük olan değişkenler, diğer değişkenleri istatistiksel analiz esnasında baskılabilir ve rollerini, analizdeki etkinliklerini azaltabilir. Bazı durumlarda değişkenlerde bulunan aşırı uç değerler kümeleme analizi üzerinde olumsuz etkilerde bulunabilir. Bu gibi durumlarda verilerin standardize edilmesi ya da belirli aralıklardaki değerlere dönüştürülmesi gereklidir.

Verilerin standardize edilmesi ve belirli aralıklardaki değerlere dönüştürülmesi için en çok kullanılan yöntemler; z skorlarına dönüştürme, $-1 \leq x \leq 1$ aralığına dönüştürme, $0 \leq x \leq 1$ aralığına dönüştürme, serideki maksimum değer 1 olacak şekilde dönüştürme, ortalama değer 1 olacak şekilde dönüştürme, standart sapma 1 olacak şekilde dönüştürme yöntemleridir.

Kümeleme analizi uygulamasının *ikinci* aşamasında benzerlik veya uzaklık matrislerinin hesaplanması işlemi yapılmaktadır. Bu matrislerin nasıl elde edileceği konusunda bilgiler kitabınızın 4. ünitesinde ayrıntılı olarak verilmiştir. Fakat eldeki veri setinden farklı benzerlik ve uzaklık matrisleri ile elde edilen kümeler arasında farklılıklar olabileceği gibi, analiz için seçilen kümeleme yöntemine göre de kümelerin değişim能力和unutulmamalıdır.

Kümeleme analizinin *üçüncü* aşamasında var olan kümeleme yöntemleri arasından hangisinin kullanılacağına karar verilir ve kümeler elde edilir.

Kümeleme analizinin *dördüncü* ve son aşaması ise sonuçların yorumlanmasıdır. Doğru kume sayısını belirlemeye; uygun yapılanma konusunda yorum yaparken bir ya da iki gözlemden oluşan kümelere şüphe ile bakılmalıdır. Kümelemede kullanılan değişkenlerin, kümeler arasında anlamlı farklarının bulunmasına dikkat edilmelidir. Ayrıca elde edilen sonuçların kuramsal açıdan geçerli olmasına ve literatürle benzerlik göstermesine özen gösterilmelidir.

Kümeleme analizi tanımlayıcı bir yöntemdir. Bu yöntemde, örneklemden elde edilen bilgiler yardımıyla topluma dönük tahminler yapmak mümkün değildir. Ayrıca araştırmacı, kümeleme analizi ile aynı veri seti için birden fazla çözüme ulaşılabilirmektedir. Kume sayısı ya da gözlemin herhangi bir kümeye atanması; seçilen uzaklık yöntemine ya da kümeleme yöntemine göre değişimleme olmaktadır. Buradan da anlaşılacağı gibi kümeleme analizi tek bir çözümü olan bir yöntem değildir. Kümeleme analizi içinde kabul edilmiş en iyi yöntem bulunmamaktadır. Sonuçların yorumlanması; birçok faktör dikkate alınabilekmekte, ilgili alana yönelik ayrıntılı bilgi ve tecrübe gereksinim duyulabilemektedir. Bunlar kümeleme analizinin olumsuz yönleri olarak ele alınmaktadır.

Kümeleme analizinde gruplandırma, benzerlik ya da farklılık ölçülerine göre yapılır. Bu ölçülerin hesaplanması gözlemler arası uzaklığa ya da koreasyona bağlı olarak iki farklı şekilde gerçekleştirilir. Literatürde en sık kullanılan ölçüler aşağıda belirtilmektedir:

Aşamalı olmayan kümeleme yöntemlerinden, K-ortalamalar yönteminde kume merkezleri neye göre belirlenir? Kume içi kareler toplamları hakkında ne söylenebilir?



SIRA SİZDE

UZAKLIK VE BENZERLİK ÖLÇÜLERİ

Kümeleme analizinde oluşturulan kümeler, birbirine yakın birimlerin ya da değişkenlerin oluşturdukları grup olarak tanımlanabilir. Kümeleme analizinde birim ya da değişkenler arasındaki uzaklıklar hesaplamak için en sık kullanılan uzaklık ölçüsü Öklid uzaklığıdır. Öklid uzaklıği iki obje arasındaki benzerliği ölçmede en yaygın kullanılan uzaklık ölçüsü olup iki obje arasına çizilecek bir doğrunun uzunluğunu temel alır. Bu uzaklık ölçüsü dışında birimler ya da değişkenler arasındaki uzaklık değerlerinden faydalananak kümelerin oluşturulmasında kullanılan farklı uzaklık ölçüler de vardır. Bunlar; Karesel Öklid uzaklı-

ğı, Pearson ve karesel Pearson uzaklığı, Manhattan (City-Blok) Uzaklığı, Korelasyon katsayısı ve korelasyon uzaklığı, Açısal uzaklık (cosine measure), Binary Öklid uzaklığı, Gamma benzerlik ölçüsü, Jaccard benzerlik ölçüsü, Minkowski uzaklığı, Mahalonobis uzaklığı, Hotelling T^2 Uzaklığı, Canberra Uzaklık ölçülerleri ile ilgili bilgiler Ünite 4'te bulunmaktadır.

KÜMELEME YÖNTEMLERİ

Kümeleme yöntemleri; uzaklık (distance), benzerlik (similarity) ya da farklılık (dissimilarity) matrisinden yararlanarak birimleri ya da değişkenleri kendi içinde homojen ve kendi arasında heterojen uygun kümelere ayırırken, kümeleri belirlemeye izledikleri yaklaşılara göre iki temel alt gruba ayrırlılar. Bunlar; Aşamalı kümeleme yöntemleri (Hierarchical Cluster Analysis Methods) ve Aşamalı olmayan kümeleme yöntemleri (Nonhierarchical Cluster Analysis Methods) olarak ele alınmaktadır. Her iki yöntemde de ortak amaç kümeler arasındaki farklılıklarını ve kümeler içi benzerliklerini yüksek düzeye çıkarmaktır. Yani, kume içi homojenlik artırılırken kümeler arası homojenlik ise azaltılmaktadır. Hangi tekninin kullanılacağı kume sayısına bağlı olmakla birlikte her iki tekniğin birlikte kullanılması çok daha yararlıdır. Böylece hem sonuçları hem de iki tekniğin hangisinin daha uygun sonuçlar verdiği karşılaştırmak mümkün olmaktadır. Bu iki yöntem dışında ileri sürülmüş bir takım kümeleme algoritmaları varsa da bu yöntemler yaygın kullanımı olan yöntemler değildir.

AŞAMALI KÜMELEME YÖNTEMLERİ

Aşamalı kümeleme yöntemleri, birimleri/değişkenleri birbirleri ile farklı aşamalarda bir araya getirerek ardışık biçimde kümeler belirlemeyi ve bu kümelere girecek elemanların hangi uzaklık (ya da benzerlik) düzeyinde kume elemanı olduğunu belirlemeye yönelik yöntemlerdir.

Aşamalı kümeleme yöntemleri, veri matrisinde bulunan birimlerin ya da değişkenlerin analizin başlangıç aşamasında kaç kume oluşturduğuna ve kume elemanlarını belirlemeye başlangıçta hangi kriterin seçildiğine göre iki temel gruba ayrılır. Bunlar; Birleştirici aşamalı kümeleme yöntemleri (Agglomerative hierarchical clustering prosedures) ve Ayırıcı aşamalı kümeleme yöntemleridir (Divisive hierarchical clustering prosedures).

Birleştirici Aşamalı Kümeleme Yöntemleri

Birleştirici (agglomerative) aşamalı kümeleme yöntemleri, başlangıçta veri setinde bulunan tüm birimlerin farklı birer kume oluşturduğu kabul edilerek analize başlanır. Veri setinde bulunan n birimi aşamalı olarak sırasıyla; n kume, n-1 kume, n-2 kume, ..., n-r kume, ..., 3 kume, 2 kume, 1 kumeye yerleştirmeyi amaçlayan bir yaklaşımdır. Bu yöntemde, her birim başlangıçta tek başına farklı birer kume olarak kabul edilir. Daha sonra birbirleri ile yüksek derecede benzerlik gösteren iki birim, bir kume oluşturur. Bir sonraki adımda bu kumeye farklı benzerlik düzeylerinde diğer birimler eklenecek birimlerin tamamı bir kumede toplanacak biçimde birbirleri ile bağlanırlar (birleştirilirler, kümelenirler).

Birleştirici aşamalı kümeleme yöntemleri, birimlerin oluşturduğu kümelerin şekillenmesinde, birbirleri ile hangi aşamada ve hangi benzerlik düzeyinde ortak özelliklere sahip kümeler oluşturduklarını göstermeleri açısından yaygın olarak kullanılan kümeleme yaklaşımıdır.

Ayırıcı Aşamalı Kümeleme Yöntemleri

Ayırıcı (divisive) aşamalı kümeleme yöntemlerinde, başlangıçta veri setinde bulunan tüm birimlerin bir kume olduğu varsayılarak analize başlanır. Diğer bir ifadeyle işlem, birleştirici aşamalı kümeleme yönteminde olan aşamaların tam tersine işler. İlk olarak tüm bi-

rimleri içeren büyük bir küme ele alınır. İzleyen aşamalarda en farklı (uzak) birimler bir-birinden ayrılarak daha küçük kümeler oluşturulur. Bu aşamalar her birim kendi başına farklı bir küme oluşturuncaya kadar devam eder. Veri setinde bulunan n birimi sırasıyla aşamalı olarak 1 küme, 2 küme, 3 küme, ..., n-r küme, n-3 küme, n-2 küme, n-1 küme, n kümeye ayırmayı amaçlayan bir yaklaşımındır.

Ayırıcı aşamalı kümeleme yöntemleri uygulamada yaygın olarak kullanılmayan bir yöntem olduğundan bu bölümde sadece birleştirici kümeleme yöntemlerine değinilecektir. Bundan böyle aşamalı kümeleme yöntemi denildiğinde birleştirici aşamalı kümeleme yöntemleri anlaşılacaktır.

Kümeleme Analizi, Diskriminant Analizinde olduğu gibi tahmin amaçlı olarak kullanılabilir mi? Ayrıca, Faktör Analizinde olduğu gibi varsayımları var mıdır?



SIRA SİZDE

2

Dendrogramlar (Ağaç Diyagramları)

Kümeleme analizinde sonuçlar dendrogram (ağaç diyagramı) adı verilen grafiksel yöntemle sunulurlar. Dendrogramlarda bağlantılar, uzaklıklar ve birimlerin bağlanma düzeyleri bir ağaç biçiminde ele alınarak şekillendirilir ve kümelenme süreci bu şekilde ayrıntılı bir biçimde özetlenir. Genellikle dendrogramlar; x ekseninde birimler ve y ekseninde de uzaklıklar olacak şekilde yapılandırılırlar.

Dendrogramlarda değişkenlerin ya da birimlerin hangi aşamada ve hangi uzaklık ya da benzerlik düzeyinde bir araya gelerek küme oluşturdukları ayrıntılı biçimde görülmektedir. Dendrogramlar şekil 7.1'de ve şekil 7.2'de görüldüğü gibi dikey çizilebileceği gibi bazı hazır istatistiksel paket programlarda yatay olarak da çizilebilmektedir.

Birleştirici ve aşamalı yöntemleri dendrogram üzerinde aşağıdaki örnek yardımıyla inceleyelim.

5 birimden elde edilen 3 farklı değişkene ait ölçüm değerleri Tablo 7.1'de verilmiştir. Bu 5 birimin Birleştirici ve Ayırıcı aşamalı yöntemlerle kümelenmesi sırasıyla şekil 7.1 ve şekil 7.2'de gösterilmiştir.

ÖRNEK 1

D1	D2	D3
41	77	66
62	71	51
49	62	83
63	93	93
53	74	53

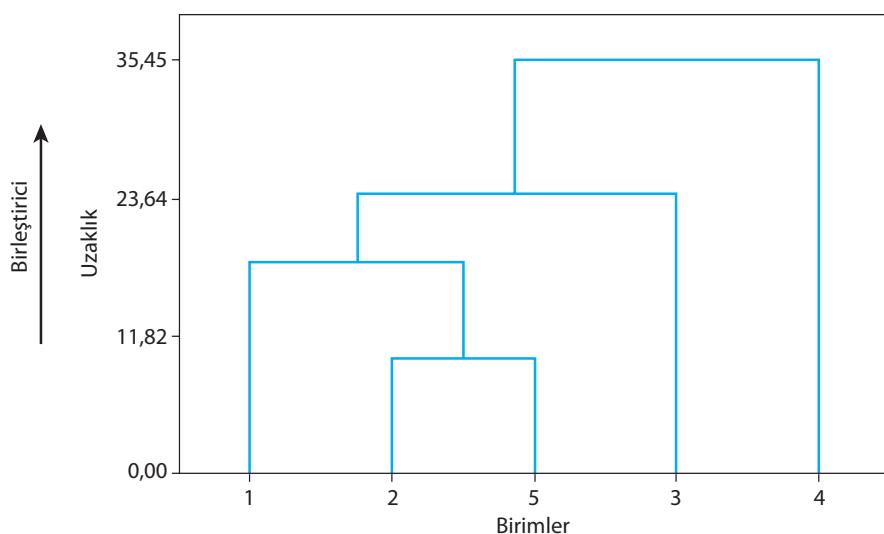
Tablo 7.1
Beş Birimden Elde Edilen Üç Değişkene İlişkin Veriler

Birimler arası öklid uzaklık matrisi aşağıdaki gibi elde edilir.

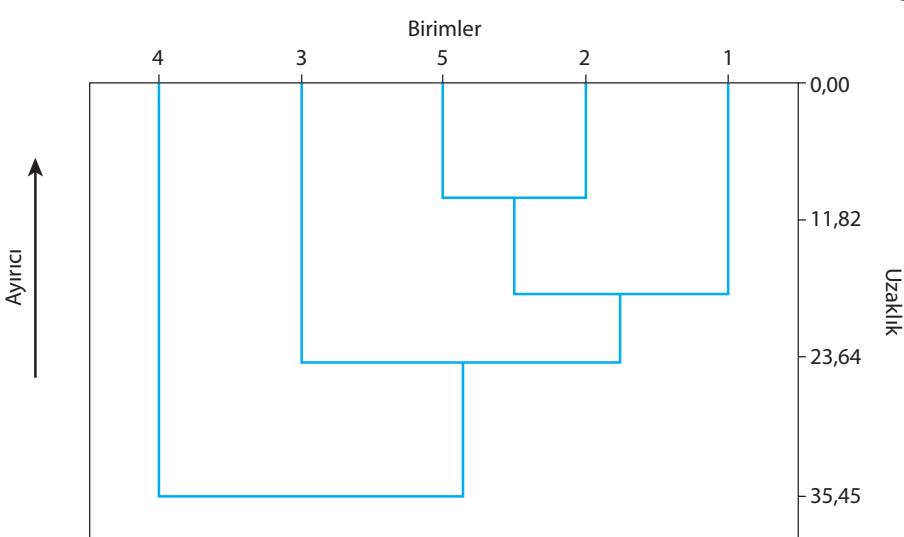
0,0000	26,4953	24,0416	38,3275	17,9444
26,4953	0,0000	35,6931	47,4236	9,6954
24,0416	35,6931	0,0000	35,4542	32,5576
38,3275	47,4236	35,4542	0,0000	45,3982
17,9444	9,6954	32,5576	45,3982	0,0000

Şekil 7.1

*Beş Birimin 3
Değişkenine İlişkin
Öklid Uzaklığna Göre
Birleştirici Yöntem ile
Kümelenmesi*

**Şekil 7.2**

*Beş Birimin 3
Değişkenine İlişkin
Öklid Uzaklığna Göre
Ayırıcı Yöntem ile
Kümelenmesi*



Yukarıda verilen prototip örnekte, Birleştirici kümeleme yöntemine göre birimlerin kümelenmesinde tabanda tüm birimler ayrı birer küme olarak görülürken (Şekil 7.1), Ayırıcı yönteme göre çizilen dendrogramda y ekseninde bulunan uzaklıklara göre yorumlama yapılmıştır. y ekseninde bulunan 5.00 değerinde tüm birimler farklı bir kümede yer alırken, 20.00 değerinde 3 küme oluşmuş 1., 2. ve 5. birimler bir kümede yer alırken 3. ve 4. birimler ise bu kümeden farklı 2 küme daha oluşturmuştur. 36.00 uzaklık düzeyinde ise tüm birimler aynı küme içerisinde yer almıştır. Bu durumu aşağıdaki Tablo 7.2'deki gibi bir tabloda göstermek mümkündür.

Tablo 7.2
*Birimlerin Değişik
Uzaklık Düzeylerinde
Kümelenmeleri*

Uzaklık Düzeyleri (ya da benzerlik düzeyleri)	Kümeler ve elemanları
5.00	[1], [2], [3], [4], [5]
11.82	[5,2], [1], [3], [4]
20.00	[1,2,5], [3], [4]
36.00	[1, 2, 3, 4, 5]

Ayırıcı aşamalı kümeleme yöntemi, Birleştirici aşamalı kümelemenin tersidir. Birleştirici yönteme ilişkin sonuçlardan ayrıcı yönteme ilişkin sonuçlar da elde edilebilir. Aşamalı kümeleme yöntemlerinde, birimlerin benzerlikleri yüzde yüze yakınsarken (similarity \rightarrow %100), farklılıklar sıfıra doğru yakınsar (dissimilarity \rightarrow 0).

Kümeleme analizinde en sık kullanılan uzaklık ölçüsü nedir? Kisaca açıklayınız



SIRA SİZDE

BİRLEŞTİRİCİ KÜMELEME YÖNTEMLERİ

Birleştirici aşamalı kümeleme yöntemlerinde, birimlerin birbirleri ile birleştirilmesinde farklı yöntemler kullanılmaktadır. Bunlardan sıkılıkla kullanılan ve genel kabul görmüş olanları aşağıdaki gibi sayılabilir.

- **Tek Bağlantı Kümeleme Yöntemi** (TekBKY, SINGLE Linkage [SLINK], En Yakın Komşuluk, Nearest Neighbour Method)
- **Tam Bağlantı Kümeleme Yöntemi** (TamBKY, COMPLETE linkage Method [CLINK], Furthest Neighbor Method)
- **Ortalama Bağlantı Kümeleme Yöntemi** (OrtBKY, AVERAGE Linkage Method, [ALINK])
- **McQuitty Bağlantı Kümeleme Yöntemi** (McQuitty linkage Method)
- **Küresel Ortalama Bağlantı Kümeleme Yöntemi** (KOBKY, CENTROID linkage Method)
- **Medyan Bağlantı Kümeleme Yöntemi** (MBKY, MEDIAN linkage Method)
- **Ward Bağlantı Kümeleme Yöntemi** (WBKY, WARD linkage Method, En Küçük Varyans Kümeleme Yöntemi)

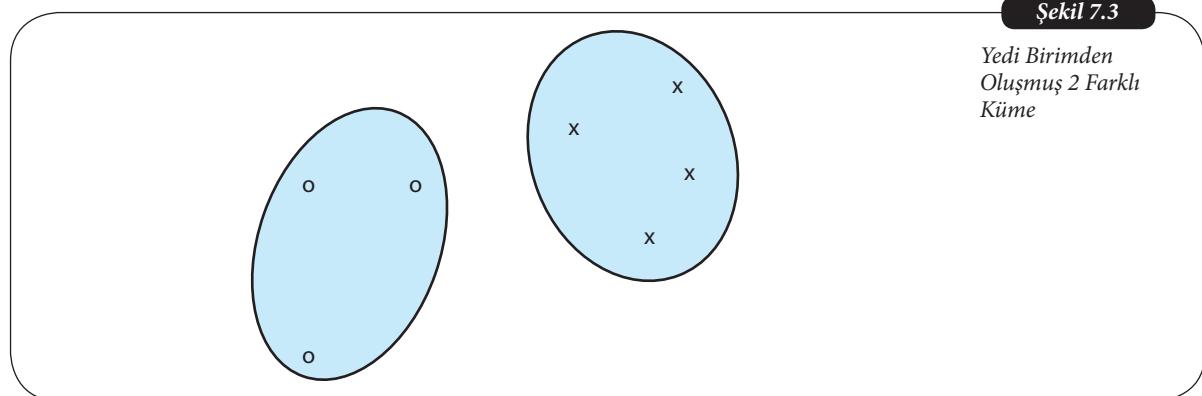
Yukarıda sayılan yöntemlerin, birimleri birleştirmede uydukları kriterler aşağıdaki alt başlıklar altında açıklanmıştır.

Tek Bağlantı Kümeleme Yöntemi

TekBKY en basit aşamalı kümeleme yöntemidir. Bu yöntem, farklı veri yapılarındaki kümelenmeleri tanımlayabilmesi açısından uygulayıcılar tarafından sıkılıkla tercih edilmektedir. Küme elemanları arasındaki en küçük uzaklık değeri temel alınarak kümelerin oluşturulması esasına dayanır.

Sekil 7.3

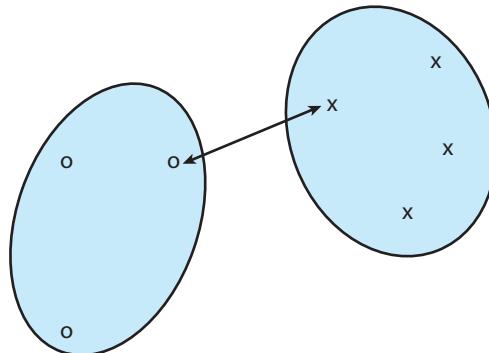
*Yedi Birimden
Oluşmuş 2 Farklı
Küme*



Şekil 7.3, 7 birime ait veri setinin 2 küme altında yapılanmasını ifade etmektedir. Şekil 7.4'te ise en küçük iki uzaklık değeri temel alınmış ve kümeler bu doğrultuda oluşturulmuştur.

Şekil 7.4

Tek Bağlantı
Kümeleme
Yönteminde Birimler
Arası En Küçük
Uzaklık



Literatürde en yakın komşuluk olarak da bilinen tek bağlantı kümeleme yöntemi, uzaklık matrisini kullanarak birbirine en yakın (uzaklık değerleri en küçük) birim ya da değişkenleri birleştirerek kümelerin olmasını sağlamaktadır. Bu yöntemin ilk aşamasında uzaklık matrisindeki en yakın (en küçük uzaklık) iki birim dikkate alınarak ilk küme oluşturulur. İkinci aşamada ise bir sonraki en küçük uzaklık belirlenir ve ilk oluşturulan kümeye bu birim ya da değişken eklenir ya da bu birim ile iki birimden oluşan yeni bir kume oluşturulur. İşlem, tüm birimler bir kümeye yerleşinceye kadar devam eder. Birleştirme yapılırken kümelerin eleman sayısının birden fazla olması koşulu yoktur. Tek bir birim de bir kume oluşturabilir. Bu yöntemde, m ve j kümeleri arasındaki uzaklık;

$$d_{mj} = \min(d_{kj}, d_{lj})$$

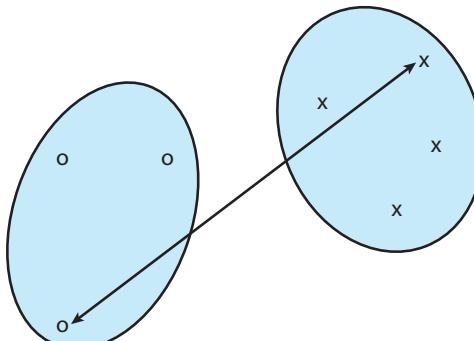
biçiminde hesaplanmaktadır.

Tam Bağlantı Kümeleme Yöntemi

Bu yöntem, en uzak komşuluk olarak da bilinmektedir. Tek bağlantı kümeleme yöntemine çok benzemekle birlikte bu yöntemdeki tek farklılık oluşturulan her kümedeki eleman çiftleri arasındaki uzaklığın maksimum olanının ele alınmasıdır.

Şekil 7.5

Tam Bağlantı
Kümeleme
Yönteminde Birimler
Arası Maksimum
Uzaklık



Bu yönteme tam bağlantı kümeleme yöntemi denmesinin nedeni, bir kume içindeki tüm birimlerin birbirlerine maksimum uzaklık veya minimum yakınlığa bağlı olmasıdır (Şekil 7.5). Tam bağlantı teknigideki uzaklıklar,

$$d_{mj} = \max(d_{kj}, d_{lj})$$

biçiminde hesaplanmaktadır.

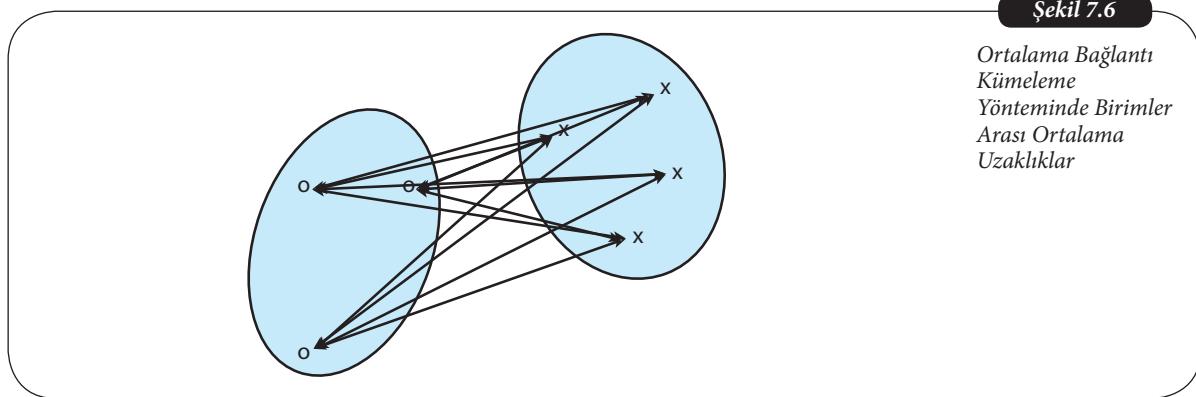
TamBKY, Maksimum Yöntem, Sıralama Tip Analizi (Rank Order Typical Analysis), En Uzak Komşu Analizi (Furthest Neighbor Analysis), Çap Yöntemi isimleriyle de anılmaktadır. TamBK yöntemi, TekBK yönteminin aksine en uzak komşu niteliğine sahip birimleri birbirleriyle birleştirerek kümeye oluşturmuyor içermektedir.

Ortalama Bağlantı Kümeleme Yöntemi

Bu yöntemde, tek bağlantı ve tam bağlantı yöntemlerinde olduğu gibi işlem başlatılır. Fakat kümeleme kriteri olarak, bir kümeye içindeki birim ile diğer kümeye içindeki birimler arasındaki ortalama uzaklıklar dikkate alınır. Ortalama bağlantı kümeleme yöntemindeki kümeleme kriteri, bir kümeyi tüm birimlerden elde edilen ortalama uzaklığın diğer kümeyi tüm birimlere olan ortalama uzaklığını olarak ele alınır.

Şekil 7.6

Ortalama Bağlantı
Kümeleme
Yönteminde Birimler
Arası ortalama
Uzaklıklar



Kümeler arasındaki ortalama uzaklığın en düşük değeri temel alınarak işlem yapılır (Şekil 7.6). Bu yöntemde uzaklıklar aşağıdaki gibi hesaplanmaktadır.

$$d_{mj} = (N_k d_{kj} + N_l d_{lj}) / N_m$$

Ortalama bağlantı yönteminde kümeler, küçük varyanslar ile birbirlerine bağlıdır ve tek bağlantı ve tam bağlantı yöntemleri arasında sonuçlar vermesi nedeniyle alternatif bir yöntem olarak araştırmacılar önerilmektedir. Bu yöntemin farklı bir özelliği de kümeye içi değişkenliği az olan kümeleri birleştirme eğilimine sahip olmasıdır. Aşırı değerlerden az etkilenen bir yapısı bulunmaktadır.

