

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

VERİ MADENCİLİĞİ YÖNTEMLERİ VE
OPTİMİZASYONA DAYALI MODELLER ÜZERİNE
BİR ARAŞTIRMA VE BİR UYGULAMASI

Şemsettin ERKEN

DANIŞMAN
Prof.Dr. Levent ŞENYAY

İZMİR - 2017

TEZ ONAY SAYFASI



YEMİN METNİ

Yüksek Lisans Tezi olarak sunduğum “Veri Madenciliği Yöntemleri ve Optimizasyona Dayalı Modeller Üzerine Bir Araştırma ve Bir Uygulaması” adlı çalışmanın, tarafımdan, akademik kurallara ve etik değerlere uygun olarak yazıldığını ve yararlandığım eserlerin kaynakçada gösterilenlerden oluştuğunu, bunlara atıf yapılarak yararlanılmış olduğunu belirtir ve bunu onurumla doğrularım.

....../....../2017

Şemsettin ERKEN

ÖZET

Yüksek Lisans Tezi

**Veri Madenciliği Yöntemleri ve Optimizasyona Dayalı Modeller Üzerine Bir
Araştırma ve Bir Uygulaması**

Şemsettin Erken

Dokuz Eylül Üniversitesi

Sosyal Bilimler Enstitüsü

Ekonometri Anabilim Dalı

Ekonometri Programı

Bilgi ve bilgisayar teknolojilerinde yaşanan gelişmelerle birlikte çok büyük miktarlarda verilerin toplanıp saklanması mümkün hale gelmiştir. Bu gelişmelere paralel olarak, büyük veri yığınlarını işlemek ve içerisinden değerli bilgiyi elde etmek için