McQuitty Bağlantı Kümeleme Yöntemi

McQuitty bağlantı kümeleme yönteminde; m. kümeyi oluşturmada k. ve l. kümelerin j. kümeye ile olan uzaklıklarını toplamının yarısı (ortalaması) hesaplanır. Ağırlıksız ortalama bağlantı yöntemi ismi ile de literatürde sıkılıkla kullanılmaktadır. Yeni oluşan m. ve j. kümeler arasındaki uzaklık;

$$d_{mj} = (d_{kj} + d_{lj}) / 2$$

şeklinde belirlenmektedir.

Küresel Ortalama Bağlantı Kümeleme Yöntemi

Bir kümeyi oluşturan gözlemlerin ortalamalarını esas alır. Kümeye sadece tek bir merkez varsa onun değeri merkez olarak kabul edilir. Ortalama bağlantı kümeleme yönteminin farklı bir biçimidir. m kümeyi j kümeye olan uzaklığını;

$$d_{mj} = (N_k d_{kj} + N_l d_{lj}) / N_m - N_k N_l d_{kl} / N_m^2$$

formülü yardımıyla hesaplanır.

Medyan Bağlantı Kümeleme Yöntemi

McQuitty bağlantı kümeleme yönteminin farklı bir biçimidir. Bu yöntemde m. ve j. kümeler arasındaki uzaklık;

$$d_{mj} = (d_{kj} + d_{lj})/2 - d_{kl}/4$$

formülü yardımıyla hesaplanır.

Ward Bağlantı Kümeleme Yöntemi

Bu yöntem, küresel ortalama ve medyan bağlantı kümeleme yöntemlerinin karma şeklidir. Küme içi varyansın minimum olduğu kümeler belirlenir ve bu doğrultuda kümeler oluşturulur. Minimum varyans yöntemi olarak da bilinen bu yaklaşım, bir kümeye yer alan bir birimin, aynı kümenin içinde bulunan birimlerden ortalama uzaklığını dikkate almaktadır. Küme bağlantılarından ziyade küme içi kareler toplamı dikkate alınmaktadır. Bu yöntem, az birimli kümeleri birleştirme eğilimindedir. Ayrıca bu yöntemin birbirine eşit sayıda birim içeren kümeler oluşturma gibi bir eğilimi de vardır. Bundan dolayı, araştırmacının *kümelerdeki birim sayılarının benzer (yakın) olduğu* beklenisi durumunda bu yönteme başvurması önerilmektedir. Aşırı değerlerden etkilenmektedir. Sıklıkla kullanılan aşamalı kümeleme yöntemidir.

Bu yöntemde, m ve j kümeleri arasındaki uzaklık;

$$d_{mj} = ((N_j + N_k)d_{kj} + (N_j + N_l)d_{lj} - N_jd_{kl})/(N_j + N_m)$$

biriminde hesaplanır.

SIRA SİZDE



4

Aşamalı kümeleme yöntemlerinden olan birleştirici kümeleme yöntemleri nelerdir?

R PROGRAMINDA TEK BAĞLANTı KÜMELEME YÖNTEMİ UYGULAMASI

ÖRNEK 2

Ondokuz ülkenin D1, D2, D3 ve D4 değişkenlerine ilişkin verileri Tablo 7.3'te verilmiştir. Bu dört değişkenin göre ülkeler gruplara ayrılmak istenmektedir. Aşamalı tek bağlantı kümeleme yöntemini kullanarak aşağıdaki şekilde kümeleme analizi yapılabilir.

Tablo 7.3
Ondokuz Ülkenin
D1, D2, D3 ve D4
Değişkenlerine İlişkin
Veri Seti

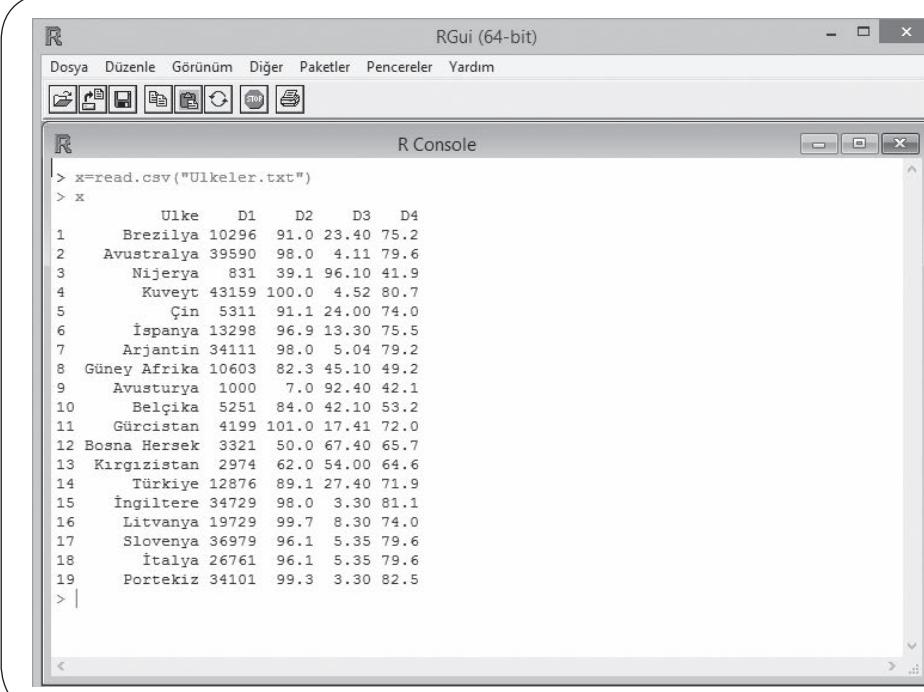
Sıra No	Ülkeler	D1	D2	D3	D4
1	Brezilya	10296	91	23.4	75.2
2	Avustralya	39590	98	4.11	79.6
3	Nijerya	831	39.1	96.1	41.9
4	Kuveyt	43159	100	4.52	80.7
5	Çin	5311	91.1	24	74
6	İspanya	13298	96.9	13.3	75.5
7	Arjantin	34111	98	5.04	79.2
8	Güney Afrika	10603	82.3	45.1	49.2
9	Avusturya	1000	7	92.4	42.1
10	Belçika	5251	84	42.1	53.2
11	Gürcistan	4199	101	17.41	72
12	Bosna Hersek	3321	50.0	67.4	65.7
13	Kırgızistan	2974	62	54	64.6
14	Türkiye	12876	89.1	27.4	71.9
15	İngiltere	34729	98	3.3	81.1
16	Litvanya	19729	99.7	8.3	74
17	Slovenya	36979	96.1	5.35	79.6
18	İtalya	26761	98.4	5.93	81
19	Portekiz	34101	99.3	3.3	82.5

Veriler ilk olarak adı ve uzantısı *ulkeler.txt* olarak herhangi bir kelime işlemcide (WordPad, NotePad, Excel vd.) hazırlanarak sabit diskte bulunan C klasörüne kaydedilir. Veriler herhangi bir editörde aşağıdaki formda görüntülenebilir.

```
Ulke, D1, D2, D3, D4
Brezilya, 10296, 91, 23.4, 75.2
Avustralya, 39590, 98, 4.11, 79.6
Nijerya, 831, 39.1, 96.1, 41.9
Kuveyt, 43159, 100, 4.52, 80.7
Çin, 5311, 91.1, 24, 74
İspanya, 13298, 96.9, 13.3, 75.5
Arjantin, 34111, 98, 5.04, 79.2
Güney Afrika, 10603, 82.3, 45.1, 49.2
Avusturya, 1000, 7, 92.4, 42.1
Belçika, 5251, 84, 42.1, 53.2
Gürcistan, 4199, 101, 17.41, 72
Bosna Hersek, 3321, 50, 67.4, 65.7
Kirgızistan, 2974, 62, 54, 64.6
Türkiye, 12876, 89.1, 27.4, 71.9
İngiltere, 34729, 98, 3.3, 81.1
Litvanya, 19729, 99.7, 8.3, 74
Slovenya, 36979, 96.1, 5.35, 79.6
İtalya, 26761, 96.1, 5.35, 79.6
Portekiz, 34101, 99.3, 3.3, 82.5
```

Verilere R programında kümeleme analizi uygulamak için öncelikle R programı çalıştırılır. Daha sonra `x=read.csv("c:/ulkeler.txt")` komutu yardımıyla veriler veri dosyası olan *ulkeler.txt* dosyasından alınır.

Şekil 7.7



```
RGui (64-bit)
Dosya Düzenle Görünüm Diğer Paketler Pencereler Yardım
R Console
> x=read.csv("Ulkeler.txt")
> x
   Ulke     D1     D2     D3     D4
1  Brezilya 10296  91.0  23.40  75.2
2  Avustralya 39590  98.0   4.11  79.6
3    Nijerya   831  39.1  96.10  41.9
4     Kuveyt  43159 100.0   4.52  80.7
5      Çin   5311  91.1  24.00  74.0
6    İspanya 13298  96.9  13.30  75.5
7    Arjantin 34111  98.0   5.04  79.2
8  Güney Afrika 10603  82.3  45.10  49.2
9    Avusturya  1000   7.0  92.40  42.1
10   Belçika   5251  84.0  42.10  53.2
11  Gürcistan  4199 101.0  17.41  72.0
12 Bosna Hersek  3321  50.0  67.40  65.7
13  Kirgızistan  2974  62.0  54.00  64.6
14  Türkiye 12876  89.1  27.40  71.9
15  İngiltere 34729  98.0   3.30  81.1
16  Litvanya 19729  99.7   8.30  74.0
17  Slovenya 36979  96.1   5.35  79.6
18    İtalya 26761  96.1   5.35  79.6
19  Portekiz 34101  99.3   3.30  82.5
> |
```

*R Programında
Verilerin Sabit
Diskten Çağrılması ve
Görüntülenmesi*

Veri setine ilişkin uzaklık matrisini bulmak için ***dist.x=dist(x,method="euclidean")*** komutu kullanılır. Bu komut yardımıyla *x* veri setinde birimler arasındaki Öklid uzaklıklar hesaplanır ve hesaplanan uzaklıklar ***dist.x*** matrisine atanır. Eğer bu matris görüntülenmek istenirse komut satırına (>) ***dist.x*** yazmak yeterlidir ve aşağıdaki sonuçlar elde edilir. Anılan işlemler Şekil 7.8'de görüntülenmektedir.

Sekil 7.8*Veri Setine İlişkin Uzaklık Matrisinin Elde Edilmesi ve Görüntülenmesi*

```
> dist.x=dist(x,method="euclidean")
> dist.x
      1         2         3         4         5         6         7         8         9
2 32751.69607
3 10582.72843 43334.07195
4 36741.95893 3990.26415 47324.32230
5 5573.39964 38325.09484 5009.90679 42315.35784
6 3356.36353 29395.35181 13939.03757 33385.61507 8929.74799
7 26625.98888 6125.70833 37208.39207 10115.97193 32199.38730 23269.64364
8 345.45546 32408.50621 10925.68674 36398.76523 5916.75605 3013.49902 26282.80843
9 10394.02043 43145.18488 192.37054 47135.43177 4821.50002 13750.28447 37019.51530 10736.94363
10 5640.57928 38392.20717 4942.35011 42382.46820 74.25177 8996.92323 32266.50441 5983.72082 4753.89095
11 6816.66663 39568.34474 3767.35214 43558.60769 1243.32691 10172.99409 33442.63667 7160.03250 3579.27271
12 7798.58419 40550.07497 2784.24338 44540.33176 2225.91039 11154.91772 34424.38226 8141.66267 2595.68680
13 8186.38914 40937.99375 2396.68028 44928.25359 2613.28437 11542.74498 34812.29444 8529.53468 2208.41623
14 2884.53430 29867.17422 13467.09623 33857.43657 8457.92860 472.17133 23741.46840 2541.50640 13278.32959
15 27316.93561 5434.76355 37899.34072 9425.02690 32890.33389 23960.58987 690.95101 26973.75779 37710.46244
16 10546.43270 22205.27451 21128.97346 26195.53777 16119.82647 7190.07963 16079.56640 10203.31736 20940.15654
17 29832.50870 2919.18785 40414.89221 6909.45160 35405.90749 26476.16479 3206.52223 29489.32160 40226.00781
18 18408.44299 14343.25828 28990.90538 18333.52151 23981.84023 15052.09520 8217.55014 18065.29023 28802.04995
19 26614.81146 6136.88966 37197.22409 10127.15219 32188.20956 23258.46523 12.02121 26271.63749 37008.34857
          10        11        12        13        14        15        16        17        18
2
3
4
5
6
7
8
9
10
11 1176.83688
12 2158.37101 984.90081
13 2545.94888 1370.92104 388.48065
14 8525.05255 9701.19647 10683.00006 11070.85697
15 32957.45295 34133.58300 35115.32988 35503.24217 24432.41581
16 16186.96642 17363.07104 18344.90719 18732.77953 7661.92621 16770.51275
17 35473.02105 36649.15803 37630.89045 38018.80751 26947.98702 2515.57897 19286.08802
18 24048.97057 25225.08829 26206.86826 26594.76625 15523.92731 8908.49532 7862.01949 11424.07171
19 32255.33092 33431.45817 34413.21006 34801.12087 23730.29241 702.12859 16068.38828 3217.70626 8206.37024
> |
```

Sekil 7.9

Tek Bağlantı	> h=hclust(dist.x,method="single")
Kümleme Yönteminin	> h
Uygulanması	call: hclust(d = dist.x, method = "single") Cluster method : single Distance : euclidean Number of objects: 19

Elde edilen Öklid uzaklık matrisi yardımıyla, verilere Hiyerarşik kümeleme yöntemlerinden Tek bağlantı kümleme yöntemi uygulamak için ise ***h=hclust(dist.x,method="single")*** komutu kullanılır. Elde edilen sonuçları görmek için ise komut satırına (>*h*) ***h*** yazmak yeterlidir ve Şekil 7.9'da da belirtilen sonuçlar elde edilir.

Şekil 7.9'da kullanılan komutlar ve çıktılar görülmektedir. Bu bilgi ekranında 19 ülkenin Öklid uzaklığından yararlanılarak, Tek bağlantı kümleme yöntemi ile Aşamalı (Hiyerarşik) kümelenmesinin yapıldığı özetlenmiştir.

Kümelenme adımlarını görüntülemek için ise ***h\$merge*** komutu kullanılır ve kümelere ait birleşmeler Şekil 7.10'daki gibi elde edilir.

Ülkelerin eldeki değişkenlere göre 3 Alt kümede toplanması istenildiğinde ise ***clusters=cutree(h, k=3)*** komutu kullanılır ve ülkeler 3 farklı kümeye ayrılır. Ayrıca ***clusters*** komutu yardımıyla ülkelerin veri giriş sırasına göre küme üyelikleri de görüntülenebilmektedir. Yapılan analizler sonucunda elde edilen dendrogramın görüntülenmesi için ise ***plot(h, labels=x\$Ulke)*** komutu kullanılır. Veri dosyasında bulunan ***ülke*** sütunundaki ülkelere ait isimlerin dendrogramda gösterimi için komutta bulunan ***labels=x\$Ulke*** ifadesi kullanılmıştır. Bu işlemler Şekil 7.11'de görülmektedir.

Bu komutun işleme konması ile ülkelerin kümelenmesine ilişkin elde edilen dendrogram Şekil 7.12'de verilmiştir.

Şekil 7.12'de çizdirilen dendrogram, ülkelere ait kümelenmelerin daha net olarak görülebilmesi için ***rect.hclust(h, K=3)*** komutu yardımıyla 3 farklı dikdörtgen ile böülümlendirilebilmektedir. Bu komut yardımıyla (**>rect.hclust(h, k=3)**) elde edilen dendrogram ise şekil 7.13'te verilmiştir.

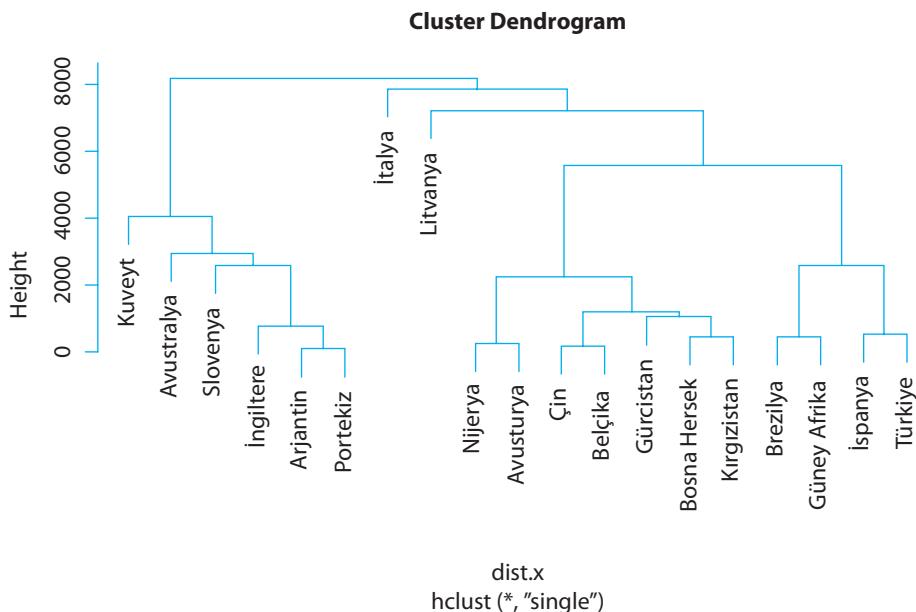
Şekil 7.10

Kümelenme Adımları		
> h\$merge	[,1]	[,2]
[1,]	-7	-19
[2,]	-5	-10
[3,]	-3	-9
[4,]	-1	-8
[5,]	-12	-13
[6,]	-6	-14
[7,]	-15	1
[8,]	-11	5
[9,]	2	8
[10,]	3	9
[11,]	-17	7
[12,]	4	6
[13,]	-2	11
[14,]	-4	13
[15,]	10	12
[16,]	-16	15
[17,]	-18	16
[18,]	14	17

```
> clusters=cutree(h, k=3)
> clusters
[1] 1 2 1 2 1 1 2 1 1 1 1 1 2 1 2 3 2
> plot(h, labels=x$Ulke)
```

Şekil 7.11

Kümelenmenin Yapılması ve Dendrogramın Cizdirilmesi



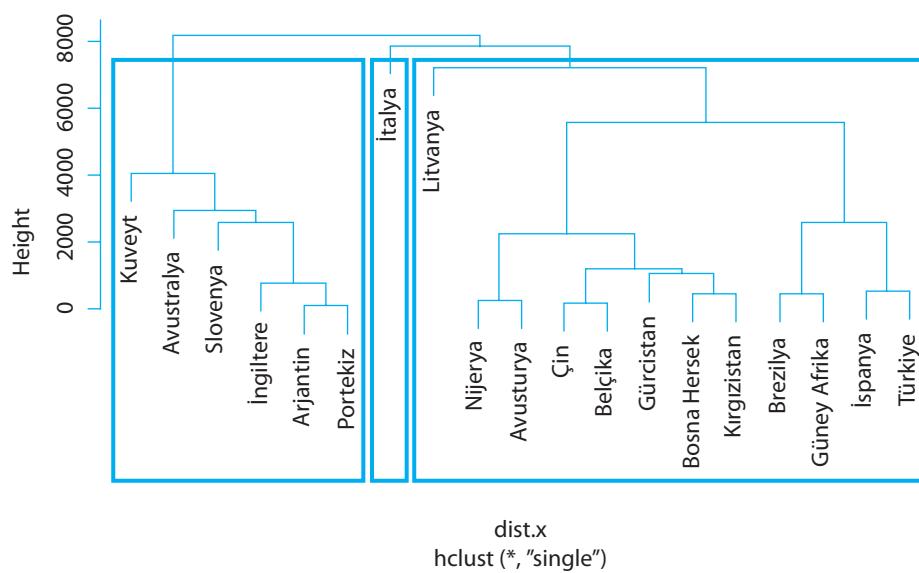
Şekil 7.12

Öklid Uzaklık Matrisi ile Tek Bağlantı Kümeleme Analizi Sonuçlarını Gösteren Dendrogram

Şekil 7.13

Öklid Uzaklık
Matrisi ile Tek
Bağlantı Kümeleme
Analizi Sonuçlarını
Küme Ayırımları
ile Gösteren
Dendrogram

Cluster Dendrogram



Şekil 7.13'teki dendrogram inceleneyece olursa, *Kuveyt*, *Avustralya*, *Slovenya*, *İngiltere*, *Arjantin*, *Portekiz* gibi ülkeler incelenen dört değişken bakımından bir küme içerisinde yer almışlardır. *Litvanya*, *Nijerya*, *Avusturya*, *Çin*, *Belçika*, *Gürcistan*, *Bosna Hersek*, *Kırgızistan*, *Brezilya*, *Güney Afrika*, *İspanya*, *Türkiye* farklı uzaklıklarda bir küme oluşturmuştur. *İtalya* ise tek başına bir küme oluşturmuştur.

AŞAMALI OLMAYAN KÜMELEME YÖNTEMLERİ

Birimlerin kendi içinde homojen ve kendi aralarında heterojen olan kümelere ayrılmasını hedefleyen ve elde edilen kümelerin aracılığı ile alt toplum yapılarına ilişkin tahmin yapmayı amaçlayan yöntemlerdir. Aşamalı kümelemede hem birimler hem de değişkenler birbirileşirile farklı benzerlik düzeylerinde kümeler oluştururken, aşamalı olmayan kümeleme yöntemlerinde sadece birimler kümelenmektedir. Birbirleri ile benzer birimlerin aynı kümeye toplanması koşuluyla veri setindeki n birimin k sayıda kümeye ayrılması amaçlanmaktadır. Bu yöntemlerde kümeye sayısı önceden belirlenir. Diğer bir ifadeyle, eğer oluşturulacak kümeye sayısı ile ilgili olarak önsel bir bilgi var ise aşamalı olmayan kümeleme yöntemleri kullanmak daha çok tercih edilmektedir. Örneğin; kabul gören sağlık veya ekonomik göstergeler bakımından ülkeler 4 farklı kümeye ayrılmak istenilebilir. Bu kümeler ise, *geri kalmış, az gelişmiş, gelişmekte olan, gelişmiş* ülkeler olarak isimlendirilebilir.

Aşamalı kümeleme yöntemleri daha çok küçük veri setleri için uygundur. Buna karşılık aşamalı olmayan kümeleme yöntemleri ise daha çok büyük veri setlerine uygulanmaktadır. Bunun nedeni aşamalı olmayan kümeleme yöntemlerinde başlangıçta benzerlik ve uzaklık matrislerinin hesaplanmamasıdır. Ayrıca aşamalı olmayan kümeleme yöntemleri veri setinde bulunan aşırı uç değerlerden daha az etkilenmektedir.

Aşamalı olmayan Kümeleme Yöntemleri arasında en yaygın kullanılan yöntem K-ortalamlar kümeleme (*k-means clustering*, *MacQueens' Method*) yöntemidir. Bu yöntem birçok istatistik hazır yazılımda bulunmaktadır. Bunun dışında Medoid kümeleme ve Fuzzy kümeleme gibi aşamalı olmayan kümeleme yöntemleri de bulunmaktadır.

Aşamalı olmayan kümeleme yöntemlerinde veri setinde bulunan birimlerin k kümeye ayrılma işlemi rastgele yapılabileceği gibi verilerin değerlendirilmesi sonucunda araştırmanın belirlediği şekilde de yapılabılır. Birimlerin ayrılabilecekleri küme sayısına karar verilmesinin ardından, kümeler için belirlenen küme belirleme kriterlerine göre birimlerin hangi kümelere girebileceklerine karar verilir ve birimlerin bu kümelere atama işlemleri yapılır. Aşamalı olmayan birçok kümeleme yöntemi vardır. Yöntemlerden bazıları küme sayısını tamamen deneme yoluyla (kullanıcı tanımlı) belirlemeyi önerirken, bazı yöntemler ise çekirdek noktalar seçip bu noktaları oluşacak kümelerin merkezi kabul ederek çekirdekler etrafında birimlerin atanmasını önermektedir.

k-Ortalamlar Yöntemi

Mac Queen'in k-ortalamlar adını verdiği yöntem gözlemleri kümelerin önceden belirlenmiş sayısına gruplandırmakla işleme başlamaktadır. Bu yöntem, değişkenlerin ortalama vektörlerini küme merkezi olarak ele alır ve kümeleme süreci bunun etrafında şekillenir. Bu kümeleme yöntemi, veri setinde bulunan birimleri küme içi kareler toplamlarını minimize (en küçük) edecek biçimde k sayıda kümeye ayırmayı amaçlar. Daha sonra her kümenin kendisini temsil edebilecek tipik bir gözlemi seçilir ve benzer gözlemler, tipik gözlemin etrafında işlem sırasıyla sırayla kümelendirilir. Birimlerin kümelere yerleştirilmesi işlemi tekrarlı bir biçimde yapılır. Birimler her iterasyonda farklı kümelere atanır ve en uygun çözüm permütasyon yaklaşımına benzer bir şekilde belirlenir. Bu yöntemde başlangıçta seçilen küme merkezi noktaları değişmeksizin birimlerin kümeleme atanma işlemleri yapılır. Farklı aşamalardaki atamalarda, kümeler arası heterojeniteye bağlı olarak birimlerin atandıkları kümelerden çıkarılarak başka bir kümeye atanması mümkün olabilmektedir. Birimlerin kümelere atanma işlemleri; her kümeleme aşamasında denetlenen küme içi homojenite ya da kümeler arası heterojenitenin maksimizasyonun sağlanması ile son bulur. K-ortalamlar yönteminde kümelerin belirlenmesinde kullanılan çekirdek noktaların veri setinde bulunan gözlenen değerlerden seçilmesi zorunlu değildir. Olasılık kurallarına göre de k çekirdek nokta seçilerek veri seti üzerinde kümeleme yapılabilir.

k-ortalamlar algoritması basittir. Veri setine uygulanması ve yorumlanması kolaydır. Sürekli yapıdaki veri setleri için ilk tercih edilen k ortalamlar kümeleme algoritmasıdır. Karışık yapıda ya da kesikli değişken içeren veri setleri için uygun bir seçim değildir. Az sayıda iterasyon ile yeterli yakınsamayı sağlayan hızlı bir algoritması bulunmaktadır. Genellikle değişken bazında standardizasyon ya da normalleştirme uygulanarak, algoritmanın farklı değişim aralığı gösteren değişkenlerden etkilenmesi önlenmeye çalışılır. K-ortalamlar yönteminin bazı kısıtları da bulunmaktadır. Eğer veri setindeki değişkenler asimetrik bir yapı sergiliyor ya da çok sayıda sapan değer içeriyor ise ortalama uygun bir konum parametresi olmamaktadır. Böyle durumlarda küme merkezleri, üyelerini doğru temsil edememektedir. Bazı durumlarda, en küçük farklara bağlı kümeler olmuş olsa bile homojen bir yapı oluşmayabilir. Bu durumda başka algoritmaların kullanılması önerilmektedir.

k-Medyanlar Yöntemi

Medyan değerlerine ait vektörleri küme merkezi olarak kullanan k merkezli algoritmadır. Veri setindeki değişkenlerin asimetrik olduğu durumlarda kullanılmaktadır. Bu yöntemde de uzaklık ölçüsü seçimi keyfi olarak gerçekleştirilebilir, ancak yakınsamanın sağlanıp sağlanmadığı mutlaka göz önünde bulundurulmalıdır. Bu algoritmada, Manhattan uzaklık ölçüsü en sık tercih edilen kümelenme ölçütüdür.

k-Medoidler Yöntemi

Veri setinin asimetrik olduğu durumlarda k-medyanlar yöntemi, k-ortalamalar yöntemine göre daha çok tercih edilmektedir. Fakat k-medyanlar yönteminde de yakınsama gözlenmediği durumlar olabilmektedir. Özellikle değişkenlerin birbirinden bağımsız olmadığı ve değişkenler arasında korelasyon olduğu durumlarda k-medyanlar yöntemi veri setini grüplamada (kümelemede) başarılı olmamaktadır. Bu durumda kümeleme için k-medoidler yöntemi önerilmektedir. Medoid, diğer küme elemanları ile aralarında en az fark görülen seçilmiş küme elemanları olarak tanımlanabilmektedir. Bu algoritma k-ortalamalar ve k-medyanlar yöntemlerine göre daha çok işlem gerektirmektedir. Çünkü, medoidler belirlenirken tüm ikili uzaklık ölçüleri hesaplanmaktadır.

k-Ortalamalar Yönteminin Uygulanması

K-Ortalamalar Kümeleme Yöntemi sadece birimleri kümelemekte kullanılan bir yöntemdir.

Birimlerin k-ortalamalar yöntemi ile kümelenmesi için uzaklık matrisi ya da benzerlik matrisi hesaplamak gerekmektedir. Verilerin kümelenmesinde kullanılacak olan küme sayısını önceden belirlemek yeterlidir. Küme sayısını belirlemek için ise farklı yaklaşımalar bulunmaktadır. Bunlar;

- Aşamalı kümeleme yöntemlerinden elde edilen dendrogramları inceleyerek karar vermek,
- Olasılıklı olarak başlangıç noktalarını rassal olarak belirlemek,
- Ardışık olarak (Küme sayısı 2, 3, 4, ..., k biçiminde) her seferinde küme sayısını bir artırarak oluşan kümelemede birimlerin hangi kümeye ait olduğunu ilişkin küme üyeliklerini belirlemek. Yeni veri yapısına Ayırma (Discriminant) Analizi uygulamak ve en yüksek önemliliği bulunan Wilk's Lamda değerine sahip olan küme sayısını, uygun kümeleme olarak kabul etmek,
- İlk n_b birimin değişkenlere ait ortalamalarını başlangıç ortalama vektörü olarak ele alıp birimleri bu kümelere atama yaklaşımlarından birini seçmek,
- Farklı rastgele başlatma konfigürasyonları seçerek küme sayısını bulmak,

mömkündür.

Bu yöntem, veri setinde bulunan n birimin çoğunun profilini yansıtan çekirdek noktalı bir kümeye atanmasını içerir. Bu yöntemde ortalamalar, başlangıçta ele alınan k noktanın değerleridir. k-ortalamalar yöntemi, birimleri aşağıdaki şekilde kümelere ayırrı.

- a. Verilerden elde edilecek ön bilgilere göre ilk k nokta çekirdek nokta olarak alınır. Bu noktaların herbiriin p değişken değerleri birer küme ortalama vektörü olarak ele alınır ve küme ortalama vektöründen her bir birimin uzaklıklar hesaplanır.
- b. Geriye kalan $n-k$ birim kendisine en yakın ortalama vektörlü kümeye atanır. Her atama sonrasında oluşan kümenin ortalama vektörü yeniden hesaplanır. Böylece, her aşamada çekirdek noktaların verilerinden hesaplanan ortalama vektörü değişir ve birimlerin yeni oluşan küme ortalama vektörüne göre uzaklıkları hesaplanır. En yüksek benzerliğe sahip birimler bir araya getirilir.
- c. Küme içi varyansın en az ve kümeler arası varyansın en fazla olduğu kümeleme yapısı oluşturuluncaya kadar tüm birimler k kümeye atanmaya çalışılır. Tekrarlamalı (iterative method) en uygun kümeleme sağlanır. Her birimin farklı aşamalarda kümelerde yer olması sağlanır. Her aşamada birimlerin kümelerde yer alma olasılığı 0 ile 1 arasında değişir. Küme içi kovaryans matrisinin minimum olduğu koşul sağlanıncaya ve yakınsama kriterine eşit ya da daha küçük varyans farkına ulaşılınca kadar kümelenme işlemine devam edilir.

K-ortalamalar yöntemi, küme sayısı ikiden başlayarak her defasında birer arttırarak en uygun kümelemeyi bulmak şeklinde de uygulanabilir. Bu durumda uygulamacının veri yapısını çok iyi tanıma zorunluluğu getirilmektedir. Bu yaklaşımda toplam küme içi varyans matrisinin izi (trace) "tr(W)" minimize edilir.

Uygun küme sayısının belirlenmesinde oluşan kümeler içi kovaryans matrisi ve küme-lerarası kovaryans matrislerinden yararlanılarak Wilk's Lamda değerleri hesaplanır. 2, 3, 4 ve daha fazla küme için hesaplanan Wilk's Lamda değerleri içinden en yüksek önemliliğe sahip olan "k kümeye ayrılma" çözümü uygun kümeleme olarak dikkate alınır. Ayrıca her kümeleme sonucunda hesaplanan Mahalanobis Uzaklık matrisinin Hotelling T² istatistikleri bulunarak bu istatistiklerin önemlilik düzeylerinin en yüksek olduğu durum uygun kümeleme olarak kabul edilebilir. Bu bağlamda Mahalanobis D² uzaklık matrisi, kümelerin birbirlerinden olan ayrimını (discrimination) değerlendirmeye yarayan bir uzaklıktır. Bu matrisi kümelere ayrmada yeterince farklılaşmayı sağlayıp sağlayamadığı çok değişkenli analizlerden Hotelling T² testi ile test edilmektedir. Kümeleme sonuçlarının değerlendirilmesinde literatürde çok farklı kriterler kullanılmaktadır.

Birbirinden farklı sayıdaki küme konfigürasyonlarının uygunluğunu değerlendirmek için Kümeçi Kareler Toplami (KKT_k) hesaplanır. KKT, k küme için aşağıdaki gibi hesaplanır.

$$KKT_k = \left[\frac{NP}{NP - m} \right] \sum_{k=1}^K \sum_{i=1}^P \sum_{j=1}^{n_i} (1 - d_{ijk}) (z_{ij} - c_{ik})^2$$

Yukarıdaki eşitlikte, KKT_k, k küme için kümeçi kareler toplamı; N, birim sayısı; P, değişken sayısı; m, k, küme sayısı; d_{ijk}, i. değişken, j. birim ve k. kümenin uzaklık ölçüsü; z_{ij}, i. değişken j.birimin standardize değeri; c_{ik} i. değişkenin k. kümedeki ortalama değeridir.

Örnek 2 verilerine aşamalı kümeleme yöntemi uygulanmıştır. Aynı veri setine karşılaşma-tırma amaçlı olarak k-ortalamalar kümeleme yönteminin uygulanışı da aşağıda verilmiştir.

R PROGRAMINDA K-ORTALAMALAR KÜMELEME YÖNTEMİ UYGULAMASI

R paket programında k-ortalamalar yöntemine göre birimlerin kümelenmesi için Örnek 2 verilerini kullanalım.

Ondokuz ülkenin D1, D2, D3 ve D4 değişkenlerine ilişkin verileri Tablo 7.3'te verilmiştir. Ülkeler bu dört değişkene göre kümelere ayrılmak istenmektedir. Aşamalı olmayan k-ortalamalar kümeleme yöntemini kullanarak ve aşağıdaki işlem adımları izleyerek bi-rimlerin kümeleme işlemi yapılabilir. Verilerin R programına aktarılması için Örnek 2'de yapılan işlemleri hatırlayalım,

Veriler ilk olarak adı ve uzantısı **ulkeler.txt** olarak herhangi bir kelime işlemcide (WordPad, NotePad, Excel vd.) hazırlanmış sabit diskte bulunan C klasörüne kaydedilmiştir.

Verilere R programında kümeleme analizi uygulamak için öncelikle R programı çalıştırılır. Daha sonra **x=read.csv("c:/ulkeler.txt")** komutu yardımıyla veriler veri dosyası olan ulkeler.txt dosyasından alınır. R programında artık veri matrisi **x** ile tanımlanır ve görünütlenmek istediğiinde **x** yazmak yeterlidir. Bu işlemler için Şekil 7.14'ü inceleyebilirsiniz.

Şekil 7.14

R Programında
Verilerin
Sabit Diskten
Çağrılması ve
Görüntülenmesi

```

RGui (64-bit)
Dosya Düzenle Görünüm Diğer Paketler Pencereler Yardım
R
R Console
> x=read.csv("Ulkeler.txt")
> x
   Ulke      D1      D2      D3      D4
1  Brezilya 10296  91.0 23.40 75.2
2  Avustralya 39590  98.0  4.11 79.6
3    Nijerya   831  39.1 96.10 41.9
4    Kuveyt 43159 100.0  4.52 80.7
5      Çin  5311  91.1 24.00 74.0
6  İspanya 13298  96.9 13.30 75.5
7  Arjantin 34111  98.0  5.04 79.2
8  Güney Afrika 10603  82.3 45.10 49.2
9  Avusturya  1000  7.0 92.40 42.1
10 Belçika  5251  84.0 42.10 53.2
11  Gürçistan 4199 101.0 17.41 72.0
12 Bosna Hersek 3321  50.0 67.40 65.7
13 Kırgızistan 2974  62.0 54.00 64.6
14  Türkiye 12876  89.1 27.40 71.9
15  İngiltere 34729  98.0  3.30 81.1
16  Litvanya 19729  99.7  8.30 74.0
17  Slovenya 36979  96.1  5.35 79.6
18     İtalya 26761  96.1  5.35 79.6
19 Portekiz 34101  99.3  3.30 82.5
>

```

Veri setini içinde barındıran *x* matrisi, *ulkeler.variable=x* komutu yardımıyla *ulkeler.variable* matrisine atanır. Daha sonra analizlerde kullanılmayacak olan ve veri matrisinin ilk sütununda bulunan ülke isimlerini veri matrisinden çıkartmak için *ulkeler.variable\$Ulke=NULL* komutu kullanılır. Tekrar *ulkeler.variable* komutu yardımıyla veri matrisi görüntülenecek olursa, ilk sütunda bulunan ülke isimlerinin *ulkeler.variable* matrisinde bulunmadığı görülecektir. Bu işlemler şekil 7.15'te görülmektedir.

Şekil 7.15

R Programında
Ülkeler
Sütununun Veri
Matrisinden
Çıkartılması

```

R
R Console
> ulkeler.variable=x
> ulkeler.variable$Ulke=NULL
> ulkeler.variable
   D1      D2      D3      D4
1 10296  91.0 23.40 75.2
2 39590  98.0  4.11 79.6
3   831  39.1 96.10 41.9
4 43159 100.0  4.52 80.7
5  5311  91.1 24.00 74.0
6 13298  96.9 13.30 75.5
7 34111  98.0  5.04 79.2
8 10603  82.3 45.10 49.2
9  1000  7.0 92.40 42.1
10 5251  84.0 42.10 53.2
11 4199 101.0 17.41 72.0
12 3321  50.0 67.40 65.7
13 2974  62.0 54.00 64.6
14 12876  89.1 27.40 71.9
15 34729  98.0  3.30 81.1
16 19729  99.7  8.30 74.0
17 36979  96.1  5.35 79.6
18 26761  96.1  5.35 79.6
19 34101  99.3  3.30 82.5
>

```

Veri setine k-ortalamalar yönteminin uygulanabilmesi için **results=kmeans(ulkeler.variable,3)** komutu kullanılır. Bu komut yardımıyla **ulkeler.variable** veri matrisi k-ortalamalar yöntemi ile 3 kümeye ayrılr ve sonuçlar ise **results** komutu yardımıyla görüntülenebilir. Bu işlemler Şekil 7.16'da görüntülenmektedir.

Şekil 7.16

k-Ortalamlar Yönteminin Uygulanması ve Sonuçların Görüntülenmesi

```
> results=kmeans(ulkeler.variable,3)
> results
K-means clustering with 3 clusters of sizes 7, 7, 5

Cluster means:
      D1        D2        D3        D4
1 35632.857 98.25714 4.507143 80.52857
2 3269.571 62.02857 56.201429 59.07143
3 13360.400 91.80000 23.500000 69.16000

Clustering vector:
[1] 3 1 2 1 2 3 1 3 2 2 2 2 2 3 1 3 1 1 1

within cluster sum of squares by cluster:
[1] 158303372 20158320 57792915
  (between_SS / total_SS =  94.2 %)

Available components:

[1] "cluster"     "centers"     "totss"       "withinss"    "tot.withinss" "betweenss"   "size"
[8] "iter"         "ifault"
```

Şekil 7.16'da görüldüğü gibi, k-ortalamalar yöntemine göre ülkeler her bir kümede sırasıyla 7, 7 ve 5 ülke bulunan 3 kümeye ayrılmıştır. Elde edilen 3 farklı kümeye ait D1, D2, D3 ve D4 değişkenlerine ilişkin ortalamalar ise **Cluster means** başlığı altında sonuç ekranında verilmiştir. Örneğin D3 değişkeninin ortalaması, **birinci kümede** 4.51, **ikinci kümede** 56.20 ve **üçüncü kümede** ise 23.50 olarak elde edilmiştir. Benzer şekilde değişkenlerin diğer kümelerdeki ortalamaları da yorumlanabilir. **Clustering vector** başlığı incelendiğinde ise ülkelerin veri setindeki giriliş sırasına göre hangi kümelerde yer aldıkları (küme üyeleri) görülecektir. Ayrıca **within cluster sum of squares by cluster** başlığı ile küme içi kareler ortalamaları verilmiştir. Bu arada, "cluster", "centers", "totss", "withinss", "tot.withinss", "betweenss", "size", "iter", "ifault" gibi komutlar yardımıyla sonuçlara ilişkin bileşenlerde görüntülenebilmektedir. Örneğin, **results\$size** komutu sonuçlardan sadece küme boyutlarını verirken ([1] 7 7 5), **results\$cluster** komutu sonuçlardan sadece kümelere dağılımı verir ([1] 3 2 1 2 1 3 2 3 1 1 1 1 1 3 2 3 2 2 2). Bu işlemler Şekil 7.17'de görüntülenmektedir.

Şekil 7.17

```
> results$size
[1] 7 7 5
> results$cluster
[1] 3 1 2 1 2 3 1 3 2 2 2 2 3 1 3 1 1 1
```

results\$size ve
results\$cluster
Komutlarının
Kullanılması

Ülkelerin kümelere göre dağılımı ise **table(x\$Ulke,results\$cluster)** komutu yardımıyla görüntülenmektedir. Bu komutun kullanımı ve elde edilen sonuçlar Şekil 7.18'de verilmiştir.

Şekil 7.18

`table(x$Ulke,
results$cluster)`
Komutunun
Kullanılması

```

RGui (64-bit)
Dosya Düzenle Görünüm Diğer Paketler Pencereler Yardım
R
R Console
> table(x$Ulke, results$cluster)
   1 2 3
Arjantin 0 1 0
Avustralya 0 1 0
Avusturya 1 0 0
Belçika 1 0 0
Bosna Hersek 1 0 0
Brezilya 0 0 1
Çin 1 0 0
Güney Afrika 0 0 1
Gürcistan 1 0 0
İngiltere 0 1 0
İspanya 0 0 1
İtalya 0 1 0
Kirgızistan 1 0 0
Kuveyt 0 1 0
Litvanya 0 0 1
Nijerya 1 0 0
Portekiz 0 1 0
Slovenya 0 1 0
Türkiye 0 0 1
>

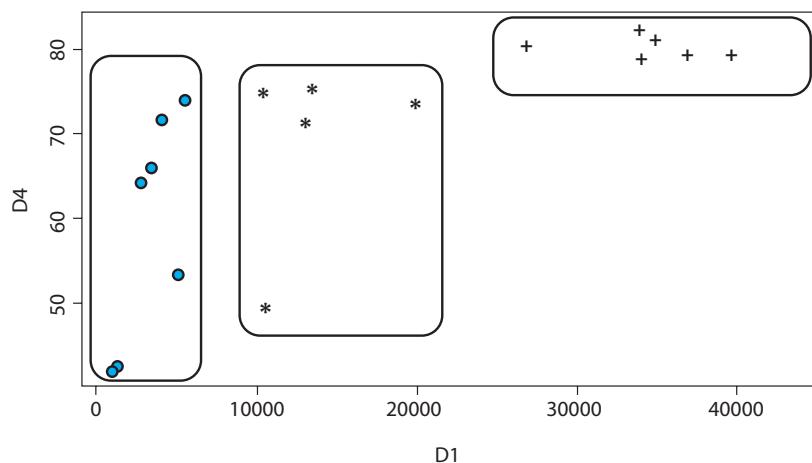
```

Şekil 7.18'de elde edilen sonuçlar incelenecak olursa *Avusturya*, *Belçika*, *Bosna Hersek*, *Çin*, *Gürcistan*, *Kirgızistan*, *Nijerya* gibi ülkeler incelenen dört değişken bakımından birinci kümeye yer almışlardır. *Arjantin*, *Avustralya*, *İngiltere*, *İtalya*, *Kuveyt*, *Portekiz*, *Slovenya* farklı uzaklıklarda ikinci kümeyi oluşturmuşlardır. *Brezilya*, *Güney Afrika*, *İspanya*, *Litvanya*, *Türkiye* ise üçüncü kümeye birleşmişlerdir.

Rda Eğer D1 ve D4 değişkenlerine göre, kümelere ait grafik çizdirilmek istenirse `plot(x[c("D1","D4")], col=results$cluster)` komutu kullanılır. Grafik Şekil 7.19'da verilmiştir. Burada kümelerdeki birimler sırasıyla “o”, “*” ve “+” işaretleri ile gösterilmiş ve kolay anlaşılmasına için ise kutucuklar içerisinde alınmıştır.

Şekil 7.19

Veri Setindeki D1 ve
D4 Değişkenlerine
İlişkin Çizdirilen
Grafik



SIRA SIZDE



R programını kullanarak, Örnek 2'deki D2 ve D3 değişkenleri üzerinde K-ortalamalar yöntemi uygulayınız, sonuçları elde ediniz ve yorumlayınız.

5

Özet



Kümeleme analizinin kullanım alanlarını tanımlamak

Veri madenciliği içerisinde kümeleme analizinin büyük bir yeri ve önemi bulunmaktadır. Veri setinde bulunan değişkenleri veya birimleri belli benzerlik ve uzaklık ölçütleri yardımıyla kümelere ayırmak, alt toplumları belirlemek araştırmacılara birçok yönden avantaj sağlamaktadır. Bu yöntemde veri setinde oluşturulan gruplarda, grup içi değişimin en az, gruplararası değişimin en fazla olacağı bir yapı ortaya koymak amaçlanmaktadır. Tip, ziraat, psikoloji, sosyoloji, arkeoloji gibi gözlemlerin sınıflandırılmasına ihtiyaç duyulan pek çok bilim dalının yararlandığı sıklıkla uygulamalarda kullanılan bir yöntemdir.



Çok değişkenli analizler ile kümeleme analizi arasındaki ilişkileri karşılaştırmak

Kullanımı yaygın olan çok değişkenli istatistiksel analizler ile kümeleme analizi arasındaki benzerlikler ve farklılıklar bölüm içerisinde ayrıntılı olarak ele alınmıştır. Gerek varsayımları gerekse uygulamadaki farklılıklar nedeniyle kümeleme analizi, kendi içlerinde homojen ve kendi aralarında heterojen kümelerin oluşmasında oldukça yaygın kullanımı olan birçok değişkenli analiz yöntemidir.



Benzerlik ya da uzaklık matrislerini belirlemek

Veri setinden yararlanılarak benzerlik ya da uzaklık matrislerinin elde edilmesi Ünite 4'te ayrıntılı olarak ele alınmıştır. Kümeleme analizinde bu matrislerin hesaplanması, elde edilen bu matrisler yardımıyla birim ya da değişkenlerin kümelenme uygulamalarının yapılması ve sonuçların yorumlanması R paket programı yardımıyla ayrıntılı olarak verilmiş ve sonuçlar yorumlanmıştır.



Aşamalı ve Aşamalı olmayan kümeleme yöntemlerini açıklamak

Kümeleme analizi; uzaklık ya da benzerlik matrisinden yararlanarak birimleri ya da değişkenleri kendi içinde homojen ve kendi aralarında ise heterojen kümelere ayırmakta kullanılmaktadır. Bu işlemleri yaparken kümeleri ortaya koymada izledikleri yaklaşımına göre iki temel alt gruba ayırlırlar. Bunlar; Aşamalı kümeleme yöntemleri ve Aşamalı olmayan kümeleme yöntemleri olarak ele alınmaktadır. Her iki yönteminde ortak amacı kümeler arasındaki farklılıklarını ve kümeler içi benzerlikleri en yüksek düzeye çıkarmaktır. Yani, küme içi homojenlik artarken kümeler arası homojenlik ise azalmaktadır. Hangi tekninin kullanılacağı küme sayısına bağlıdır fakat her uygulamada iki tekniğin beraber kullanılması çok daha yararlı olabilmektedir.



R paket programında kümeleme analizini uygulamak ve sonuçlarını yorumlamak

Elde edilen veri setlerinde kümeleme analizi uygulamalarının yapılması ve yorumlanması hakkında bilgi ve beceri kazanmak araştırmacıların işlerini oldukça kolaylaştırmaktadır. Bu işlem için bilişim teknolojilerinin kullanımı oldukça önemlidir. Bilgisayar ve konu ile ilgili yazılım kullanımını bilmek, verileri analiz etmek, sonuçları yorumlayabilmek ve rapor hâline getirebilmek hem zaman hem de kaynak kullanımını bakımından uygulamacılara avantaj sağlamaktadır.

Kendimizi Sınayalım

- 1.** Aşağıdakilerden hangisi Birleştirici aşamalı kümeleme yöntemlerinden biri **değildir**?
 - a. Tek Bağlantı Kümeleme Yöntemi
 - b. Ortalama Bağlantı Kümeleme Yöntemi
 - c. Tam Bağlantı Kümeleme Yöntemi
 - d. McQuitty Bağlantı Kümeleme Yöntemi
 - e. k-ortalamalar Yöntemi

- 2.** Veri setine Kümeleme analizinin uygulanması sonucunda elde edilen **dendrogramın** R programında görüntülenmesi için aşağıdaki komutlardan hangisi kullanılmalıdır?
 - a. `>plot(h,labels=x$Ülke)`
 - b. `>x=read.csv("c:/ulkeler.txt")`
 - c. `>table(x$Ülke,results$cluster)`
 - d. `>results$size`
 - e. `>clusters=cutree(h, k=3)`

- 3.** Aşağıdakilerden hangisi aşamalı olmayan kümeleme yöntemlerinden biridir?
 - a. Ortalama Bağlantı Kümeleme Yöntemi
 - b. k-ortalamalar Yöntemi
 - c. Tam Bağlantı Kümeleme Yöntemi
 - d. McQuitty Bağlantı Kümeleme Yöntemi
 - e. Tek Bağlantı Kümeleme Yöntemi

- 4.** Özellikle değişkenlerin birbirinden bağımsız olmadığı ve değişkenler arasında korelasyon olduğu durumlarda hangi aşamalı olmayan kümeleme yöntemi veri setini grüplamada (kümelemede) başarılı olmaktadır?
 - a. Ortalama Bağlantı Kümeleme Yöntemi
 - b. k-ortalamalar Yöntemi
 - c. k-medoidler Yöntemi
 - d. k-medoidler Yöntemi
 - e. Tek Bağlantı Kümeleme Yöntemi

- 5.** “**ulkeler.variable**” olarak adlandırılan ve 3 kümeye ayrılması istenilen veri setine K-ortalamalar kümeleme yönteminin uygulanabilmesi için aşağıdaki komutlardan hangisi kullanılmalıdır?
 - a. `>results=kmeans(ulkeler.variable,4)`
 - b. `>results=kmeans(ulkeler.variable,3)`
 - c. `>results= ulkeler.variable (kmeans,3)`
 - d. `>results= ulkeler.variable (kmeans,3/3)`
 - e. `>results=kmeans(ulkeler.variable,3/nonhier)`

- 6.** Aşamalı olmayan kümeleme yöntemlerinden hangisi değişkenlerin ortalama vektörlerini küme merkezi olarak ele alır ve kümeleme süreci bunun etrafında şekillenir?
 - a. Ortalama Bağlantı Kümeleme Yöntemi
 - b. k-ortalamalar Yöntemi
 - c. k-medyanlar Yöntemi
 - d. k-medoidler Yöntemi
 - e. Tek Bağlantı Kümeleme Yöntemi

- 7.** Minimum varyans yöntemi olarak da bilinen, küme bağlantılarından çok küme içi kareler toplamını dikkate alan ve küme içi varyansları minimize ederek kümeler oluşturma amacıyla sahip *aşamalı kümeleme yöntemi* aşağıdakilerden hangisidir?
 - a. Ortalama Bağlantı Kümeleme Yöntemi
 - b. Tam Bağlantı Kümeleme Yöntemi
 - c. Ward Bağlantı Kümeleme Yöntemi
 - d. McQuitty Bağlantı Kümeleme Yöntemi
 - e. Tek Bağlantı Kümeleme Yöntemi

- 8.** En uzak komşu niteliğine sahip birimleri birbirleriyle birleştirerek küme oluşturmayı hedefleyen, Maksimum Yön tem, Sıralama Tip Analizi, En Uzak Komşu Analizi, Çap Yöntemi gibi isimlerle de anılan aşamalı kümeleme yöntemi aşağıdakilerden hangisidir?
 - a. Ortalama Bağlantı Kümeleme Yöntemi
 - b. Tam Bağlantı Kümeleme Yöntemi
 - c. Ward Bağlantı Kümeleme Yöntemi
 - d. McQuitty Bağlantı Kümeleme Yöntemi
 - e. Tek Bağlantı Kümeleme Yöntemi

- 9.** Aşağıdakilerden hangisinde, **x** veri setindeki birimler arası uzaklıklar Öklid yöntemine göre hesaplanır ve **dist.x** matrisine atanır?
 - a. `>dist.x=dist(euclidean)`
 - b. `>dist.x=dist(x,method)`
 - c. `>dist.x=dist(x,method="euclidean")`
 - d. `>dist.x=dist(method="euclidean")`
 - e. `>dist.x=dist(x,method="euclidean")`

- 10.** Veri setindeki değişkenlerin asimetrik olduğu durumlarda medyan değerlerine ait vektörleri küme merkezi olarak kullanan aşamalı olmayan kümeleme yöntemi aşağıdakilerden hangisidir?
 - a. Tek Bağlantı Kümeleme Yöntemi
 - b. k-ortalamalar Yöntemi
 - c. Ortalama Bağlantı Kümeleme Yöntemi
 - d. k-medyanlar Yöntemi
 - e. k-medoidler Yöntemi

Kendimizi Sınavalım Yanıt Anahtarı

1. e Yanınız yanlış ise “Aşamalı Kümeleme Yöntemleri” bölümünü yeniden gözden geçiriniz.
2. a Yanınız yanlış ise “R Programında Tek Bağlantı Kümeleme Yöntemi uygulaması” bölümünü yeniden gözden geçiriniz.
3. b Yanınız yanlış ise “Aşamalı Olmayan Kümeleme Yöntemleri” bölümünü yeniden gözden geçiriniz.
4. d Yanınız yanlış ise “Aşamalı Olmayan Kümeleme Yöntemleri” bölümünü yeniden gözden geçiriniz.
5. b Yanınız yanlış ise “Aşamalı Olmayan Kümeleme Yöntemleri” bölümünü yeniden gözden geçiriniz.
6. b Yanınız yanlış ise “Aşamalı Olmayan Kümeleme Yöntemleri” bölümünü yeniden gözden geçiriniz.
7. c Yanınız yanlış ise “Aşamalı Kümeleme Yöntemleri” bölümünü yeniden gözden geçiriniz.
8. b Yanınız yanlış ise “Aşamalı Kümeleme Yöntemleri” bölümünü yeniden gözden geçiriniz.
9. e Yanınız yanlış ise “Aşamalı Kümeleme Yöntemleri” bölümünü yeniden gözden geçiriniz.
10. d Yanınız yanlış ise “Aşamalı Olmayan Kümeleme Yöntemleri” bölümünü yeniden gözden geçiriniz.

Sıra Sizde 4

Aşamalı kümeleme yöntemlerinden birleştirici kümeleme yöntemleri aşağıda sıralanmıştır

- Tek Bağlantı Kümeleme Yöntemi
- Tam Bağlantı Kümeleme Yöntemi
- Ortalama Bağlantı Kümeleme Yöntemi
- McQuitty Bağlantı Kümeleme Yöntemi
- Küresel Ortalama Bağlantı Kümeleme Yöntemi
- Medyan Bağlantı Kümeleme Yöntemi
- Ward Bağlantı Kümeleme Yöntemi

Sıra Sizde 5

Sırası ile, aşağıdaki komutlar aracılığı ile çözüme ulaşılmaktadır.

```
> x=read.csv("c:/ulkelerD2D3.txt")
> ulkeler.variable=x
> ulkeler.variable
> results=kmeans(ulkeler.variable,3)
> results
> table(x$ulke,results$cluster)
```

Şekil 7.20: k-Ortalamlar Yönteminin Uygulanması ve Sonuçların Görüntülenmesi

K-means clustering with 3 clusters of sizes 10, 6, 3

Cluster means:

D2	D3
1 98.31000	6.998
2 83.25000	36.000
3 32.03333	85.300

Clustering vector:

```
[1] 2 1 3 1 2 1 1 2 3 2 1 3 2 2 1 1 1 1 1
```

Within cluster sum of squares by cluster:

```
[1] 225.9662 1429.6750 1486.8667
```

(between_SS / total_SS = 88.7 %)

Available components:

```
[1] "cluster"  "centers" "totss"  "withinss" "tot.withinss"
[6] "betweenss" "size"    "iter"   "ifault"
```

Sıra Sizde Yanıt Anahtarı

Sıra Sizde 1

Bu kümeleme yöntemi, değişkenlerin ortalama vektörleri ni küme merkezi olarak ele alır ve kümeleme süreci bunun etrafında şekillenir. Ayrıca bu yöntem, veri setinde bulunan birimleri “küme içi kareler toplamları minimum” olacak biçimde k sayıda kümeye ayırmayı amaçlar.

Sıra Sizde 2

Kümeleme Analizi, diğer bir çok değişkenli analiz yöntemi olan Diskriminant Analizinde olduğu gibi tahmin amaçlı kullanılmamakta Faktör Analizinde olduğu gibi de varsayımları bulunmamaktadır.

Sıra Sizde 3

Kümeleme analizinde birim ya da değişkenler arasındaki uzaklıklar hesaplamak için en sık kullanılan uzaklık ölçüsü Öklid uzaklığıdır. Öklid uzaklığının iki obje arasındaki benzerliği ölçümede en yaygın kullanılan uzaklık ölçüsü olup iki obje arasına çizilecek bir doğrunun uzunluğunu temel alır.

Ülkeler her bir kümede sırasıyla 10, 6, 3 ülke bulunan 3 kümeye ayrılmıştır. Elde edilen 3 farklı kümeye ait D2 ve D3 değişkenlerine ait ortalamalar Cluster means başlığı altında verilmiştir. D2 değişkeninin ortalaması birinci kümeye 98.3, ikinci kümeye 83.25, üçüncü kümeye 32.03 olarak elde edilmiştir. Ülkelerin kümelere göre dağılımı ise aşağıdaki gibi görülmektedir. İlk küme Arjantin, Avustralya, Gürcistan, Ingiltere, İspanya, İtalya, Kuveyt, Litvanya, Portekiz, Slovenya, ikinci küme, Belçika, Brezilya, Çin, Güney Afrika, Kırgızistan ve Türkiye, üçüncü küme ise Avusturya, Bosna Hersek, Nijerya ülkelerinden oluşmaktadır (Şekil 7.21).

Şekil 7.21: k-Ortalamalar Yönteminin Uygulanması ve Sonuçların Görüntülenmesi

```
RGui (64-bit)
Dosya Düzenele Görünüm Diğer Paketler Pencereler Yardım
R Console
> table(x$Ulke, results$cluster)

      1 2 3
Arjantin    0 1 0
Avustralya   0 1 0
Avusturya    0 0 1
Belçika      1 0 0
Bosna Hersek 0 0 1
Brezilya     1 0 0
Çin          1 0 0
Güney Afrika 1 0 0
Gürçistan    0 1 0
İngiltere    0 1 0
İspanya      0 1 0
İtalya        0 1 0
Kirgızistan  1 0 0
Kuveyt       0 1 0
Litvanya     0 1 0
Nijerya      0 0 1
Portekiz     0 1 0
Slovenya     0 1 0
Türkiye      1 0 0
> |
```

Yararlanılan ve Başvurulabilecek Kaynaklar

- Alpar, R. (2003) *Uygulamalı çok değişkenli istatistiksel yöntemlere giriş 1*, Nobel Yayın Dağıtım.
- Atbaş, A., & Günay, C. (2008) *Kümeleme analizinde küme sayısının belirlenmesi üzerine bir çalışma*, Yayınlanmamış Yüksek Lisans Tezi, Ankara: Ankara Üniversitesi Fen Bilimleri Enstitüsü.
- Casgrain, P., & Legendre, P. (2001) *The R package for multivariate and spatial analysis*, Version, 4, d5.
- Çelik, Ş. (2013). *Kümeleme analizi ile sağlık göstergelerine göre Türkiye'deki illerin sınıflandırılması*, Doğuş Üniversitesi Dergisi, 14(2).
- Everitt, B. S., Landau, S., & Leesse, M. (2001). *Cluster Analysis* Arnold. A member of the Hodder Headline Group, London.
- Feil, B., Balasko, B., & Abonyi, J. (2007). *Visualization of fuzzy clusters by fuzzy Sammon mapping projection: application to the analysis of phase space trajectories*, Soft Computing, 11(5).
- Green, P. E., Krieger, A. M., & Wind, Y. (2001). *Thirty years of conjoint analysis: Reflections and prospects*, Interfaces, 31(3_supplement).
- Hamarat, B., Bal, C., Özdamar, K. (1999) *Ülkelerin Sağlık Göstergeleri Bakımından Gelişmişlik Düzeylerinin Belirlenmesi*, 1. İstatistik Kongresi, 5-9 Mayıs 1999, Belek, Antalya, Bildiriler Kitabı 182-185.
- Johnson, R. A., & Wichern, D. W. (1988) *Multivariate statistics, a practical approach*
- Özdamar, K. (1999) *Paket programlar ile istatistiksel veri analizi*, Kaan Kitabevi, Eskişehir.
- R Core Team (2015) *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. URL: <http://www.R-project.org/>
- Tatlıdil, H. (1996) *Uygulamalı çok değişkenli istatistiksel analiz*, Cem Web Ofset, Ankara, 329.
- Tekin, B. *Temel Sağlık Göstergeleri Açısından Türkiye'deki İllerin Gruplandırılması: Bir Kümeleme Analizi Uygulaması*.
- Yılmaz, Ş. K., & Patır, S. (2011) *Kümeleme Analizi ve Pazarlamada Kullanımı*, Akademik yaklaşım dergisi, 2(1).

8

Amaçlarımız

Bu üniteyi tamamladıktan sonra;

- 🕒 Web madenciliği ile yararlı bilgi keşfi sürecini açıklayabilecek,
- 🕒 Web madenciliğinde kullanılan veri türlerini sınıflandırabilecek,
- 🕒 Veri türüne göre Web madenciliğini sınıflandırabilecek,
- 🕒 Sosyal medya verilerini analiz edebilecek bilgi ve becerilere sahip olabileceksiniz.

Anahtar Kavramlar

- İçerik
- Yapı
- Kullanım
- Veritabanı
- Örüntü
- Sosyal Medya
- Twitter
- Facebook

İçindekiler

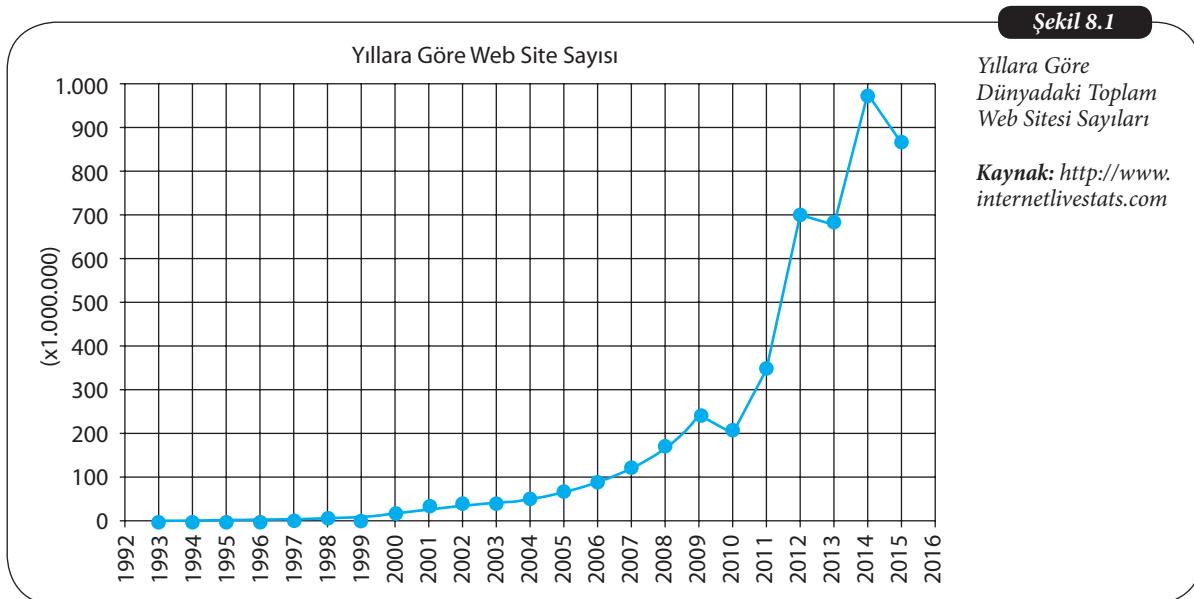


Web Madenciliği ve Sosyal Medya Madenciliği

GİRİŞ

Internet, günümüzde gerek kişisel gerek kurumsal olsun bilgiye ulaşmanın en etkin ve hesaplı araçlarından bir tanesi olarak karşımıza çıkmaktadır. Şekil 8.1 ve Şekil 8.2'den de görülebileceği gibi 2000'li yılların başından itibaren bilişim teknolojilerindeki yükselişin ivme kazanması, buna paralel olarak internete erişim yollarının çeşitliliğindeki artış ve erişim kolaylığı sayesinde internet hayatımızın vazgeçilmez bir parçası olmuştur. Başta iletişim olmak üzere, e-ticaret, bankacılık işlemleri, reklam, kurumsal işlemler ve eğitim gibi birçok işlem internet üzerinden gerçekleştirilmektedir. Dolayısıyla insanlar internete her erişimlerinde metin, ses, görüntü vb. bir takım bilgiler bırakmakta ve böylece sanal ortamda çok büyük bilgi ve veri yığınları oluşmaktadır. İnternet ortamındaki verilerin yapılandırılmamış olması, boyutlarının aşırı büyük olması, dinamik ve düzensiz bir yapı içinde olması web madenciliğinin önemini her geçen gün biraz daha artttırmaktadır.

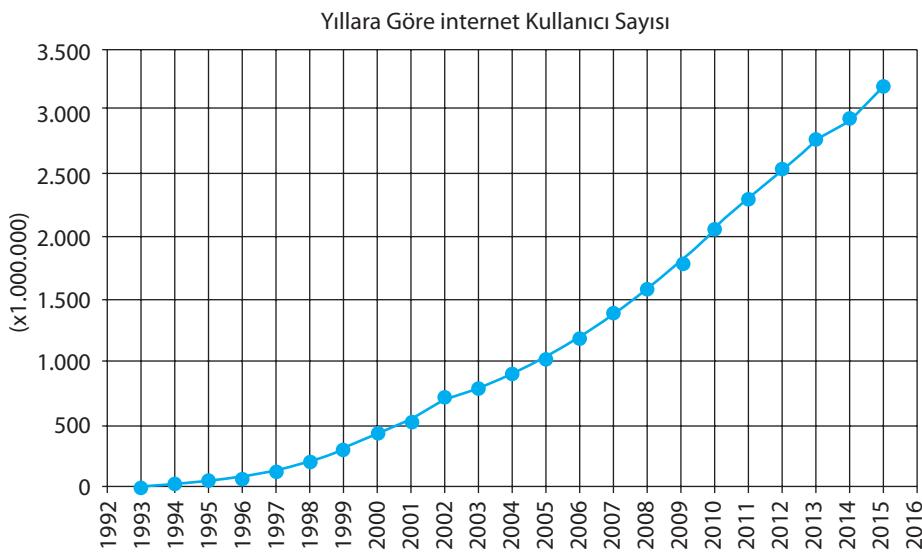
Şekil 8.1



Şekil 8.2

*Yıllara Göre
Dünyadaki Toplam
İnternet Kullanıcı
Sayları*

Kaynak: <http://www.internetlivestats.com>



VERİ MADENCİLİĞİ VE WEB MADENCİLİĞİ

Veri madenciliği, veri mühendisliği ve bilgi keşfi alanlarında son zamanlarda oldukça günceldir. Temelde veri madenciliği, e-ticaret uygulamalarındaki işlem verileri ya da biyoinformatik alanında genetik ifadeler gibi farklı veri türlerinden oluşan büyük miktarda veri yiğini içerisinde, anlamlı ve yararlı bilginin ortaya çıkarılması anlamına gelir. Verinin türü ne olursa olsun veri madenciliğinin temel amacı, mevcut veri yiğini içerisindeki gizli veya daha önce farkedilmemiş bilginin açığa çıkarılması yani keşfedilmesidir. İlişki kuralları, sıralı örüntü madenciliği, denetimli ve denetimsiz öğrenme algoritmaları son yıllarda yaygın olarak kullanılan ve üzerinde yoğun araştırmalar yapılan veri madenciliği alanlarıdır.

Günümüzde veri madenciliği hem akademik olarak hem de sektörde giderek artan bir ilgi görmekte ve bu alanda sağlanan gelişmeler birçok uygulamada hayat bulmaktadır. Son on yılda veri madenciliği, web belgeleri de dahil olmak üzere web nesneleri, web bağlantı yapısı, web kullanıcı işlemleri, web semantik vb. gibi birçok web veri yönetimi araştırmalarında başarılı bir şekilde uygulanmaktadır. Çeşitli web verilerinden elde edilen anlamlı ve yararlı bilgiler, web nesneleri arasındaki ilişkileri anlamamızı ve keşfetmemizi sağlamakta ve web veri yönetiminin geliştirilmesi için kullanılmaktadır.

Web, kapsamlı veri madenciliği veya otomatik öğrenme yaklaşımları aracılığıyla tespit edilebilen çok miktarda gizli kalmış yararlı bilginin yanısına, çeşitli veri tiplerini barındıran büyük bir veri deposu ve kaynağıdır. Web madenciliği ise, web dokümanlarından bilginin ayıklanması veya keşfedilmesini sağlayan bir veri madenciliği teknigidir. Web madenciliğinde kullanılan yöntemler sayısal zeka olarak da bilinen ve genel olarak **veritabanı**, veri madenciliği, otomatik öğrenme ve bilgi çıkarsama vb. gibi geniş bir uygulama alanına sahip akıllı hesaplama yaklaşımlarına dayanmaktadır. Web madenciliği kullanıcıların aradıkları cevaba hızlı ve doğru bir şekilde ulaşabildikleri devrim niteliğinde bir süreçtir. Web madenciliğinin yıllar itibarıyle gelişimi Tablo 8.1'de sunulmuştur.

Veritabanı, büyük miktardaki bilgileri depolamada yetersiz kalan dosya-ismen sistemine alternatif olarak geliştirilen ve birbirleriyle ilişkili bilgilerin depolandığı alanlardır.

Adımsal Gelişim	Kullanılan Teknoloji	Özellikler
Veri toplama (1960'lar)	<ul style="list-style-type: none"> Bilgisayar Manyetik bantlar Diskler 	<ul style="list-style-type: none"> Geçmişe dönük Statik veri iletimi
Veri erişimi (1980'ler)	<ul style="list-style-type: none"> İlişkisel veritabanı (RDBMS) Yapı sorgu dili (SQL) Açık veritabanı bağlantıları (ODBC) 	<ul style="list-style-type: none"> Geçmişe dönük Rekor seviyede dinamik veri iletimi
Veri ambarı & Karar destek (1990'lar)	<ul style="list-style-type: none"> On-Line Analistik İşleme (OLAP) Çok boyutlu veritabanları Veri ambarları 	<ul style="list-style-type: none"> Geçmişe dönük Çoklu seviyelerde dinamik veri iletimi
Veri Madenciliği (2000'ler)	<ul style="list-style-type: none"> Gelişmiş algoritmalar Çok işlemcili bilgisayarlar Çok büyük veritabanları 	<ul style="list-style-type: none"> İleriye dönük Tahmin amaçlı bilgi iletimi
Web Madenciliği (Günümüzde)	<ul style="list-style-type: none"> WWW Internet Devasa ölçekli veritabanı 	<ul style="list-style-type: none"> Güçlü, ekonomik değeri olan, çoklu madencilik fonksiyonlarını hızlı ve verimli kullanan ilişkisel veritabanları Büyük veri ambarı madencilik araçları

Tablo 8.1
Web Madenciliğinin Evrimsel Gelişimi

Kaynak: Sharma K., Shrivastava G., Kumar V. (2011). Web mining: Today and tomorrow, 3rd International Conference on Electronics Computer Technology, 399-403.

Çeşitli kaynaklarda veri veya bilgi keşfi olarak da adlandırılan veri madenciliği, veritabanlarındaki gizli bilgi veya örüntülerin keşfedilmesi ve yararlı bilgi şeklinde özetlenmesinden oluşan bir süreçtir. Veri madenciliği aynı zamanda birbirile ilişkili büyük veritabanları arasındaki çok sayıdaki ilişkileri de ortaya çıkarmak için kullanılan bir süreçtir. Buna karşılık web madenciliği, web ortamındaki dokümanları tekrar geri elde etmek için kullanılan bir tekniktir. Veri madenciliği,

- Verinin elde edilmesi
 - Verinin saklanması ve yönetimi
 - Veri erişiminin sağlanması
 - Verinin analiz edilmesi
 - Analiz sonuçlarının anlaşılır bir biçimde sunulması
- temel adımlarından oluşan bir süreçtir. Veri madenciliği ile web madenciliğini farklı açılardan karşılaştırması Tablo 8.2'de verilmiştir.

Veri ambarı, veritabanı üzerindeki yükü hafifletmek için oluşturulmuş, birbirile ilişkili veriler kolay, hızlı ve doğru bir biçimde sorgulama ve analiz yapabilmek için gerekli işlemlerin yapılabildiği bir veri deposudur.

Tablo 8.2
Veri Madenciliği ve
Web Madenciliği
Karşılaştırması

Kaynak: Kaur S., Kaur K. (2015). Web Mining and Data Mining: A Comparative Approach, International Journal of Novel Research in Computer Science and Software Engineering Vol. 2(1), 36-42.

	Web Madenciliği	Veri Madenciliği
Amaç	<ul style="list-style-type: none"> Web belgelerinden bilgi çıkarmak. 	<ul style="list-style-type: none"> Veritabanından gizli bilgilerin keşfedilmesi.
Ölçek	<ul style="list-style-type: none"> Sunucu veritabanı 10 milyon iş içermesine rağmen işleme süreci kısadır. 	<ul style="list-style-type: none"> Veritabanı 1 milyon iş içerir ancak işleme süreci uzundur.
Yapı	<ul style="list-style-type: none"> Bilgiler yapılandırılmış, yarı yapılandırılmış ve yapılandırmamış web formlarından elde edildiği için geniş bir veritabanından bilgi sağlanır. 	<ul style="list-style-type: none"> Belirli yapıya sahip verilerden bilgi elde edildiğinden web madenciliğine kıyasla geniş bir veritabanından tüm bilginin elde edilmesi mümkün değildir.
Erişim	<ul style="list-style-type: none"> Veri gizli değildir. Sadece kayıt dosyalarına erişebilmek için izin gereklidir. 	<ul style="list-style-type: none"> Veri kişisel ve gizlidir. Ancak yetkili kullanıcı tarafından erişilebilir.
Veri	<ul style="list-style-type: none"> Çevrimiçi veriler kullanılır. 	<ul style="list-style-type: none"> Çevrimdışı veriler kullanılır.
Veri Depolama	<ul style="list-style-type: none"> Veriler, sunucu günlükleri ve web sunucusu veritabanında saklanır. 	<ul style="list-style-type: none"> Veriler, veri ambarlarında saklanır.
Uygulama Alanları	<ul style="list-style-type: none"> E-öğrenme Dijital Kütüphaneler E-Devlet Elektronik Ticaret E-Siyaset E-Demokrasi Güvenlik ve Suç Soruşturması vb. 	<ul style="list-style-type: none"> Bankacılık Pazarlama İmalat Sağlık Sigorta Hukuk Hava yolları Bilgisayar donanımı ve yazılımı Hükümet ve savunma vb.
Dezavantajları	<ul style="list-style-type: none"> URL'ler izlenerek veriye erişilebilir. Olaylar ve URL'ler çok çeşitlidir. Verilerin büyük bir kısmı kullanılmadan kalır. 	<ul style="list-style-type: none"> Gizlilik sorunları Güvenlik sorunları Bilginin kötüye kullanımı Eksik bilgilendirmeler
Zorluklar	<ul style="list-style-type: none"> Web sayfalarının karmaşıklığı Webin büyülüğu Bilginin bağlantısı Bilginin dinamikliği Kullanıcı iletişimiminin çeşitliliği 	<ul style="list-style-type: none"> Ağ ayarları Veri kalitesi Gizliliğin korunması Ölçeklenebilirlik Karmaşık ve heterojen veri
Teknikler	<ul style="list-style-type: none"> Web içerik madenciliği Web yapı madenciliği Web kullanım madenciliği 	<ul style="list-style-type: none"> Yapay sinir ağları Karar ağaçları Ilişki kuralları En yakın komşu yöntemi

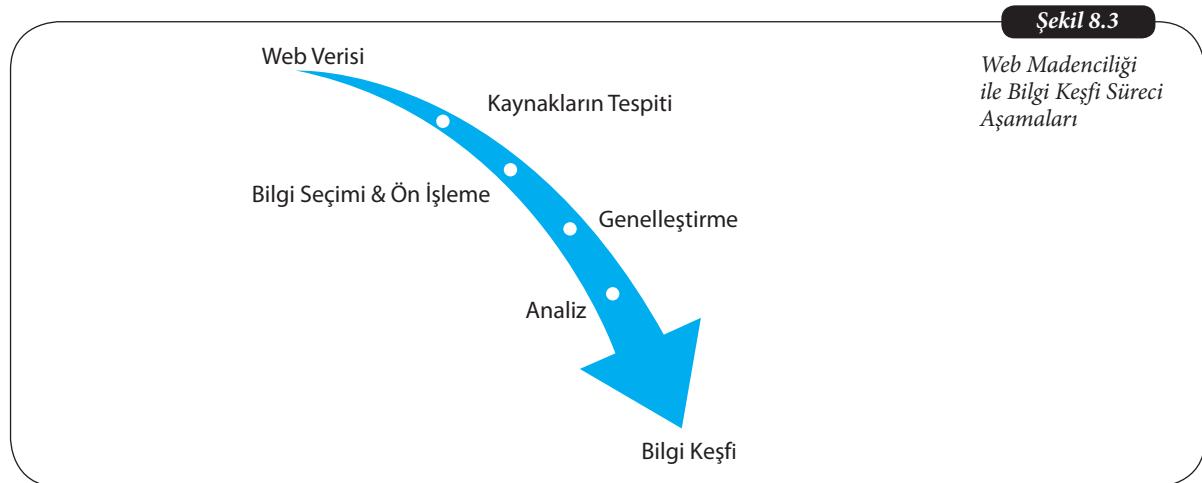
WEB MADENCİLİĞİ SÜRECİ

Web madenciliği web veri yönetimi kapsamında akıllı hesaplama tekniklerinden birisidir. Genel olarak web madenciliği, webdeki veri yiğinları içerisindeki veri madenciliği yöntemleri aracılığı ile yararlı bilgilerin ayıklaması ve sonuç çıkarılması işlemleri olarak tanımlanabilir. Web madenciliği araştırmaları özellikle veri madenciliği, bilginin keşfi ve otomatik öğrenme (machine learning) alanlarının yanısıra veritabanı yönetimi, bilgi erişimi ve yapay zeka vb. gibi alanlarda çalışan birçok akademisyen ve mühendisin ilgisini çekmektedir.

Internet ortamından yararlı bilginin keşfi için, web madenciliği sürecini dört temel adımda ele alabiliriz.

1. *Kaynakların Tespiti*: İlgilenilen konuda bilgi içeren web dokümanlarının belirlenmesi ve elde edilmesi.
2. *Bilgi Seçimi ve Ön İşleme*: Elde edilen kaynaklardan ihtiyaç duyulan bilginin otomatik olarak seçilmesi ve kullanılabilir hâle getirilmesi.
3. *Genelleştirme*: Bireysel web sitelerindeki örüntü (pattern) veya kuralların otomatik olarak çıkarılması ve diğer web siteleri ile karşılaştırarak genellenmesi.
4. *Analiz*: Elde edilen genel örüntü veya kuralların doğruluklarının onaylanması ve / veya yorumlanması.

Şekil 8.3



Webden yararlı bilgilerin keşfi sürecinde izlenecek temel adımlar nelerdir? Kısaca açıklayınız.



SIRA SİZDE

İnternetteki mevcut bilginin günden güne katlanarak artan bir hızla büyümesiyle web, birçok yararlı bilginin keşfine imkan vermesinin yanısıra bilgileri saklamak, yarmak ve almak için güçlü bir platform hâline gelmiştir. Doğası gereği web verilerinin aşırı büyük, dağınık, dinamik ve yapılandırılmış olması nedeniyle web madenciliği çalışmalarında birçok zorlukla karşılaşılmaktadır. Webden bilgi çıkarım uygulamalarında genel olarak karşılaşılan sorunlar biçimde açıklanabilir.

1. *Araştırılan konuyu bulma*: Webde belirli bir bilgiyi bulmak için, genellikle ya doğrudan web belgeleri taranır ya da bir arama motoru kullanılır. Bilgiye erişim amacıyla arama motoru kullanıldığında, araştırılan konuya ilişkin bir ya da birkaç anahtar kelime girilir ve girilen kelime(ler)le ilişkili sayfalar sıralanır. Sorgu tabanlı internet taramalarının iki ana sorunu vardır. Birincisi konuya alakasız birçok sayfanın sıralanmasına neden olan düşük hassasiyet, ikincisi ise web sayfalarının indekslenme kapasitesinin azlığından kaynaklanan düşük duyarlılıktır. Sorgu ile daha çok ilişkili sayfaların nasıl bulunacağı son yılların popüler konuları arasındadır.
2. *İstenilen bilgiyi bulma*: Arama motorları çoğunlukla bir ya da birkaç kelime üzerinden aramayı gerçekleştirir. Bazen bu kelime(ler) içerisinde eş sesli (sesteş) kelimelerin olmasından dolayı araştırılan konunun dışında sonuçlarla karşılaşılır. Yani kelimenin bütün içerisindeki anlamı çoğunlukla dikkate alınmaz.

3. **Yararlı bilgi keşfi:** Geleneksel web arama servislerinde, anahtar kelime(ler)e uyan sonuçlar kullanıcıya sıralı liste hâlinde sunulur. Çoğu zaman, web sayfalarının sadece sorgulanın kelime(ler)i içerip içermemesi değil aynı zamanda bu sayfaların konu hakkında içerdikleri yararlı bilgi miktarı önemlidir. Son zamanlarda bilgi keşfi ve karar verme temeline dayalı olarak webden nasıl yararlanılması gerekiği konusunda birçok çalışma yapılmaktadır.
4. **Bilgiyi kişiselleştirme:** Kullanıcıların internet gezinme alışkanlıklarının kişiden kişiye değişiklik göstermesinden dolayı internette sunulan bilgilerin görsellik ve içerik bakımından farklı şekillerde olması gerekmektedir. Dolayısıyla web tasarımcı veya geliştiricilerinin, web sitesini daha kaliteli bir hâle getirmek, kullanıcı oranını arttmak ve kullanıcıdan kullanıcıya değişebilen, kişiye özel web tasarımları sunabilmek için kullanıcıların tercihlerini ve gezinme alışkanlıklarını bilmeleri gerekmektedir.
5. **Web toplulukları ve sosyal ağlar:** Veritabanı yönetim sistemlerindeki klasik veri şemasının aksine, web nesneleri tamamen farklı özelliklerdir ve farklı yönetim stratejisi gerektirirler. Web nesneleri arasında kurulmuş olan ilişkiler oldukça önemlidir. Bu ilişkiler, web nesneleri düğümleri ve linkler ise bu nesneler arasındaki bağlantıları gösterecek şekilde grafiksel olarak ifade edilebilir. Dolayısıyla web verisini çözümleyebilmek için web toplulukları tasarlanabilir ve hatta bazı sosyal ağ uygulamaları için bu tür tasarımlar yapılabilir.

WEB MADENCİLİĞİ VERİ KAYNAKLARI

Web madenciliğinde kullanılabilecek veriler genel olarak, **sunucu** (server), **istemci** (client) ve **vekil** (proxy) sunucu gibi farklı kaynaklardan elde edilirler. Farklı kaynaklardan elde edilecek veriler de farklı yapılara sahip olmaktadır. Web madenciliğinde kullanılan verileri dört başlıkta inceleyebiliriz.

1. **İçerik verisi:** Web sayfalarında kullanıcının erişimine sunulan verilerdir. Bunlar şekil, resim, grafik, görüntü ve ses dosyaları gibi gerçek verilerin yanı sıra, tanımlayıcı kelimeler, etiketler ve doküman özellikleri gibi verilerden oluşmaktadır. İçerik verisi düz metin gibi yapılandırılmamış, HTML dokümanları gibi yarı yapılandırılmış veya veritabanlarından elde edilen veriler şeklinde yapılandırılmış verileri içerir.
2. **Yapı verisi:** Bir web sitesinin içeriğinde yer alan sayfaların birbirleri ile veya diğer web siteleri ile olan bağlantılarının, tasarımını yapan kişi tarafından nasıl düzenlenmeye dair bilgilerdir. Yapı verisi, bir web sayfasının oluşturulmasında kullanılan HTML veya XML etiketleri gibi veri yapıları olabileceği gibi, sayfalar hatta siteler arası bağlantıları sağlayan linkler şeklindeki veri yapıları da olabilir. Daha kısa bir ifadeyle yapı verisi, bir web sitesinin site haritalama araçları ile otomatik olarak oluşturulabilen harita bilgisidir.
3. **Kullanım verisi:** Kullanıcıların web kaynaklarına erişimleri sırasında sunucu ya da tarayıcılar tarafından kayıt altına alınan verilerdir. Kullanım verisinin önemli bir bölümünü sunucular üzerinde tutulan erişim kayıt(log) dosyaları oluşturmaktadır. Kayıt dosyaları içerisinde kullanıcının ve sunucunun IP adresi, bağlantı yapılan tarih ve saat, yönlendiren kaynak bilgisi, kullanılan tarayıcı ve sürüm bilgisi, sistem bilgisi ve yapılan veri transferi miktarı gibi bilgiler tutulmaktadır.
4. **Kullanıcı profil verisi:** Bir web sitesine kayıt olma sürecinde kullanıcılar tarafından sağlanan demografik bilgilerin yanı sıra kullanıcıların ilgi ve tercihlerinden oluşan verilerdir. Kullanıcı profil verisi, kayıt formları veya anketler aracılığı ile ya da web kullanım günlüklerinin analiz edilmeleri ile elde edilirler. Bunlara ek olarak kullanıcının web sitesindeki alışveriş veya ziyaret geçmişi gibi bilgiler de kayıt altına alınabilmektedir.

Yapıldırılan bir ağ üzerindeki diğer ağ bileşenlerinin(kullanıcıların) erişebileceği, kullanıcıma ve/ veya paylaşımı açık kaynakları barındırın, güçlü donanım ve yazılım bileşenlerinden oluşan bilgisayar birimine **sunucu (server)** denir.

Bir ağ üzerinde sunucu bilgisayardan hizmet alan, bilgiye erişim yetkileri sunucu tarafından belirlenen kullanıcı bilgisayarlara **istemci (client)** denir.

Bir ağ üzerinde sunucu ile istemci bilgisayarlar arasındaki bilgi akışına aracı -güvenlik duvarı, önbellekleme sistemi v.b. -olarak görev gören ara sunuculara vekil sunucu(proxy server) ya da kısaca **vekil (proxy)** denir.

Web Verisinin Özellikleri

Web ortamında bulunan verilerin standart veritabanı yönetim sistemleri verilerinden farklı olarak kendine özgü özellikleri vardır. Genel olarak web üzerinde var olan veriler izleyen özelliklere sahiptir:

- *Web ortamındaki veri miktarı aşırı büyülüktedir.* İnternete erişim olanaklarının giderek artması ve kolaylaşmasına paralel olarak web ortamındaki verinin boyutu her geçen gün katlanarak artmaktadır. Şekil 8.1'de sunulan yıllara göre web site sayılarındaki değişimini ifade eden grafik ve Şekil 8.2'de sunulan yıllara göre internet kullanıcı sayılarındaki değişimini ifade eden grafik incelemişinde de çok kısa sayılabilecek bir süre zarfında web verisindeki inanılmaz artışı kestirmek zor değildir. Günümüzde internet üzerinde bulunan verinin büyüklüğünün kestirilmesi neredeyse imkansız bir hâl almıştır. Bu nedenle geleneksel veritabanı teknikleri ile bu web verisinin üstesinden gelmek mümkün değildir.
- *Web ortamındaki veri dağıtık ve heterojen bir yapıdadır.* Internet üzerinden tüm veriler dünyanın dört bir tarafına yayılmış bilgisayarlar ve sunucular vasıtasiyla bir şekilde birbirleriyle bağlantılı olabildiğinden dağıtık bir yapıdadır. Aynı zamanda metin, resim, ses ve video gibi farklı verilerin web ortamında bir arada olmasından dolayı heterojen bir yapıya sahiptir.
- *Web ortamındaki veri yapılandırılmamıştır.* Web sayfalarının HTML gibi sınırları belirli bir formatla hazırlanmaları gerekliliği olmasına rağmen, sabit ve standart bir şablonlarının olmaması ve tasarımcıdan tasarımcıya değişen keyfi yapılara sahip olmalarından dolayı web dokümanları tekweise bir yapıya sahip değildir.
- *Web ortamındaki veri dinamiktir.* Özellikle web tabanlı veri yönetim sistemlerinde kullanılan uygulamaların çok çeşitli olmasından dolayı web dokümanlarının sunumları da veritabanlarının güncellenmelerine bağlı olarak farklılıklar göstermektedir. Alan veya dosya adlarının değiştirilmesi veya ortadan kaldırılması bunlar arasındaki bağlantı yapılarının da değiştirilmesine veya taşınmasına neden olur. Dolayısıyla web dokümanlarının yapıları da sürekli olarak değişir.

Genel olarak web ortamında karşılaşılan veri hangi özelliklere sahiptir ve web madenciliği açısından hangi sınıflara ayrılır?



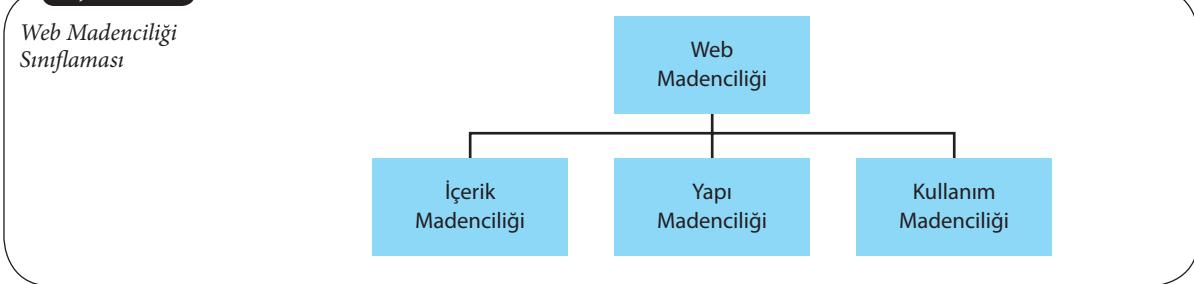
SIRA SİZDE

2

WEB MADENCİLİĞİNİN SINIFLANDIRILMASI

Web madenciliği, veri madenciliği ile çok güçlü bir ilişki içerisinde olmasına rağmen farklı bir alandır. Çünkü web madenciliği sürecinde çeşitli kategori ve biçimlerdeki internet verileri farklı alanlarda uygulanan analiz tekniklerinin kombinasyonu teknikler kullanılarak analiz edilirler.

Web madenciliği web doküman ve servislerindeki yararlı bilgileri otomatik olarak ayıklamak ve elde etmek için veri madenciliği tekniklerini kullanır. Internette yer alan bilgiler farklı veri türlerini barındırdıkları için web madenciliği, veri madenciliği sürecinde kullanılan web verilerinin türüne göre Web İçerik Madenciliği, Web Yapı Madenciliği ve Web Kullanım Madenciliği şeklinde sınıflandırılır.

Şekil 8.4

Web İçerik Madenciliği

Web içerik madenciliği temel olarak web sayfalarında kullanıcıya sunulan içerik verilerinden yararlı bilginin ortaya çıkarılması, keşfedilmesi olarak tanımlanabilir. İçerik verileri genellikle metin şeklindeki dokümanların yanı sıra tablo, şekil, resim, grafik, ses ve videolardır. Metin şeklinde sunulan içeriğin analizi metin madenciliği olarak adlandırılır ve günümüzde en çok araştırılan web içerik madenciliği alanlarından birisidir. Resim, ses ve görüntü vb. gibi kaynaklardan yararlı bilginin çıkarılması ise multimedya madenciliği olarak ifade edilmektedir. Bu alanda da başta görüntü işleme olmak üzere birçok teknik geliştirilmektedir.

Web içerik madenciliğinde kullanılan verilerin bu denli farklı yapıda olması, verilerin analizini de zorlaştırmaktadır. Dolayısıyla farklı içerik madenciliği yaklaşımıları geliştirilmiştir. Web içerik madenciliğinin uygulandığı alanlara göre kullanılan iki yaklaşım vardır.

- i. **Bilgiye Erişim Yaklaşımı:** Kullanıcı profili temel alınarak kullanıcılar sunulacak bilgileri filtrelemek, sınıflandırmak ve kullanıcı profil bilgilerini geliştirmek için kullanılan yaklaşımındır. Bilgiye erişim yaklaşımında yararlanılan temel teknik, arama motorları tarafından kullanılan web aramalarıdır. Web aramalarının ana veri kaynağı ise web sayfalarının içerdikleri metinlerdir. Bu yaklaşım ile web sayfalarının sınıflandırılması ve aynı içeriye sahip sayfaların listelenmesi sağlanabilir.
- ii. **Veritabanı Yaklaşımı:** Webdeki veriyi veritabanına kaydederek modellemek ve veriyi bütünlüğe getirmek, karmaşık bilgilerin yönetilmesi için kullanılan yaklaşımındır. Bu yaklaşım sayesinde webde yer alan veriler sınıflanarak veya kümelenerek modellenmek suretiyle veritabanları ve veri ambarları oluşturulur. Bu sayede de anahtar kelime tabanlı arama yerine daha gelişmiş sorgular çalıştırma mümkün olur.

Web Arama

Sürekli değişen ve gelişen yapısıyla internet sosyal, kültürel ve bilgi amaçlı kullanılan bir iletişim ortamıdır. Arama motorları ise klasik bilgiye erişimi çok daha kolay hâle getiren, bilgiye erişim yaklaşımı temeline dayanan web içerik madenciliğinin vazgeçilmez bir enstrümanıdır. İnternetteki milyonlarca bilgi arasından istenilen bilgiye ulaşmanın en kısa ve en kolay yoludur. Web arama, bilinen bilgiye erişim yaklaşımı model sonuçlarının kullanıldığı ve içerisinde birçok karmaşık algoritmayı barındıran bir süreç olarak tanımlanabilir.

Klasik bilgiye erişim yaklaşımında internet ortamındaki her bir doküman temel bir bilgi kaynağı ve metinsel veritabanının bir parçası olarak kabul edilir. İnternette her bir web sayfası bir doküman niteliği taşıdığından, web aramanın bilgiye erişim yaklaşımının tek ve en önemli uygulaması olarak nitelendirilmesi yanlış olmaz.

Kısa Metin İşleme

Kısa metin işleme, web sitelerinde var olan metinsel verinin derlenmesi ve sınıflandırılması işlemi olarak tanımlanabilir. Konuya göre dokümanların sınıflandırılmasında ve web sayfalarının alt kategorilere ayrılmışında kullanılan algoritmalar bütünüdür. Kısa metinlerin en bilindik uygulaması arama motorlarının kullanıcıya sunduğu aranılan kelimeyi tamamlayıcı nitelikte olan “*ilgili aramalar*” uygulamasıdır. Kısa metin işleme algoritmaları, klasik metin işleme yaklaşımlarından farklı olarak çok daha az sayıda kelimenin analiz edilmesi temeli üzerine kurulan algoritmalarıdır.

Bilgi Keşfi

Çok fazla sayıdaki web dokümanı içerisindeki ilgilenilen konu hakkında yararlı bilginin etkin bir şekilde elde edilebilmesi için yapısal yöntemlere gereksinim duyulur. Bu yöntemlerin temelinde ise konularına göre web sayfalarının sınıflanması ve ilgilenilen konu hakkında daha detaylı bilgi edinmek için benzer içerikteki diğer web sayfalarının taranması yer almaktadır. Bir dokümanın otomatik olarak konusunun belirlenmesi, arama sonuçlarının özetlenmesi, görüş madenciliği, istenmeyen dokümanların filtrelenmesi vb. gibi birçok alanda da kullanılmaktadır. Örneğin, belirli bir konu üzerinde derinlemesine araştırma yapmak isteyen bir kullanıcının birkaç kelime ile arama yaparak istediği detaylı bilgiyi elde etmesi pek mümkün olmamaktadır. Dolayısıyla webde yer alan dokümanların sahip oldukları içerik yönünden belirli konu başlıklarına göre ayırmaları büyük önem taşımaktadır.

Web Görüş Madenciliği

Internet salt bilginin yanı sıra insanların bir konu, ürün ya da hizmetlarındaki şahsi görüşlerini paylaştıkları bir ortamdır. Bunun için e-ticaret sitelerindeki ürünleri satın alanların yapmış oldukları yorumların yanı sıra genel olarak oluşturulan forumlar, tartışma grupları ve bloglar vb. gibi ortamlarda kullanıcı görüşleri yer almaktadır. Dolayısıyla buralarda yer alan kullanıcı görüşlerinin analiz edilerek değerlendirilmesi de büyük önem taşımaktadır. Bu analizlerin yapılabilmesi kullanım açısından oldukça yararlı olmasına rağmen kullanılan dilin anlaşılması ve işlenmesi gerektiğinden dolayı teknik açıdan bir takım zorluklar içermektedir.

Web görüş madenciliği ile bir ürün veya hizmet hakkında yapılan olumlu veya olumsuz görüşler analiz edilerek kullanıcı eğilimleri tespit edilebilir, web sayfaları ona göre düzenlenebilir ve hatta sayfaya konulacak reklamların içerikleri düzenlenebilir. Örneğin, kullanıcılarından olumlu yorumlar almış bir ürün için reklam vermek suretiyle ürünün satışını artırmak hedeflenebilir. Tam tersine olumsuz görüşlere sahip bir ürün için ise reklam vermek yerine ürünün yeniden ele alınarak geliştirilmesi ve beğenilir hâle getirilmesi sağlanabilir.

Web içerik madenciliğinde kullanılan temel yaklaşımlar nelerdir?



SIRA SİZDE

Web Yapı Madenciliği

Web yapı madenciliği, web sitesinin yapısal özétini yanı kendi içerisindeki sayfalarla ve diğer sitelerle olan bağlantı yapılarını elde ederek, bu yapılardan yararlı bilginin ortaya çıkarılması olarak tanımlanabilir. Bu sayede web sayfaları sınıflandırılabilir ve farklı web siteleri arasındaki benzerlik ve ilişkiler ortaya çıkarılabilir. Böylece web sitelerinin verimlilik ve kullanışlılık değerlendirmeleri yapılabilir. Web yapı madenciliği ile internet ortamında birçok insan tarafından başvurulan ve alanında otorite olarak nitelendirilen önemli web sayfaları da belirlenebilmektedir.

Web'de yer alan tüm bilgi ve belgeler basılı dokümanların aksine birbirleriyle *hyper-links* denilen bağlantı köprüleri ile bağlıdır. Dolayısıyla internetteki tüm bilgiler belirli bir düzen ve yapı olmaksızın bu köprüler ile bir arada tutulur. Bu köprüler içerisinde gizli kalmış yapıların ortaya çıkarılması ve internette arama yapmak gibi birçok web uygulamasında kullanılabilmesi için web yapı madenciliği algoritmaları kullanılmaktadır.

İnternette Arama ve Bağlantı Köprüleri

İnternete erişimin giderek kolaylaşması var olan bilginin de bir çığ gibi büyümeye neden olmaktadır. Dolayısıyla edinilmek istenilen bilgiye erişebilmek için arama motorları denilen web siteleri kullanılmaktadır. Internetteki bilgi miktarı henüz bu denli büyük boyutlarda değil iken arama motorları yaygın konu başlıklarını için listeler oluştururlardı. Internetin yayılması ve genişlemesi ile bu yöntem hem çok maliyetli hem de uygulanamaz oldu. Bunun üzerine web sayfalarının içerikleri üzerinde kelime eşleştirme yapacak şekilde arama motorları otomatikleştirildi. Ancak otomatik arama sonucunda ortaya çıkan sonuçların önem ve ilgi düzeyi bakımından sıralanması gerekliliği doğdu. Bu problemi çözmek için de içerisinde gizli bilgi barındıran bağlantı köprülerinden yararlanıldı. Aranılan bilgi için diğer web sayfalarından en çok bağlantı yapılan web sayfaları önemli bilgi kaynakları olarak değerlendirildi ve arama sonuçlarının sıralanmasında verimliliği artırmak için kullanıldı.

1998 yılında bağlantı köprüleri temelne dayanarak PageRank ve HITS adında iki önemli algoritma geliştirildi. Her iki algoritma da web sayfalarına yapılan bağlantı köprülerinin yapısını ve sayısını kullanarak sayfaların önem düzeylerini belirlemede kullanmaktadır. PageRank, Google arama motorunun kullandığı ve sayfaların önem düzeylerini veren bir algoritmadır.

Atıf Analizi

Atıf analizi, akademik olarak yazarlar ile yayınları arasındaki ilişkiyi kurmak için yapılan alıntıları inceleyen bir araştırma alanıdır. Bir yayın başka bir yayından alıntı yaptığından bu iki yayın arasında bir ilişki veya bağlantı kurulmuş olur. Dolayısıyla atıf analizinde de bu bağlantılar incelenerek yayınların önem düzeyleri ortaya konulmaya çalışılır. Günümüzde bir yayının önemini belirleyen en önemli ölçü “*impact factor*” yani etki faktörüdür. Ortak atıf ve bibliyografik eşleme, HITS algoritmasını temel alan ve atıf analizinde dokümanların kümelenmesinde kullanılan benzerlik ölçüleridir.

Web Topluluğu Keşfi

Web topluluğu, belirli bir konu üzerinde kaynak sağlayan web sayfaları topluluğudur. Web topluluklarını keşfetme nedenlerini şu şekilde sıralayabiliriz:

- i. Topluluklar kullanıcıya değerli ve güncel bilgiler sağlarlar.
- ii. Topluluklar webin sosyolojisini simgeler ve onları inceleyerek webin gelişimini anlayıp öğrenebiliriz.
- iii. Topluluklar belirli bir kitleye erişmek açısından son derece idealdir.

Web toplulukları ilişki yakınlığı ve içerik uyumu şeklinde ortaya çıkan bir olgu olarak nitelendirilebilirler. Web topluluklarını keşfedebilmek için en çok akış modeli ve bir topluluk grafiği oluşturmak gibi farklı algoritmalar geliştirilmiştir.

Web Şeması Ölçüm ve Modellemesi

İnternette bulunan bilgiler birbirlerine web sayfaları arasındaki bağlantı köprüleri ile bağlıdır. En genel bakış açısıyla web, bağlantı köprülerinden oluşan büyük bir şema şeklinde değerlendirilebilir. Web şeması, bir enerji nakil şebeke ağı gibi karmaşık bir yapıya sahiptir

ve grafiksel özellikler taşır. Yapılan birçok istatistiksel ve deneysel araştırmalar sonucunda sıradan bir grafiksel yapının özelliklerinden farklı olarak gözle görülebilen bir güç yasası bağlantı yapısına sahip olduğu ortaya konulmuştur. Bu grafiksel yapı içerisinde var olan milyonlarca hatta milyarlarca bağlantı arasından güçlü veya zayıf olanlarının belirlenmesi gerekir. Dolayısıyla güçlü bağlantı yapılarının modellenmesi ile oluşturulacak algoritma tabanlı web araçları webin etkinliğinin daha da artmasına neden olmaktadır.

Web Sayfalarının Sınıflandırılması

Web sayfaları standart metin dokümanlarının aksine resim, ses ve video gibi multimedya dosyalarını da içermektedir ve köprü bağlantıları ile de diğer dokümanlarla bağlantılıdır. Köprü bağlantıları webin düzenlenmesinde, arama ve analiz yeteneğinin geliştirilmesinde giderek artan bir öneme sahip olmaktadır. Dolayısıyla köprü bağlantıları (veya atıflar) web sayfalarının sınıflandırılmasında da kullanılmaktadır. Atıf veya bağlantı analizi orijinal doküman ile alıntı yapan doküman arasında kurulmuş olan bağı temel alır. Kurulmuş olan bu bağ, aynı zamanda alıntı yapanın amaç veya düşünceleri hakkında bilgiyi barındıran bir bağdır. Örneğin, model uçaklarla ilgili kurulan bir web sayfasında model uçak üreten firmaların web sayfalarının veya bu alanda yapılan yarışma ve aktivitelerin bağlantılarına hatta uçak simülasyon oyunlarına ilişkin bağlantılar yer verilebilir. Dolayısıyla web sayfasında verilecek bu bağlantılar ilgili sayfayı tasarlayan kişinin görüş, düşünce ve amacına ilişkin gizli bilgileri de içerisinde barındırır.

Bağlantılar aracılığı ile web sayfalarının sınıflandırılmasında kullanılan farklı teknikler bulunmaktadır. Bu tekniklerden bir tanesi, ilgili web sayfasına yönlendirme esnasında arama motorlarına girilen cümle ya da kelimeler kullanılarak oluşturulacak *sanal liste* aracılığı ile yapılan sınıflandırmadır. Örneğin, “En sevdiğim hobim model uçak uçurmaktır.” şeklinde bir cümle ile yapılacak bir web aramasında temel kelime olan “*model uçak*” etrafındaki “en sevdiğim”, “hobim”, “sevdiğim hobim”, “uçurmak” vb. gibi kelime veya kelime grupları değerlendirilerek ilgili web sayfası sınıflandırılabilir.

Web Kullanım Madenciliği

İnternette herhangi bir kaynağa erişim sağlandığında tarayıcı veya sunucular tarafından bir takım veriler kayıt altına alınır. Bunlar sunucular tarafından depolanan kullanıcı erişim kayıtları, tarayıcı kayıtları, kullanıcı profilleri, cerezler, fare tıklamaları, sayfa kaydırımları, sayfa içerik özellikleri vb. gibi kayıtlıdır. Web kullanım madenciliği, kullanıcıdan elde edilen bu bilgiler aracılığı ile kullanıcıların internet gezinme alışkanlıklarını analiz ederek kişiye özel modeller oluşturmayı amaçlar. Bu sayede kişinin ilgi alanları belirlenebilir ve ilgi alanları ile ilgili öneriler sunulabilir. Kullanıcı erişim kayıtlarının ve kullanıcı profil verilerinin analizi ile aynı zamanda web sitesini daha etkin hâle getirebilmek için çeşitli düzenlemeler de yapılabilir.

Web Kullanım Madenciliği Aşamaları

Web kullanım madenciliğinin temel amacı, kullanıcıların internette gezinme esnasında sunucularda depolanan bilgi ve izlerin veri madenciliği yöntemleri ile analiz edilerek kullanıcılarla yönelik yararlı bilginin elde edilmesidir. Web kullanım madenciliği veri ön işleme, örüntü keşfi ve örüntü analizi olmak üzere üç aşamada gerçekleştirilir:

Veri Ön İşleme

Tüm veri madenciliği uygulamalarında olduğu gibi, web kullanım madenciliğinin de ilk ve en önemli aşaması verinin işlenerek analize uygun hâle getirilmesidir. Veri ön işleme aşaması sunucularda depolanan kullanıcı erişim dosyalarının düzensiz ve karmaşık bir

yapıda olmalarından ve çok büyük boyutlarda olabilmelerinden dolayı uzun bir uğraş gerektiren ve en zor aşamasıdır. Bu aşama, genel hatlarıyla ifade etmek gerekirse,

- i. *Verinin Temizlenmesi*: Kullanıcı erişim dosyaları içerisinde yer alan geçerliliği olmayan veri ve gereksiz bilgilerin ayıklanması işlemidir.
- ii. *Kullanıcı Bilgisinin Belirlenmesi*: Web kayıt dosyalarında yer alan erişim bilgilerinin, kullanıcıların kimlik bilgilerinden ziyade, aynı kullanıcıya ait olup olmadığıının tespit edilmesi işlemidir.
- iii. *Oturum Bilgisinin Belirlenmesi*: Kullanıcının bir web sitesine giriş-çıkışı arasında geçen süre yani bir oturumda gerçekleştiği davranış ve aktivitelerin kümelenmesi işlemidir.
- iv. *İz (Yol) Tamamlama*: Kullanıcı erişim kayıtları içerisinde çeşitli sebeplerden dolayı yer almayan eksik referansların veya kayıt dışı bağlantıların tamamlanması işlemidir. şeklinde dört alt adımdan oluşur.

Örüntü Keşfi

Bu aşama, veri ön işleme aşamasından geçirilmiş analize hazır olan verilere veri madenciliği tekniklerinin uygulanarak yararlı bilginin ortaya çıkarılması aşamasıdır. Bu aşamada işlenmiş veriden önemli ve anlamlı bilgiyi ortaya çıkarabilmek adına istatistiksel analiz, ilişki kuralları, sınıflandırma analizi, kümeleme analizi ve sıralı örüntüler vb. gibi veri madenciliği teknikleri kullanılır.

Istatistiksel Analiz: Bir web sitesini ziyaret eden kullanıcılar hakkında bilgi edinmek için kullanılan en yaygın yöntemdir. Kullanıcıların oturum bilgileri üzerinden en sık erişilen sayfalar, ortalama görüntülenme süreleri ve erişim yolunun ortalama uzunluğu gibi istatistik bilgiler elde edilebilir. Elde edilen bu bilgiler, sistem geliştirme, web sitesi güncelleme ve iş zekası gibi uygulama alanlarında sıkça yararlanılan bilgilerdir.

İlişki Kuralları: Web kullanım madenciliğinde ilişki kuralları, kullanıcıların bir oturumda eş zamanlı erişim sağladıkları farklı içeriğe sahip sayfalar arasında bir ilişki kurulabilmesi için kullanılır. Bu sayfaların birbirlerine bağlantı köprüleri ile bağlanmış olmaları gerekmek. İlişki kuralları ile elde edilen bilgiler özellikle web sitesi güncelleme alanında kullanılmaktadır.

Sınıflandırma Analizi: Sınıflandırma, elde edilen yeni bir verinin daha önceden belirlenmiş olan çeşitli sınıflardan birisine eşleştirilmesidir. Web kullanım madenciliğinde ise sınıflandırma analizi, yeni kullanıcıların daha önceden var olan kullanıcı profil sınıflarından uygun olan sınıfın belirlenmesinde kullanılır. Sınıflandırma analizinde kullanılan birçok farklı algoritma mevcuttur.

Kümeleme Analizi: Kümeleme, benzer özelliğe sahip öğelerin kümeler hâlinde gruplandırmasıdır. Kümeleme analizinde, sınıflandırma analizindeki gibi daha önceden oluşturulmuş gruplar yoktur. Web kullanım madenciliğinde kümeleme analizi kullanıcıların veya web sayfalarının kümelendirilmesi için kullanılır. Kullanıcıların kümelenmesi ile benzer web tarama stratejisine sahip kullanıcı kümelerinin oluşturulması sağlanırken buradan elde edilen bilgiler web sitesi güncelleme ve iş zekası uygulamalarında kullanılır. Sayfaların kümelenmesi ile ise yine benzer içeriğe sahip web sayfası kümelerinin oluşturulması sağlanır. Buradan elde edilen bilgiler ise arama motorları ve web destek sağlayıcıları için yararlı bilgilerdir.

Sıralı Örüntüler: Sıralı örüntüler ile kullanıcıların belirli bir zaman aralığındaki farklı oturumları arasında birbirini takip eden kullanıcı hareketleri arasındaki ilişkilerinin ortaya konulmasıdır. Buradan elde edilen bilgiler kullanıcıların gelecekteki eğilimlerini belirlenmesi açısından önemlidir ve özellikle kişiye ve alana özel pazarlama, ilan ve reklam planlamasında kullanılır.

Örüntü Analizi

Web kullanım madenciliğinin son adımda, bir önceki adımda elde edilen örüntüler (veya kurallar) içerisinde ilginç olmayanların elenerek ilginç yani kullanılabilir olanların elde edilmesi amaçlanır. Ayrıca bir önceki adım olan örüntü keşfi adımda kullanılan veri madenciliği tekniğinin işe yarar ve kullanılabilecek sonuçlar üretip üretmediği de böylece ortaya çıkar.

Uygulanan veri madenciliği algoritmalarından elde edilen sonuçlar genellikle karmaşık ve kolay anlaşılması zor olan yapıdadırlar. Dolayısıyla sonuçların öncelikle kolay anlaşılabilen bir hâle getirilmeleri gereklidir. Bunun için geliştirilmiş bir takım analiz yöntemlerinden ve araçlardan yararlanılır. Ortaya çıkan örüntülerin analizi sırasında daha iyi bir değerlendirmeye sağlayabilmek için görselleştirme tekniklerinden yararlanılabilir. Örüntü analizi için yaygın olarak kullanılan iki **araç** vardır. Bunlardan ilki, analiz yapan kişinin ihtiyacına göre verileri sıralama, filtreleme ve birleştirme imkanı sunan SQL (Structured Query Language / Yapısal Sorgu Dili) gibi bir bilgi sorgulama mekanizması kullanmaktadır. Diğer ise ilişkisel veritabanları üzerindeki verileri çok boyutlu veri küpleri hâline getirerek hızlı bir şekilde çözümlenmesini sağlayan OLAP (Online Analytical Processing / Çevrim içi Analitik İşleme) çevrim içi sorgulama ve raporlama işlemlerini kullanmaktadır.

Bu aşamada yapılacak analizler sonucunda elde edilecek veya keşfedilecek örüntülerin kullanılabilir yararlı bir bilgi olabilmesi için;

- İnsanlar tarafından kolayca anlaşılabilir olması
- Daha önceden keşfedilmemiş olması
- Belirli bir oranda geçerliliğinin sağlanmış olması
- İhtiyaçları karşılayan ve kullanılabilir olması gereklidir.

Veri madenciliği algoritmalarının sonucunda elde edilen çıktıları uygulanabilen herhangi bir araç veya filtreye **örüntü analiz aracı** adı verilir.

Web Kullanım Madenciliği Temel Uygulama Alanları

Web kullanım madenciliği, son yıllarda yoğun ilgi gösteren ve birçok güncel uygulama alanı sahip bir web madenciliği türüdür. Özellikle arama motorlarındaki çeşitlilik ve rekabet ortamı, kullanıcı odaklı arama performansını geliştirmek için araştırmacıları web kullanım madenciliğine yönlendirmiştir.

Kişiselleştirme (Personalization)

Web kullanım madenciliği ile elde edilen sonuçlar kullanıcıların ilgi alanlarına yönelik öneriler sunmak amacıyla kullanılabilir. Kullanıcıların profil verileri ve gezinme alışkanlıklarından yola çıkılarak çeşitli analizler ile yakın gelecekteki davranış ve ilgi odakları tespit edilmeye çalışılır. Özellikle e-ticaret alanında kabul edilebilir pazarlama stratejileri oluşturmak ve potansiyel müşterilere otomatik ürün önerisi sunmak için yoğun bir şekilde kullanılmaktadır.

Sistem Geliştirme (System Improvement)

Sistem geliştirmenin asıl amacı web sitesinin kullanıcı trafiği değerlendirilerek kullanıcı memnuniyeti açısından web sitesinin performans ve kalitesini artırmaktır. Ayrıca güvenlik amacıyla sisteme zarar verici saldırılardan, dolandırıcılık, kullanıcı hesaplarına yönelik kötü niyetli girişimler vb. gibi olayların engellenmesi için sistemin açık ve aksayan yönlerinin tespit edilmesi ve giderilmesinde de web kullanım madenciliği sonuçları kullanılmaktadır.

Web Sitesi Güncelleme (Site Modification)

Bir web sitesinin çekiciliğini ve popülerliğini artırmak için sitenin tasarımının iyi yapılmış olması gereklidir. Bir web sitesinin tasarımını yapıırken kullanıcıların farklı yollarla erişim sağlayacakları ve farklı zamanlarda farklı bilgilere ihtiyaç duyacakları da göz önündede bulundurulmalıdır. Dolayısıyla web kullanım madenciliği ile günün şartlarına paralel olarak kullanıcı taleplerine en iyi şekilde cevap verebilmek, kullanım kolaylığı sağlamak ve içerik açısından cazip hâle getirebilmek amacıyla çeşitli analizler yapılabilir. Analizler sonucunda elde edilen bilgiler ışığında da web sitesinin tasarımının gözden geçirilerek güncellenmesi sağlanabilir. Özellikle internet bankacılığı, e-ticaret, ürün katalogu ve eğitim gibi alanlarda sıkılıkla web sitelerinin güncellenmesi gerekmektedir.

İş Zekası (Business Intelligence)

İş zekasının ana hedefi şirket performansını artırmak ve pazarda rekabet avantajı sağlamak için insanların doğru kararlar almalarına yardımcı olmaktadır. Web kullanım madenciliği müşteri davranışlarını hakkında bilgileri ayıklamak ve yararlı ve etkili bir veritabanı oluşturmak için uygun bir tekniktir. İnternet üzerinden yapılan ürün ve hizmet satışları için müşteri potansiyelini artırmak, var olan müşterinin devamlılığını sağlamak, daha çok satış gerçekleştirebilmek ve daha etkin bir lojistik ve stok yönetimi gerçekleştirebilmek için web kullanım madenciliği sonuçlarından yararlanılabilir.

Kullanım Karakteristiği (Usage Characterization)

İnternet üzerinden yapılan ürün pazarlamalarında kullanıcıların bir ürünü araştırma ve satın alma davranışlarının bilinmesi kritik önem taşır. Web kullanım madenciliği ile kullanıcıların web üzerinde gerçekleştirmiş oldukları tüm aktiviteleri incelemek detaylı kullanıcı davranışları ve kullanım karakteristikleri belirlenebilir. Bu sayede kullanıcıların web sitesi tarama stratejileri tahmin edilmeye çalışılır.

Web madenciliği, farklı amaçlar doğrultusunda birçok alanda uygulanmasına rağmen sınıflandırılmasında temel ayrımları veri türüdür. Yapılan bu sınıflamanın çeşitli özellikler bakımından karşılaştırılması Tablo 8.3'te verilmiştir.

Tablo 8.3
Web Madenciliği Sınıflaması

Web Madenciliği Sınıfları			
	Web İçerik Madenciliği	Web Yapı Madenciliği	Web Kullanım Madenciliği
Veri görünümü	<ul style="list-style-type: none"> Yapısal Yarı yapısal Yapısız 	<ul style="list-style-type: none"> Link yapısı 	<ul style="list-style-type: none"> Etkileşimli veri yapısı
Kullanılan veri tipi	<ul style="list-style-type: none"> Birincil 	<ul style="list-style-type: none"> Birincil 	<ul style="list-style-type: none"> İkincil
Ana veri	<ul style="list-style-type: none"> Metin Hiper metin 	<ul style="list-style-type: none"> Link yapısı 	<ul style="list-style-type: none"> Tarayıcı kayıtları Sunucu kayıtları
Gösterim	<ul style="list-style-type: none"> Kavramsal Bağlantısal Kenar etiketli grafik n-gram Terimsel / İfadesel 	<ul style="list-style-type: none"> Grafiksel 	<ul style="list-style-type: none"> İlişkisel tablo Grafiksel
Yöntem	<ul style="list-style-type: none"> Otomatik öğrenme İlişki kuralları Özel algoritmalar İstatistiksel yöntemler 	<ul style="list-style-type: none"> Özel algoritmalar 	<ul style="list-style-type: none"> Otomatik öğrenme İstatistiksel yöntemler
Amaç	<ul style="list-style-type: none"> İçerik verilerinden yararlı bilginin ortaya çıkarılması, keşfedilmesi 	<ul style="list-style-type: none"> Web bağlantı yapılarının modellemesi 	<ul style="list-style-type: none"> Kişilerin kullanım alışkanlıklarının analizi ve modellemesi
Kapsam	<ul style="list-style-type: none"> Veritabanı yaklaşımı açısından bölgesel Bilgiye erişim yaklaşımı açısından evrensel 	<ul style="list-style-type: none"> Evrensel 	<ul style="list-style-type: none"> Evrensel
Hedef	<ul style="list-style-type: none"> Yararlı bilginin keşfi 	<ul style="list-style-type: none"> Web sitesinin yapısal özeti oluşturmak 	<ul style="list-style-type: none"> Kullanıcı profilinin ve davranışlarının analizi
Uygulama Alanları	<ul style="list-style-type: none"> Kümeleme Sınıflandırma Örütü ve kural çıkarımı Kullanıcı modellemesi Web şeması modelleme 	<ul style="list-style-type: none"> Kümeleme Sınıflandırma 	<ul style="list-style-type: none"> Kullanıcı modellemesi Web sitesi tasarımı, uyarlaması ve yönetimi Pazarlama
Zorluklar	<ul style="list-style-type: none"> Veri / Bilgi çıkarımı Çevrim içi kaynaklardan görüş çıkarma Web sayfalarının bölümlenmesi ve gürültü tespitı 	<ul style="list-style-type: none"> Her bir web sayfasının içerik bilgisinin olmaması Öncül sayfa içeriği ile ilişkisiz olması 	<ul style="list-style-type: none"> Kullanıcıların kimler oldukları ve ne kadar kaldıkları gibi bilgilerin ön işlemesi

Kaynak: Kaur S., Kaur K. (2015). *Web Mining and Data Mining: A Comparative Approach*, International Journal of Novel Research in Computer Science and Software Engineering Vol. 2(1), 36-42.

SOSYAL MEDYA MADENCİLİĞİ

Geniş bir açıdan ele alındığında web madenciliği, web içerisinde yer alan tüm bilgi ve belgelerin belirli bir amaç için taranarak sınıflandırılması işlemlerini kapsamaktadır. Son yıllarda hızla artış gösteren bir internet kullanım biçimi ise sosyal medya kullanımıdır. Sosyal ağ hizmetleri olarak da adlandırılan sosyal medya, insanların birbirleriyle daha kolay etkileşim, iletişim ve paylaşımında bulunmalarını, kısaca sosyal ilişkiler kurmalarını sağlayan internet tabanlı uygulamaları kapsayan bir platform olarak tanımlanabilir. Ba-

ğımsız ve yerleşik sosyal ağ hizmetlerinin çevrim içi alandaki çeşitliliği kesin bir sosyal medya tanımının yapılmasında karmaşa yaşanmasına neden olmaktadır. Ancak sosyal medya olarak adlandırılan tüm bu hizmetler izleyen ortak özelliklere sahiptir.

- i. Sosyal medya hizmetleri, (günümüz koşullarında) Web 2.0 internet tabanlı uyugulamalarıdır.
- ii. Sosyal medya hizmetleri, kullanıcı tarafından oluşturulan ve değiştirilebilen bir içeriğe sahiptir.
- iii. Sosyal medya hizmetlerinde, site veya uygulama için güvenliği, tasarımları ve bakımı hizmet sağlayıcı tarafından sağlanan bireysel veya grup profilleri oluşturulabilir.
- iv. Sosyal medya hizmetleri, bir kullanıcı ile diğer kullanıcılar ve/veya gruplar arasında bağlantılar kurarak çevrim içi sosyal ağlar oluşturulmasını kolaylaştırır.

Günümüzde insanlar birçok nedenden dolayı sosyal medyayı kullanmaktadır. Etkin bir şekilde kullanılmakta olan sosyal medya hizmetlerinin hangi amaçlar için kullanıldığını belirleyebilmek için insanların birbirleriyle çevrim içi etkileşim nedenlerini tüm yönleriyle ele almak gereklidir. Dolayısıyla günümüzde kullanılmakta olan sosyal medya hizmetlerini temel olarak izleyen biçimde sınıflandırmak mümkündür.

1. *Genel amaçlı veya arkadaş tabanlı:* Bu hizmetler belirli bir konu üzerine odaklanmayan arkadaşlık temeline dayanan paylaşım hizmetleridir.
2. *Bilgilendirici:* Bu hizmetlerin amacı günlük sorunlara yanıtlar sunmaktır.
3. *Mesleki:* Bu hizmetler kariyer veya meslek planlamasında yeni fırsatlar edinmek için kullanılır.
4. *Eğitim:* Bu hizmetler öğrencinin deneyimini geliştirmek için kullanılır.
5. *Hobiler:* Bu hizmetler aynı şelyelere ilgi duyan insanlar için bir buluşma noktasıdır.
6. *Akademik:* Bu hizmetler akademik ve bilimsel çalışmalar için güncel bilgi kaynağına erişim sunan hizmetlerdir.
7. *Haberler:* Bu hizmetler tüm toplumu ilgilendiren haber yayıcılığına ilişkin hizmetlerdir.

Sosyal ekosistemler dil ve coğrafyadan bağımsız, dinamik, hem kamu hem de özel kürum ve kuruluşları da içeren ve açık bir yapıya sahip olduğundan, sosyal medyanın önemi hem bireyler hem de işletmeler açısından giderek artmaktadır. Birçok işletme lisanslı web sitelerine ek olarak kendi isimlerine tescil edilmiş sosyal medya hesapları oluşturmaktadır.

Pew Research Center (<http://www.pewinternet.org/2015/10/08/social-networking-usage-2005-2015>) araştırmalarına göre Amerika Birleşik Devletlerinde yetişkinlerin %65'i sosyal medyayı kullanmaktadır. Bu oran araştırmaların yapılmaya başladığı yıl olan 2005 yılında %5 iken 2015 yılında %65'e ulaşmıştır. 18 ve 29 yaş aralığında yer alan bireylerin %90'ı sosyal medyayı kullanmaktadır. Ek olarak 65 yaş ve üzerindeki bireylerin kullanımı ise %11 düzeyinde yer almaktadır.

Sosyal medya uygulamalarının bireyler tarafından bu kadar ilgi görmesi, bu konuda hizmet vermekle olan firma sayısını da arttırmaktadır. Her geçen gün sayısı hızlı bir biçimde artmakla beraber hâlen 50'nin üzerinde sosyal medya uygulaması varlığını sürdürmektedir. En çok bilinen ve kullanılan bu sosyal medya uygulamalarından Facebook, WhatsApp, Facebook Messenger, Google Plus, QQ, WeChat, Qzone, Tumblr, Instagram, Twitter, Baidu, Tieba, Skype ve Viber ilk akla gelenler olarak sayılabilir. Bu sosyal medya uygulamalarının kullanıcı sayıları da azımsanamayacak büyükliklere ulaşmıştır. Örneğin 2016 yılı başı itibarı ile Facebook kullanıcı sayısı 1,5 milyarın üzerindedir. Twitter kullanıcı sayısı ise 320 milyon civarındadır.

2016 yılı başında yapılan araştırmalara göre Türkiye'de internet kullanıcılarının %53'ü sosyal medya sitelerine erişim sağlamaktadır. Bu oran Hong Kong için %66 iken Mısır'da %30 olarak gerçekleşmektedir.

Sosyal medyanın bu kadar büyük kitlelere ulaşılmasına olanak vermesi, şirketlerin de burada yer alma isteğini artırmaktadır. Birçok firma sosyal medya hesapları yardımıyla müşteri sayısını artırma çabaları göstermektedir. Ancak şirketlerin hangi sosyal medya kuruluşu için yatırımlar yapması gerektiğini de analiz etmeleri gerekmektedir. Hedeflenen sosyal medya ortamının yanlış seçilmesi yapılan yatırımin geri dönüş sağlamaması ile sonuçlanacak, beklenen kazanımlar elde edilemeyecektir. Örneğin Facebook için ayda ortalama erişilen gün sayısı 15 gün iken bu değer Twitter için 7,5 gün olarak ortaya çıkmaktadır. (<http://www.smartinsights.com/social-media-marketing/social-media-strategy/new-global-social-media-research>)

İnternet üzerinde sosyal medya kullanım istatistiklerini yaylayan birçok kuruluş bulunmaktadır. Bu kuruluşlardan birisi olan Socialbakers firması ülkeler bazında Facebook, Twitter, YouTube, LinkedIn, Instagram, Google+ ve VK.com sosyal medya sitelerine ilişkin kullanım istatistiklerini anlık olarak yayımlamaktadır. Örneğin firmanın resmi internet sitesi <https://www.socialbakers.com> internet sitesinden edinilen istatistiğe göre Haziran 2016 itibarı ile Türk Hava Yolları'nı Facebook aracılığıyla takip eden izleyici sayısı yaklaşık 9,5 milyondur.

Buraya kadar ele alınanlardan görüldüğü üzere sosyal medya madenciliği işletmeler için önemli bir girdi hâline gelmektedir. Sosyal medya üzerinden müşteri isteklerinde meydana gelen değişimlerin izlenmesi ve bunların ürünlere uygulanması artık giderek önem arz etmektedir. Ünitenin kalan kısmında R ile Facebook ve Twitter üzerinden nasıl veri madenciliği yapılabileceğine ilişkin bilgiler ele alınmıştır.

Günümüzde kullanıcıların hizmetine sunulan sosyal medya hizmetlerinin ortak özellikleri nelerdir?



SIRA SİZDE

4

R ile Twitter Verisinin Analizi

Bireylerin anlık bilgileri paylaşma isteğine cevap vermesi bakımından kullanıcıları için kullanışlı bir ortam olan Twitter, özellikle kısa mesajlaşma aşamasında faydalı bir platform olarak karşımıza çıkmaktadır. Twitter verileri ile R'de birçok farklı analiz gerçekleştirmek mümkündür. Ancak yapılacak analizlerde kullanılacak Twitter verilerini kullanıcının kendi verileri ve tüm kullanıcıların verileri olmak üzere iki kısımda incelemek yerinde olacaktır.

Analiz I: Kişisel Twitter Verilerinizin Analizi

Twitter'da kullanıcı hesabınızı (profilinizi) oluşturduğunuz andan itibaren ilk attığınız tweet'ten son attığınız tweete kadar olan tüm tweet verilerinize ulaşmanız ve bu verilerle bir takım analizler yapmanız mümkündür. Bunun için öncelikle Twitter hesabınızdaki verilerinizi izleyen adımlar vasıtası ile elde edilmesi gerekmektedir.

1. Twitter hesabınıza kullanıcı adı ve şifreniz ile giriş yaptıktan sonra "Ayarlar" bölümüne girilir.
2. Ayarlar ekranının sol tarafındaki menüden "Twitter Verilerin" seçeneği seçilir. Ekranın görüntülenebilmesi için güvenlik dolayısıyla tekrar şifre girişi yapılması gereklidir.
3. Gelen ekranda "Diğer veriler" başlığı altındaki "Twitter Arşivi" seçeneği seçilerek arşiv isteği yapılır.
4. Twitter veri arşivinizi indirebileceğiniz bağlantı, profilinizde kayıtlı olan e-posta adresinize gönderilir.

5. İlgili e-posta kutunuza gelen e-postada yer alan “Şimdi İndir” seçeneği tıklandığında Twitter hesabınıza tekrar yönlendirilir ve verilerinize ait dosyayı indirebilirsiniz.
6. İndirdiğiniz “zip” uzantılı dosya içerisindeki “tweets.csv” dosyası size ait Twitter veri dosyasıdır.

İlgili adımlar sonucunda elde etmiş olduğunuz Twitter verilerinizi için R’de veri ya-pısına uygun çeşitli istatistiksel analizler yapabilir, grafikler elde edebilirsiniz. Örneğin verilerinizi grafiklerle ifade etmek sırasıyla **ggplot2**, **lubridate** ve **scales** paketlerinin R’de kurulması ve hafızaya yüklenmesi gereklidir. İlgili fonksiyonlar hakkında yardım için **help(“fonksiyon adı”)** komutundan yararlanılabilir.

INTERNET



<https://cran.r-project.org/web/packages/ggplot2/>

INTERNET



<https://cran.r-project.org/web/packages/lubridate/>

INTERNET



<https://cran.r-project.org/web/packages/scales/>

Bilgisayarınıza kaydetmiş olduğunuz Twitter verilerinizin R programında “tweets” de-ğişkenine aktarılması izleyen komut satırı ile gerçekleştirilebilir.

```
> tweets <- read.csv("../tweets.csv", stringsAsFactors=FALSE)
```

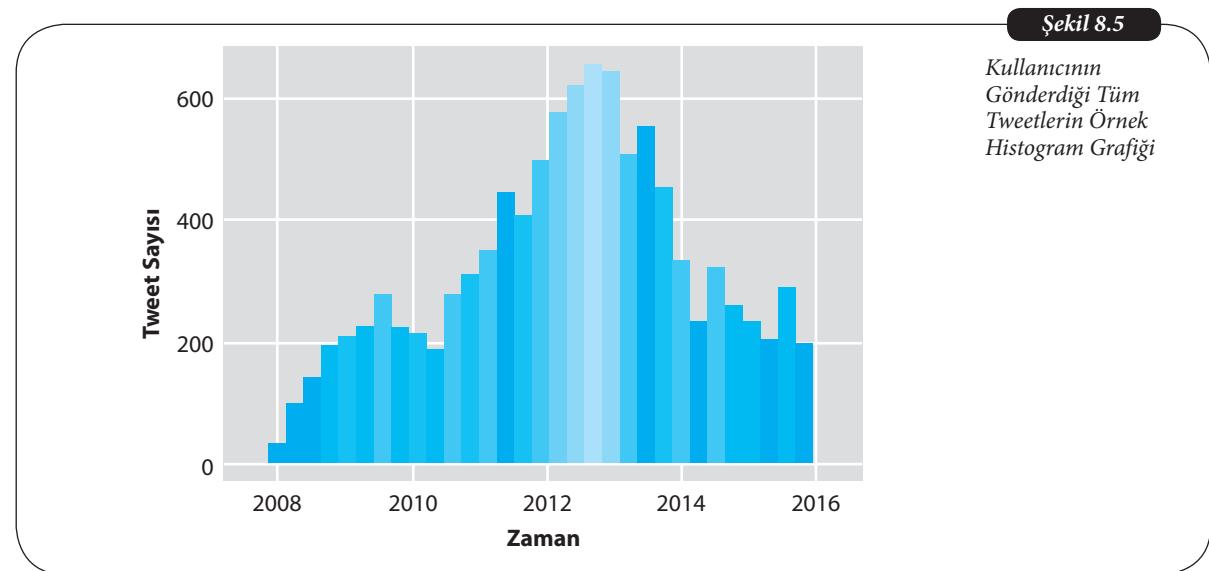
Komut satırında yer alan ve “...” şeklinde üç nokta ile işaretlenen kısıma bilgisayarınıza kaydetmiş olduğunuz Twitter veri dosyanızın dosya yolunun yazılması gerekmektedir (Örneğin C:/Users/Alper Bekki/Documents). Ünitenin ilerleyen kesiminde verilen komut satırlarının bazlarında “**koyu**” olarak işaretlenmiş olan kısımlar için kullanılan fonksiyonun yapısına bağlı olarak alternatif girişler yapmak mümkündür. Çizilecek uygulama gra-fikleri zamana bağlı grafikler olduklarından ve Twitter veri dosyasındaki zaman bilgileri metin formatında olduğundan izleyen komut dizisi ile zaman formatına dönüştürülür.

```
> tweets$timestamp <- ymd_hms(tweets$timestamp)
> tweets$timestamp <- with_tz(tweets$timestamp, "GMT")
```

Yapılan dönüşümün ardından ilk olarak zaman içerisinde göndermiş olduğunuz twe- etlerin dağılımını gösteren histogram izleyen komut satırı ile elde edilir. Sonuçta elde edi-lecek grafik için örnek bir grafik Şekil 8.5’té verilmiştir.

```
> ggplot(data = tweets, aes(x = timestamp)) +
  geom_histogram(aes(fill = ..count..)) +
  theme(legend.position = "none") +
  xlab("Zaman") + ylab("Tweet Sayısı") +
  scale_fill_gradient(low = "midnightblue", high = "aquamarine4")
```

Şekil 8.5



Gönderdiğiniz tweetlerin sırasıyla yıllara, aylara ve günlere göre dağılımlarını incelemek için yukarıda verilen komut dizisinin ilk satırının izleyen şekillerde değiştirilmesi yeterli olacaktır.

Yıllara göre dağılımı için,

```
> ggplot(data = tweets, aes(x = year(timestamp))) +
```

Aylara göre dağılımı için,

```
> ggplot(data = tweets, aes(x = month(timestamp, label = TRUE))) +
```

Günlere göre dağılımı için,

```
> ggplot(data = tweets, aes(x = wday(timestamp, label = TRUE))) +
```

Analiz II : Kelime Bulutu

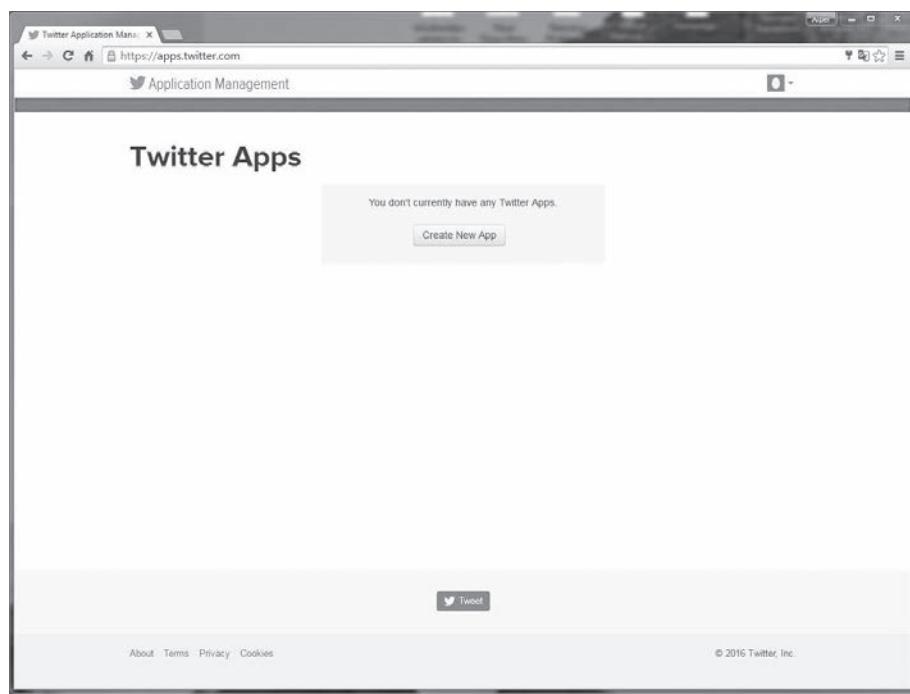
Kişisel Twitter verilerinizin haricinde tüm Twitter kullanıcılarının göndermiş oldukları tweetler, R'de belirli anahtar kelime veya kelimeler girmek suretiyle süzgeçten geçirilebilir ve sonuçta girilen anahtar kelime(ler) ile ilişkili bir kelime bulutu grafiği elde edilebilir. Twitter ortamındaki tüm tweetleri tarayabilmek için R kullanıcıların Twitter'in uygulama programlama arayüzüne(**API**) erişim sağlamaları gerekmektedir.

Twitter kullanıcı tweetlerine erişebilmek için öncelikle Twitter uygulama hizmetinden bir uygulama oluşturulması ve oluşturulan uygulama için doğrulama işleminin yapılması gerekmektedir. Twitter'den uygulama oluşturabilmek için internet tarayıcınızın adres kısmına <https://apps.twitter.com> yazdıktan sonra açılan sayfanın sağ üst köşesinde yer alan "Sign in" seçeneği seçilir ve varolan Twitter kullanıcı adı ve şifresiyle giriş yapılarak Şekil 8.6'da gösterilen uygulama oluşturma ekranına ulaşılır.

API (Application Programming Interface / Uygulama Programlama Arayüzü), bir yazılımın başka bir yazılımda tanımlanmış fonksiyonlarını kullanabilmesi için uygulama oluşturmada kullanılan alt program, protokol ve araçlar bütünüdür.

Şekil 8.6

Twitter Uygulama Oluşturma Ekranı



Uygulama oluşturma ekranında yer alan “Create New App” butonu tıklandığında Şekil 8.7’de görülen oluşturulacak yeni uygulamaya ilişkin bilgilerin girileceği ekrana ulaşılır.

Şekil 8.7

Oluşturulacak Yeni Twitter Uygulaması için Bilgi Giriş Ekranı

Create an application

Application Details

Name *
Kelime Bulutu

Your application name. This is used to attribute the source of a tweet and in user-facing authorization screens. 32 characters max.

Description *
R ile Twitter Veritabanı Analizi

Your application's description, which will be shown in user-facing authorization screens. Between 10 and 200 characters max.

Website *
http://test.com

Your application's publicly accessible home page, where users can go to download, make use of, or find out more information about your application. This fully-qualified URL is used in the source attribution for tweets created by your application and will be shown in user-facing authorization screens.
(If you don't have a URL yet, just put a placeholder here but remember to change it later.)

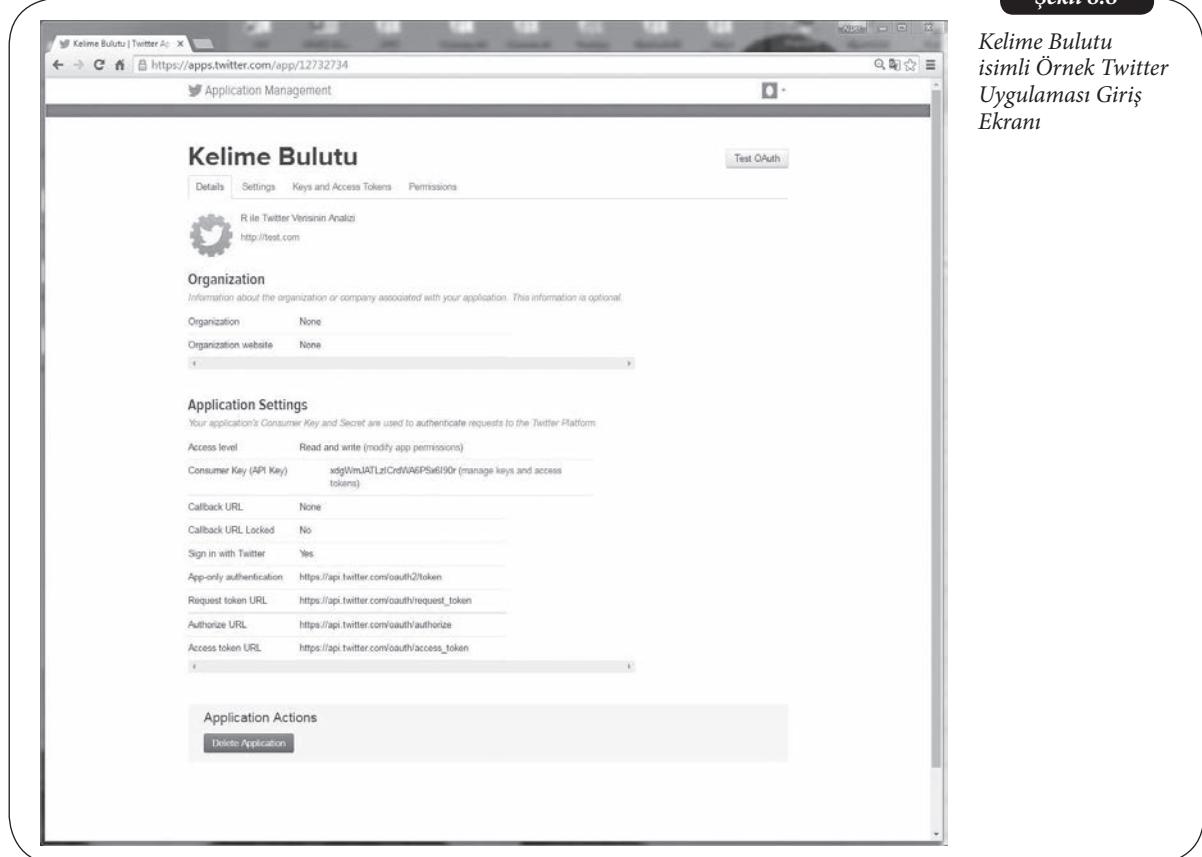
Callback URL
Where should we return after successfully authenticating? OAuth 1.0a applications should explicitly specify their oauth_callback URL on the request token step, regardless of the value given here. To restrict your application from using callbacks, leave this field blank.

Developer Agreement
 Yes, I have read and agree to the Twitter Developer Agreement.

Create your Twitter application

Şekil 8.7'den de görüldüğü üzere köşesinde kırmızı yıldız olan (Name, Description ve Website) alanların doldurulması ve en alta yer alan Twitter Geliştirici Anlaşmasını okuyup kabul ettiğinizi bildiren kutucuğun işaretlenmesi gerekmektedir. Gerekli bilgi girişleri sağlandıktan sonra “Create your Twitter application” butonu tıklanarak Twitter verilerine erişim için gereken uygulama oluşturulmuş olur. Analiz yapılabilmesi için oluşturulan Kelime Bulutu isimli örnek Twitter uygulamasının giriş ekranı Şekil 8.8'de görülmektedir.

Şekil 8.8

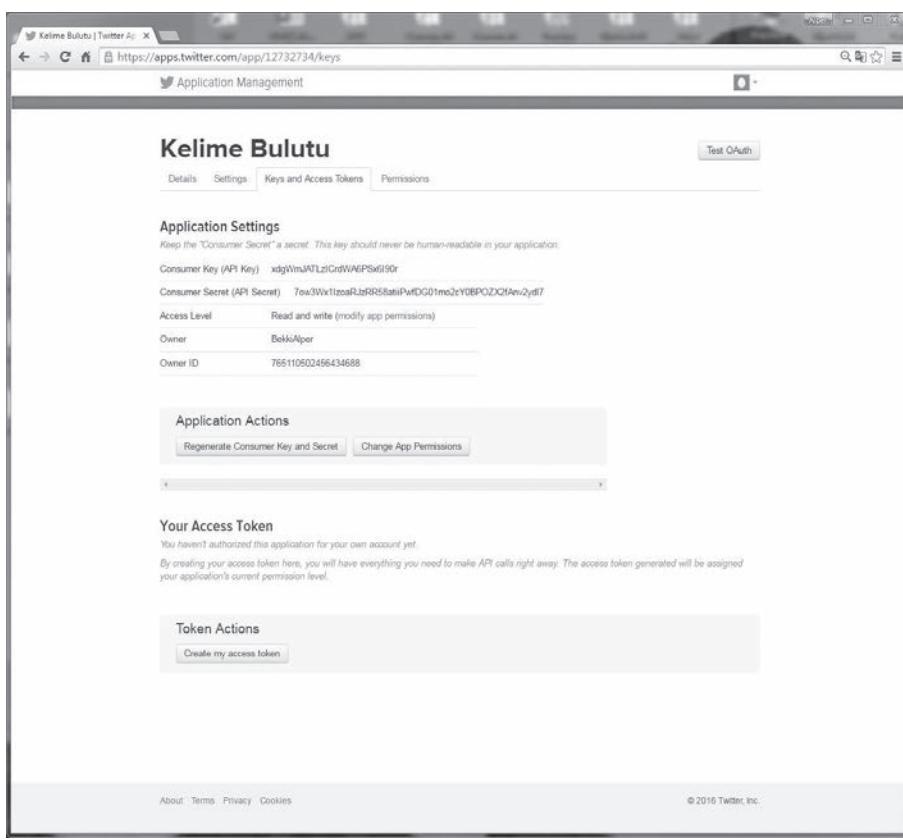


Kelime Bulutu
isimli Örnek Twitter
Uygulaması Giriş
Ekranı

Yeni uygulama oluşturulduktan sonra, R'den Twitter'a erişim sağlayabilmek için 4 adet bilgiye ihtiyaç duyulmaktadır. Bu bilgiler oluşturulan Twitter uygulamasının giriş ekranındaki “Keys and Access Tokens” sekmesinden edinilir. “Keys and Access Tokens” sekmesine ilişkin ekran Şekil 8.9'da görülmektedir. Erişim için gereken bu 4 adet bilgiden ilk ikisi *Consumer Key (API Key)* ve *Consumer Secret (API Secret)* bilgileridir. Diğer ikisi ise oluşturulmamış ise yine bu sekmeden oluşturulması gereken *Access Token* ve *Access Token Secret* bilgileridir.

Sekil 8.9

*Kelime Bulutu Örnek Twitter Uygulaması
“Keys and Access Tokens” Ekranı*



Access Token ve Access Token Secret bilgilerinin oluşturulması için sayfanın en altında yer alan “Token Actions” bölümündeki “Create my access token” butonunun tıklanması yeterlidir.

R’de Twitter verileri ile analiz yapabilmek için gereken tüm bilgiler edinildikten sonra, R’den Twitter’'a bağlantı kurabilmek için **setup_twitter_oauth()** fonksiyonundan yararlanılır. Bu fonksiyonun temel parametreleri Twitter’dan edinilen bu 4 adet şifre bilgisidir. Fonksiyonun kullanılışına ilişkin komut satırı izleyen biçimde ortaya çıkacaktır.

```
> setup_twitter_oauth(ConsumerKey, ConsumerSecret, AccessToken, AccessTokenSecret)
```

Oluşturulan Kelime Bulutu Örnek Twitter Uygulamasından edinilen 4 şifre bilgisi kullanılarak yapılan doğrulama işleminin ardından kullanıcılarının göndermiş oldukları tweetler, R’de belirli anahtar kelime(ler) girilerek taranabilir ve kelime bulutu grafiği elde edilebilir. Bunun için sırasıyla **twitteR**, **tm** ve **wordcloud** paketlerinin R’de kurulması ve hafızaya yüklenmesi gereklidir. İlgili fonksiyonlar hakkında yardım için **help("fonksiyon adı")** komutundan yararlanılabilir.

İNTERNET



<https://cran.r-project.org/web/packages/twitteR/>

İNTERNET



<https://cran.r-project.org/web/packages/tm/>

<https://cran.r-project.org/web/packages/wordcloud/>



INTERNET

Örnek olarak, son zamanlarda Türkiye'de yaşanan depremlerle ilgili atılan tweetlere ilişkin bir kelime bulutu grafiği oluşturalım. Bunun için `searchTwitter()` fonksiyonundan yararlanarak, "deprem" ve "derinlik" anahtar kelimeleri ile Türkçe olarak atılmış olan 1000 güncel (recent) tweet için sorgulama yapalım. Sorgulama sonuçlarını "deprem" değişkenine atayan komut satırı izleyen biçimde ortaya çıkacaktır.

```
> deprem <- searchTwitter('deprem + derinlik', lang="tr", n=1000, resultType="recent")
```

Girilen anahtar kelimelere göre "deprem" değişkenine atanan toplam 554 tweet olduğu görülmektedir. Kelime bulutu grafiğinin çizilebilmesi için "deprem" değişkeninin formatının izleyen komut dizisi ile önce metin formatına, daha sonra da kitaplık(korpus) formatına dönüştürülmesi gerekmektedir.

```
> deprem_metin <- sapply(deprem, function(x) x$getText())
> str(deprem_metin)
> deprem_korpus <- Corpus(VectorSource(deprem_metin))
```

Kitaplık formatına dönüştürülen ve "deprem_korpus" değişkenine atanan 554 tweet içerisindeki sırasıyla birinci ve onuncu tweet izleyen biçimde ortaya çıkmaktadır.

```
> deprem_korpus[[1]]$content
[1] "TEKELIOREN-TARSUS (MERSIN)\nBüyüklük: 2.2\nTarih:\n 15.08.2016\nSaat:\n 16:28:15\nDerinlik:\n 5.4 km. #deprem #MERSIN"
> deprem_korpus[[10]]$content
[1] "Yer: NASA-SIMAV (KUTAHYA) / Tarih: 15.08.2016 / Saat: 09:11:29 / Büyüklük: 3.2 / Derinlik: 5.4 Km #deprem"
```

Son olarak görsel açıdan daha etkin bir kelime bulutu grafiği elde etmek için kitaplık formatında elde edilen "deprem_korpus" değişkeninin içeriği tweetlerdeki ortak ve etkisiz kelimelerin, noktalama işaretlerinin ve rakamların temizlenmesi gereklidir. Verilerde yapılacak temizleme işlemi `tm` paketinde yer alan `tm_map()` fonksiyonu yardımı ile gerçekleştirilebilir.

<https://cran.r-project.org/web/packages/tm/>



INTERNET

Veri temizleme işlemi izleyen komut dizisi ile gerçekleştirilebilir.

```
> etkisizkelime <- c("acaba", "altı", "ama", "ancak", ...)
> ortakkelime <- c("Büyüklük", "Derinlik", "#deprem", ...)
> deprem_temizlik <- tm_map(deprem_korpus, removePunctuation)
> deprem_temizlik <- tm_map(deprem_temizlik, content_transformer(tolower))
> deprem_temizlik <- tm_map(deprem_temizlik, removeWords, etkisizkelime)
> deprem_temizlik <- tm_map(deprem_temizlik, stripWhitespace)
> deprem_temizlik <- tm_map(deprem_temizlik, removeNumbers)
> deprem_temizlik <- tm_map(deprem_temizlik, removeWords, ortakkelime)
```

Veri temizleme işlemlerinin ardından elde edilen ve "deprem_temizlik" değişkenine atanan kelimeler içerisinde gözlenme sayısı en fazla olan 50 kelime için `wordcloud()` fonksiyonu yardımıyla kelime bulutu grafiği çizmek için gereken komut satırı izleyen biçimde ortaya çıkacaktır. Sonuçta elde edilen kelime bulutu grafiği ise Şekil 8.10'da verilmiştir.

```
> wordcloud(deprem_temizlik, max.words=50, colors=rainbow(50))
```

Sekil 8.10

*“Deprem” ve
“Derinlik” Anahtar
Kelimelerini İçeren
Tweetlere İlişkin
Kelime Bulutu Grafiği*

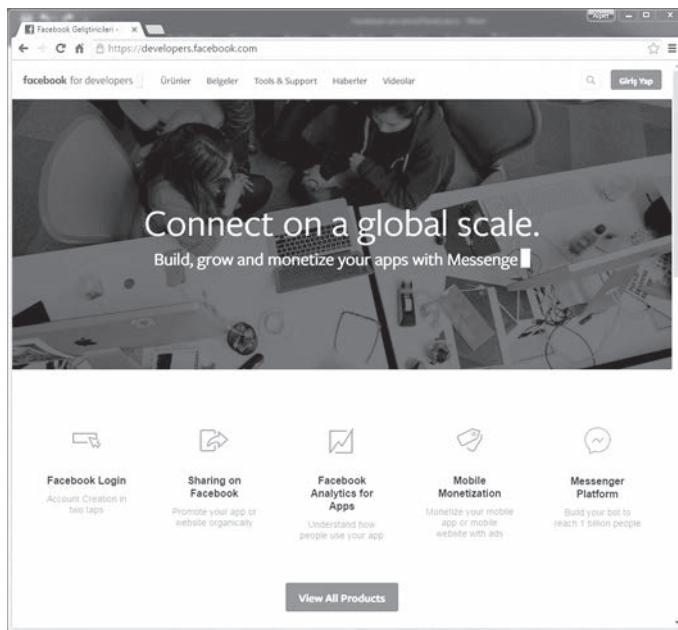


R ile Facebook Verisinin Analizi

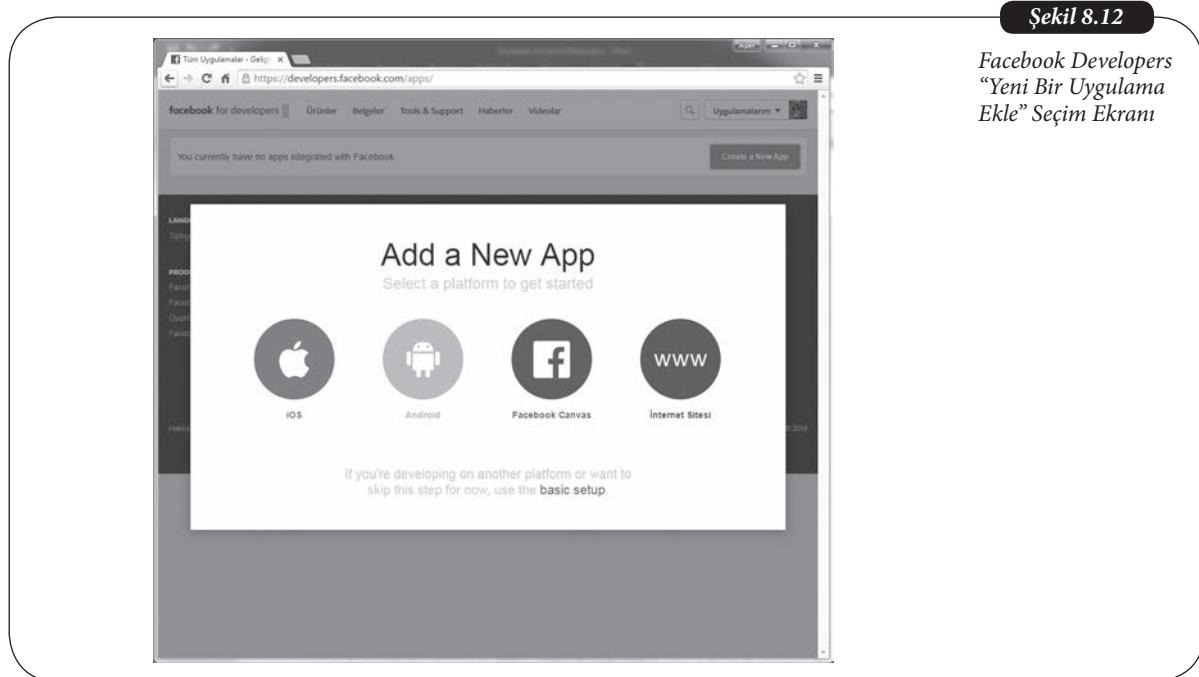
Facebook, kayıtlı kullanıcılarına profil oluşturma, fotoğraf ve video yükleme, mesaj gönderme gibi olanaklar sunan, temel olarak insanlar arasında iletişim kurulması ve bilgi alış-verisi yapılmasını amaçlayan bir sosyal ağ hizmetidir. R ile Facebook ortamındaki temel bazı bilgilere erişebilmek ve analizler gerçekleştirebilmek için kullanılan yöntemlerden birisi Facebook'un uygulama programlama arayüzü(API) olan “Facebook Developers” üzerinden bir uygulama oluşturmaktır. Bunun için öncelikle bir Facebook hesabınızın olması ve <http://developers.facebook.com> web sitesi üzerinden sisteme giriş yapılması gerekmektedir. Facebook Developers sitesi giriş ekranı Şekil. 8.11'de görülmektedir.

Sekil 8.11

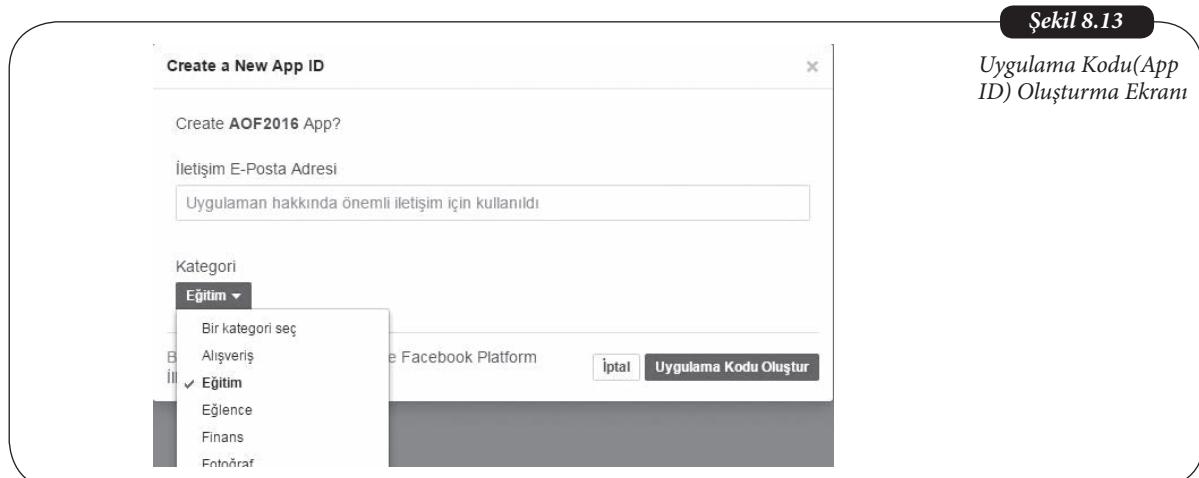
*Facebook Developers
Giriş Ekranı*



Şekil 8.11'in sağ üst köşesinde de görüldüğü üzere "Giriş Yap" butonu yardımı ile var olan Facebook hesabı bilgileri ile giriş yapılır. Yeni gelen ekran Şekil 8.11'de görülen ekrana çok benzer bir ekrandır ancak yeni ekranın sağ üst köşesinde kullanıcıya ait hesabın fotoğrafı yer almaktadır. Kullanıcı fotoğrafının hemen yanında yer alan "Uygulamalarım" açılır sekmesine tıklandığında çıkan menüde "Yeni bir uygulama ekle" sekmesi seçilir. Bu seçimle birlikte kullanıcının karşısına Şekil 8.12'de verilen ekran görüntülenir.



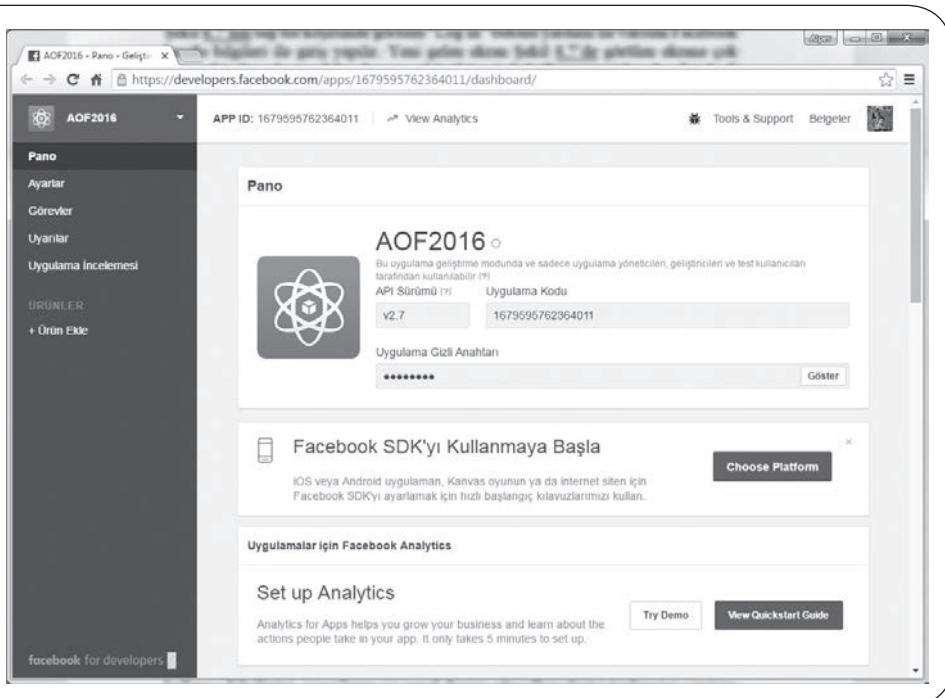
Yeni uygulama seçim ekranında yer alan seçenekler arasından en sağda yer alan "İnternet Sitesi" seçeneği seçildikten sonra çıkacak ekranda oluşturulan yeni uygulama için uygun bir uygulama ismi belirlenir. Bu çalışma için oluşturulan uygulamanın ismi AOF2016 olarak belirlenmiştir. Uygulama isminin yazıldığı alanın hemen altında yer alan "Yeni Facebook Uygulama ID'si Oluştur" sekmesi seçildiğinde açılan Uygulama Kodu (App ID) oluşturma ekranında ilgili kısma e-posta adresinizin yazılması ve uygulama sayfası için bir kategori belirlemeniz gereken Şekil 8.13'de verilen ekran görüntülenir.



Daha sonra sayfanın alt sağ tarafında yer alan “Uygulama Kodu Oluştur” butonu tiklanarak ilerlenir. Bir sonraki ekran olan güvenlik kontrolü ekranında görüntülenen güvenlik sorusuna gereken cevaplar verildikten sonra uygulama için oluşturulan internet sitesi hızlı başlangıç ekranı görüntülenir. Hızlı başlangıç ekranının en alt kısmında yer alan “Tell us about your website” bölümünün “Site URL” kısmına internet sitenizin adresinin yazılması istenir. Yapacağımız örnek uygulamanın herhangi bir internet sitesi olmadığından ve Facebook üzerinden bazı temel verilere ulaşmak amacıyla ile işlemler gerçekleştirildiğinden bu kısma “<http://localhost:1410/>” yazılmıştır. Daha sonra gelen ekranın en alt bölümünde yer alan “Skip to Developer Dashboard” seçeneği seçilir. Şekil 8.14’de görülen bir sonraki ekran, oluşturulan AOF2016 uygulamasının pano(dashboard) ekranıdır. Pano ekranı, oluşturduğunuz uygulamanın isminin, uygulama kodunun ve uygulama gizli anahtarının gösterildiği ekrandır. Pano ekranında yer alan bu bilgiler R ile yapılacak olan analizlerde Facebook verilerine erişim için kullanılacak olan fonksiyonun parametre değerleridir.

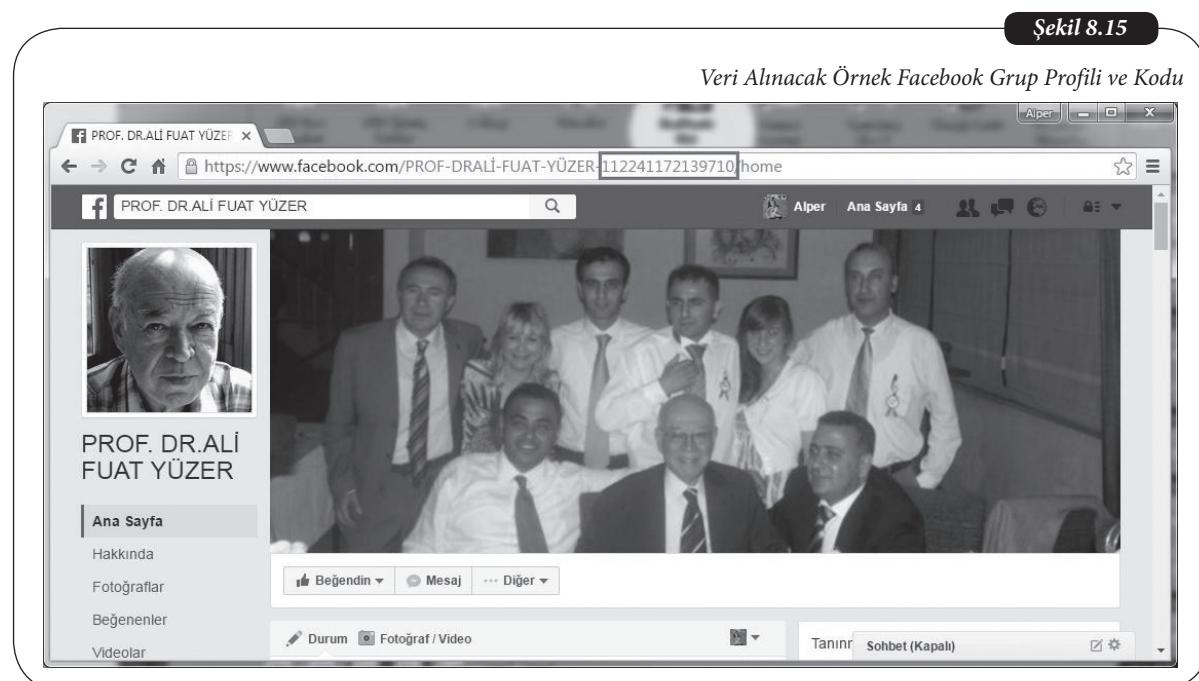
Şekil 8.14

AOF2016 Uygulaması
Pano Ekranı



Rde Facebook verilerinin analizini gerçekleştirmeden önce yapılması gereken son işlem ise verilerine erişilmek istenilen Facebook profilinin kodunun tespit edilmesidir. Bunun için yeni açılacak tarayıcı sayfasından Facebook hesabına giriş yapılarak bilgilere erişilmek istenilen kişi ve/veya grubun seçilmesi gerekir. Bu seçim yapıldığında tarayıcının adres kısmında istenilen kod görüntülenir. Yapılacak uygulamaya örnek olması bakımından Anadolu Üniversitesi Fen Fakültesi İstatistik Bölümü Emekli Öğretim Üyesi Prof. Dr. Ali Fuat Yüzer adına oluşturulmuş grup kodu Şekil 8.15’de gösterilmiştir. Veri alınabilmesi için bu grubun yöneticisi olunması zorunludur.

Şekil 8.15



R'de, profil kodu daha önceden elde edilmiş bir Facebook profili verileri ile analiz yapılabilmesi için sırasıyla **Rfacebook** ve **Rook** paketlerinin R'de kurulması ve hafızaya yüklenmesi gereklidir. İlgili fonksiyonlar hakkında yardım için **help("fonksiyon adı")** komutundan yararlanılabilir.

<https://cran.r-project.org/web/packages/Rfacebook/>



INTERNET

<https://cran.r-project.org/web/packages/Rook/>



INTERNET

R'de ilgilenilen Facebook profil verilerine erişim sağlayabilmek için **fbOAuth()** fonksiyonundan yararlanılır. Facebook'ta oluşturulan uygulama ile doğrulama yapılmasını sağlayan bu fonksiyonun temel parametreleri, oluşturulan uygulamanın pano ekranında yer alan uygulama kodu ve uygulama gizli anahtarı bilgileridir. Fonksiyonun kullanılmasına ilişkin komut satırı izleyen biçimde ortaya çıkacaktır.

```
> fb_oauth <- fbOAuth(app_id="UYGULAMA KODU", app_secret="UYGULAMA  
GİZLİ ANAHTARI")
```

Doğrulama için uygulama bilgileri ile girilen komut çalıştırıldığında,

Copy and paste into Site URL on Facebook App Settings: http://localhost:1410/
When done, press any key to continue...

uyarısı görüntülenir. Bir sonraki aşamaya geçmeden önce şayet daha önceki Facebook uygulaması internet sitesi hızlı başlangıç ekranı admımda girilmesi gereken internet sitesi adresi giriş yapılmadı ise, Facebook'ta oluşturulan uygulamanın pano ekranının sol tarafındaki menüden sırasıyla "Ayarlar" ve "Temel" seçenekleri seçilerek gelen ekranın en alt kısmında yer alan "Internet Sitesi" bölümünün "Site URL'si" kısmına internet sitenizin adresinin yazılması gereklidir. Daha sonra R'de herhangi bir tuşa basarak komut işlemi tamamlanır. Doğrulama işleminin tamamlandığını onaylamak adına R komut satırında

Waiting for authentication in browser...

Press Esc/Ctrl + C to abort

ifadesi belirirken aynı anda internet tarayıcısında boş bir sayfa açılır ve gelen ekranda

Authentication complete. Please close this page and return to R.

uyarısı görüntülenir. Açılan tarayıcı ekranı kapatıldığında R'nin komut satırında doğrulanın başarılı bir şekilde gerçekleştiğini ifade eden izleyen satırlar görüntülenir.

Authentication complete.

Authentication successful.

Doğrulama işleminin başarı ile sağlanmasıının ardından R'nin Facebook ile bağlantısı yapılmış olur ve yapılan bu bağlantıya ait kod numarası daha sonra yapılacak işlemlerde kullanılmak üzere izleyen komut satırı ile kayıt altına alınır.

```
> save(fb_oauth, file="fb_oauth")
```

Artık R ortamında Facebook kullanıcı profillerine ait temel bilgilere erişilebilir. Bunun için **getUsers()** fonksiyonu kullanılır.

Örneğin bağlantıyı gerçekleştiren kullanıcının adının ne olduğuna dair bilgiyi görüntülemek için gereken komut dizisi ve sonucu izleyen biçimde ortaya çıkacaktır.

```
> me <- getUsers("me", token=fb_oauth)
> me$name
[1] "Alper Bekki"
```

Benzer şekilde Facebook grup profillerine ait temel bilgilere de erişmek mümkündür. Bunun için ise **getGroup()** fonksiyonu kullanılır.

Hatırlanacağı üzere örnek bir uygulama yapabilmek için, Anadolu Üniversitesi İstatistik Bölümü Emekli Öğretim Üyesi Prof. Dr. Ali Fuat Yüzer adına oluşturulmuş grup kodu Şekil 8.15'te elde edilmiştir. Edinilen grup kodu kullanılarak bu gruba ilişkin verilerin elde edilmesi için gereken komut satırı ve sonucu izleyen biçimde ortaya çıkacaktır.

```
> grup <- getGroup(112241172139710, token=fb_oauth, n=500, since=NULL,
until=NULL)
```

100 posts 176 posts

Toplam 176 gönderideye ilişkin olarak alınan ve "grup" değişkenine atanan veriler **summary()** fonksiyonu yardımıyla izleyen biçimde görüntülenebilir.

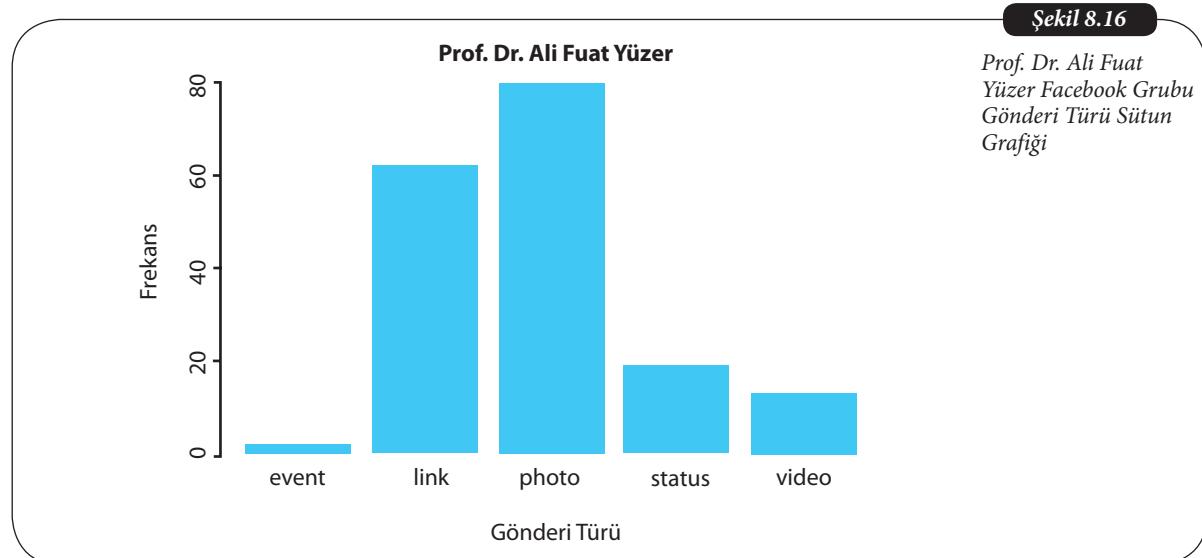
```
> summary(grup)
```

from_id	from_name	message	created_time	type
Length:176	Length:176	Length:176	Length:176	Length:176
Class :character	Class :character	Class :character	Class :character	Class :character
Mode :character	Mode :character	Mode :character	Mode :character	Mode :character

link	id	likes_count	comments_count	shares_count
Length:176	Length:176	Min. : 0.000	Min. : 0.0000	Min. : 0.0
Class :character	Class :character	1st Qu.: 0.000	1st Qu.: 0.0000	1st Qu.: 0.0
Mode :character	Mode :character	Median : 2.000	Median : 0.0000	Median : 0.0
		Mean : 3.341	Mean : 0.4034	Mean : 0.5
		3rd Qu.: 3.000	3rd Qu.: 0.0000	3rd Qu.: 0.0
		Max. :77.000	Max. :28.0000	Max. :28.0

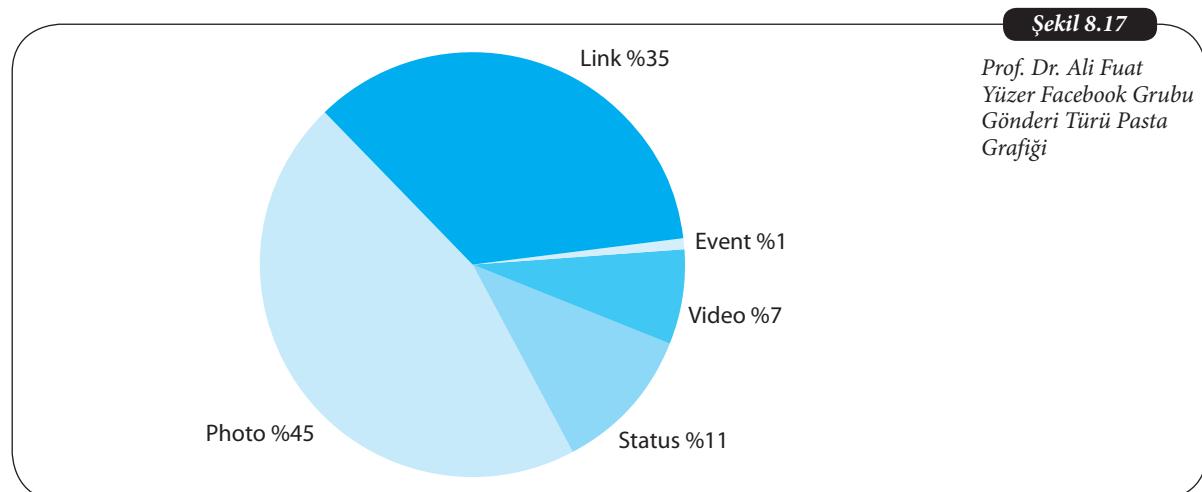
Örnek olarak incelediğimiz Prof. Dr. Ali Fuat Yüzer'in Facebook grup sayfasından elde edilen veriler için veri tipine uygun olarak çeşitli grafikler çizmek mümkündür. Örneğin toplam 176 gönderinin gönderi türü için sütun grafiği izleyen komut satırı ile elde edilir. Sonuça elde edilecek grafik ise Şekil 8.16'da verilmiştir.

```
> barplot(table(group$type),xlab="Gönderi Türü", ylab="Frekans", main="Prof. Dr. Ali Fuat Yüzer")
```



Benzer şekilde yine gönderi türlerinin toplam gönderiler içerisindeki yüzdelik oranlarını da gösterecek biçimde çizilecek pasta grafiği izleyen komut dizisi ile elde edilir. Sonuca elde edilecek grafik ise Şekil 8.17'de verilmiştir.

```
> TürYüzde <- round(table(group$type)/sum(table(group$type)),2)*100
> Etiket <- paste("%", TürYüzde, sep="")
> Etiket <- paste(names(table(group$type)), Etiket, sep=" ")
> pie(table(group$type), labels=Etiket, main="Prof. Dr. Ali Fuat Yüzer")
```

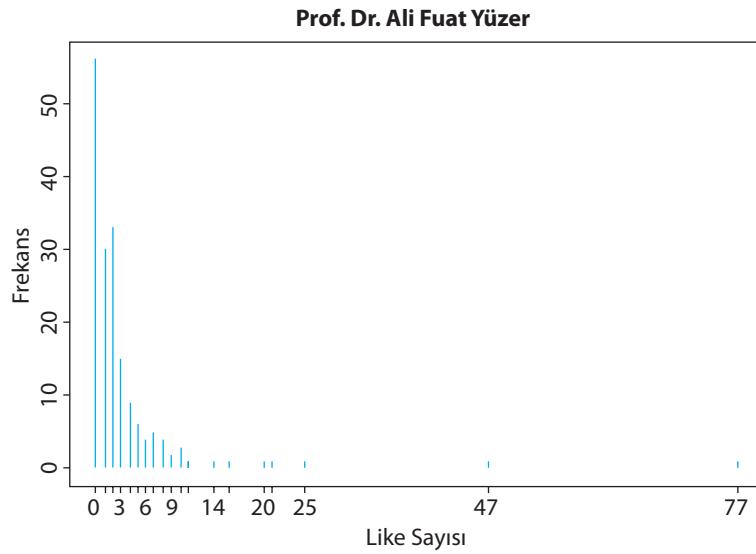


Son olarak gönderilerin almış oldukları beğenisi sayıları(likes count) için çizilecek çubuk grafiği ise izleyen komut satırı ile elde edilir. Sonuçta elde edilecek grafik ise Şekil 8.18'de verilmiştir.

```
plot(table(group$likes_count), xlab="Like Sayısı", ylab="Frekans", main="Prof. Dr. Ali Fuat Yüzer")
```

Şekil 8.18

Prof. Dr. Ali Fuat
Yüzer Facebook Grubu
Beğeni Sayısı Çubuk
Grafiği



Özet



Web madenciliği ile yararlı bilgi keşfi sürecini açıklamak
Web madenciliği, web ortamındaki dağınık, yapılandırılmamış, dinamik ve çok büyük boyutlardaki veri yoğunları içerisindeki veri madenciliği yöntemleri kullanılarak daha önceden keşfedilmemiş yararlı bilgilerin ayıklanması ve ortaya çıkarılması işlemlerinden oluşan bir süreçtir. Bu süreç ise,

- Kaynakların Tespiti
- Bilgi Seçimi ve Ön İşleme
- Genelleştirme
- Analiz

olmak üzere dört temel adımdan ibarettir. Web ortamındaki yararlı bilginin keşfedilmesi sürecinde elbette ki birçok zorluklarla da karşılaşılmaktadır. Bu süreçte karşılaşılan zorlukları ise,

- Araştırılan konuyu bulma
- İstenilen bilgiyi bulma
- Yararlı bilgi keşfi
- Bilgiyi kişiselleştirme
- Web toplulukları ve sosyal ağlar

şeklinde ana başlıklar hâlinde özetlemek mümkündür.



Web madenciliğinde kullanılan veri türlerini sınıflandırmak

Web ortamındaki sunucu, vekil ve istemci gibi farklı kaynaklardan elde edilen verileri 4 başlık altında sınıflamak mümkündür. Web madenciliğinde kullanılan veri türleri için yapılan ilk sınıflama, web sayfalarında kullanıcının bilgisine sunulan başta metin olmak üzere resim, ses, görüntü dosyaları gibi verileri içeren içeriğidir. İkinci sınıflama, bir web sayfasının tasarımının yanı iç yapısının nasıl olduğuna dair bilgileri içeren yapı verisidir. Üçüncü sınıflama, kullanıcıların internet erişimleri esnasında sunucu veya tarayıcılarla bıraktıkları internet kullanımına ilişkin bilgilerden oluşan kullanım verisidir. Dördüncü ve son sınıflama ise, herhangi bir web sayfasında düzenlenen formlar aracılığıyla edinilen, içeriği kullanıcı tarafından girilen kişisel bilgi ve ilgi alanlarına yönelik elde edilen kullanıcı profil verisidir.



Veri türüne göre web madenciliğini sınıflandırmak
İnternet ortamında var olan birbirinden farklı veri türüne, yapılacak araştırmada ulaşılacak istenen amaca ve kullanılacak tekniklere bağlı olarak web madenciliği, Web İçerik Madenciliği, Web Yapı Madenciliği ve Web Kullanım Madenciliği şeklinde sınıflandırılır.



Sosyal medya verilerini analiz etmek

Günümüzde insanlar birçok nedenden dolayı sosyal medyayı kullanmaktadır. Sosyal medya kullanımının gittikçe yaygınlaşması ve günden güne kullanıcı sayısının hızlı bir şekilde artmasından dolayı hem bireyler hem de işletmeler açısından sosyal medyanın önemi giderek artmaktadır. Buna paralel olarak sosyal medya verileri üzerinden yapılan araştırma ve analizler de gün geçtikçe artmaktadır ve çeşitlenmektedir. Dolayısıyla kitabınızın bu ünitesinde sosyal medya hizmetleri içerisinde en çok kullanıcı kitlesine sahip olan uygulamalardan Twitter ve Facebook kullanıcı verilerine nasıl erişilebileceği ve R programı ile temel bazı analizlerinin nasıl gerçekleştirileceğine dair uygulamalara yer verilmiştir.

Kendimizi Sınayalım

- 1.** Diskler ve manyetik bantlar aşağıdaki dönemlerden hangisinde kullanılan teknolojilerdir?
 - a. Günümüz (Web Madenciliği)
 - b. 2000'ler (Veri Madenciliği)
 - c. 1990'lar (Veri Ambarı)
 - d. 1980'ler (Veri Erişimi)
 - e. 1960'lar (Veri Toplama)

- 2.** Veri madenciliği ile karşılaştırıldığında aşağıda verilen özelliklerden hangisi web madenciliği ile ilgili bir özelliktir?
 - a. Veri işleme süreci uzundur.
 - b. Çevrimdışı veriler kullanılır.
 - c. Kayıt dosyaları hariç, veri gizli değildir.
 - d. Geniş bir veritabanından tüm bilgi sağlanamaz.
 - e. Veriler veri ambarlarında saklanır.

- 3.** Aşağıdakilerden hangisi web madenciliği sürecinin temel adımlarından biri **değildir**?
 - a. Veri temizleme
 - b. Genelleştirme
 - c. Analiz
 - d. Kaynakların tespiti
 - e. Bilgi seçimi ve ön işleme

- 4.** Bir ağ üzerinde sunucu ile istemci bilgisayarlar arasında bilgi akışına aracı olarak görev gören ara sunuculara ne ad verilir?
 - a. Veritabanı
 - b. Vekil
 - c. İstemci
 - d. Veri ambarı
 - e. Sunucu

- 5.** Aşağıdakilerden hangisi içerik verilerinden biri **değildir**?
 - a. Resim
 - b. Metin
 - c. Ses
 - d. Kayıt (log)
 - e. Görüntü

- 6.** Aşağıdakilerden hangisi Web Yapı Madenciliği'nin uygulama alanlarından biridir?
 - a. Bilgi Keşfi
 - b. Kişiselleştirme
 - c. Sistem Geliştirme
 - d. Atif Analizi
 - e. Kısa Metin İşleme

- 7.** Örütü analizinde verilerin çok boyutlu veri küpleri hâlinde analiz edilmesine olanak sağlayan araç aşağıdakilerden hangisidir?
 - a. HTML
 - b. HITS
 - c. SQL
 - d. XML
 - e. OLAP

- 8.** Aşağıdakilerden hangisi Web kullanım Madenciliğinde veri ön işleme adımlarından biri **değildir**?
 - a. Kullanıcı bilgisinin belirlenmesi
 - b. İz(Yol) tamamlama
 - c. Web şeması modellemesi
 - d. Oturum bilgisinin belirlenmesi
 - e. Verinin temizlenmesi

- 9.** "Web bağlantı yapılarının modellenmesi" hangi web madenciliği sınıfının temel amacıdır?
 - a. Sosyal medya madenciliği
 - b. Web yapı madenciliği
 - c. Web profil madenciliği
 - d. Web içerik madenciliği
 - e. Web kullanım madenciliği

- 10.** Bir yazılımın başka bir yazılımda tanımlanmış fonksiyonlarını kullanabilmesi için uygulama oluşturmada kullanılan alt program, protokol ve araçlar bütününe ne ad verilir?
 - a. SQL (Structured Query Language)
 - b. WWW (World Wide Web)
 - c. API (Application Programming Interface)
 - d. HTML (Hypertext Markup Language)
 - e. OLAP (Online Analytical Processing)

Kendimizi Sınavalım Yanıt Anahtarı

- | | |
|-------|---|
| 1. e | Yanıtınız yanlış ise “Veri Madenciliği ve Web Madenciliği” konusunu yeniden gözden geçiriniz. |
| 2. c | Yanıtınız yanlış ise “Veri Madenciliği ve Web Madenciliği” konusunu yeniden gözden geçiriniz. |
| 3. a | Yanıtınız yanlış ise “Web Madenciliği Süreci” konusunu yeniden gözden geçiriniz. |
| 4. b | Yanıtınız yanlış ise “Web Madenciliği Veri Kaynakları” konusunu yeniden gözden geçiriniz. |
| 5. d | Yanıtınız yanlış ise “Web Madenciliği Veri Kaynakları” konusunu yeniden gözden geçiriniz. |
| 6. d | Yanıtınız yanlış ise “Web Yapı Madenciliği” konusunu yeniden gözden geçiriniz. |
| 7. e | Yanıtınız yanlış ise “Örütü Analizi” konusunu yeniden gözden geçiriniz. |
| 8. c | Yanıtınız yanlış ise “Veri Ön İşleme” konusunu yeniden gözden geçiriniz. |
| 9. b | Yanıtınız yanlış ise “Web Yapı Madenciliği” konusunu yeniden gözden geçiriniz. |
| 10. c | Yanıtınız yanlış ise “Sosyal Medya Madenciliği” konusunu yeniden gözden geçiriniz. |

Sıra Sizde Yanıt Anahtarı

Sıra Sizde 1

Web madenciliğinde internet ortamındaki yararlı bilgilerin elde edilme süreci sırasıyla kaynakların tespiti, bilgi seçimi ve ön işleme, genelleştirme ve analiz aşamalarından oluşmaktadır. Öncelikle araştırma konusu hakkında bilgi içeren web kaynakları belirlenir. Daha sonra ise bu kaynaklardan elde edilen dokümanlardan kullanılacak olan bilgilerin seçilmesi ve kullanıma uygun hale getirilmesi gereklidir. Bu bilgiler vasıtasiyla oluşturulacak kural veya örüntülerin genelde geçerli olup olmadıkları karşılaşmalar ile ortaya konduktan sonra ise analizlerle doğruluklarının onaylanması ve sonuçların yorumlanması gereklidir.

Sıra Sizde 2

Genel olarak internet ortamındaki veriler aşırı büyük boyutlarda, dağınık, yapılandırılmamış ve sürekli değişen bir yapıya sahiptir. Web madenciliği açısından internet verileri içerik, yapı, kullanım ve kullanıcı profil verisi üzere 4 gruba ayrılmaktadır.

Sıra Sizde 3

Web içerik madenciliğinde kullanılan temel yaklaşımlardan ilki kullanıcıların ilgi alanları ve kullanım karakteristiklerine göre arama motorlarındaki sonuçların geliştirilmesine ve web'in kişiselleştirilmesine yönelik olan bilgiye erişim yaklaşımı, diğer ise webde bilgi sorgulama ve karmaşık bilgilerin yönetiminin daha iyi yapılabilmesi için web verilerinin modellenmesine yönelik olan veritabanı yaklaşımıdır.

Sıra Sizde 4

Günümüzde farklı amaçlar için kullanılmakta olan birçok sosyal medya hizmeti bulunmaktadır. Bu hizmetler kullanım amaçları farklı olsa da yapı ve kullanım özellikleri bakımından 4 ortak özelliğe sahiptir. Bu hizmetler internet tabanlı ve kullanıcılardan kullanıcıya değiştirilebilen bir içeriğe sahiptirler. İnsanlar arasında çeşitli ihtiyaçlara cevap vermek amacıyla farklı gruplar oluşturmalarına imkan verilen bu hizmetlerin bilgi ve veri güvenliği, tasarım ve bakımı hizmet sunucusu tarafından sağlanmaktadır.

Yararlanılan ve Başvurulabilecek Kaynaklar

- Aldekhail M. (2016). **Application and Significance of Web Usage Mining in the 21st Century: A Literature Review**, International Journal of Computer Theory and Engineering, Vol. 8(1), 41-47.
- Danneman N., Heimann R. (2014). **Social Media Mining with R**, ISBN: 978-1-78328-177-0, Packt Publishing Ltd..
- Kaur S., Kaur K. (2015). **Web Mining and Data Mining: A Comparative Approach**, International Journal of Novel Research in Computer Science and Software Engineering Vol. 2(1), 36-42.
- Liu, B. (2007). **Web Data Mining: Exploring Hyperlinks, Contents and Usage Data**, ISBN 13: 978-3-540-37881-5, 532p, Springer.
- Markov Z., Larose D.T. (2007). **Data-Mining the Web : Uncovering Patterns in Web Content, Structure and Usage**, John Wiley & Sons, Inc., USA.
- Obar, J.A., Wildman, S. (2015). **Social Media Definition and The Governance Challenge: An Introduction to The Special Issue**, Telecommunications Policy, 39(9), 745-750.
- Sajja P.S., Akerkar R. (2012). **Intelligent Technologies for Web Applications**, ISBN: 13: 978-1-4398-7164-1, Chapman and Hall/CRC.
- Scime A. (2004). **Web Mining: Applications and Techniques**, ISBN: 1-59140-414-2, Idea Group Publishing.
- Sharma K., Shrivastava G., Kumar V. (2011). **Web mining: Today and tomorrow**, 3rd International Conference on Electronics Computer Technology, 399-403.
- Srivastava J., Cooley R., Deshpande M., Tan P.N. (2000). **Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data**, University of Minnesota, SIGKDD Explorations, Vol 1(2), 12-23.
- Xu G., Zhang Y., Li L. (2011). **Web Mining and Social Networking, Techniques and Applications**, (ISBN 978-1-4419-7734-2), Springer, New York.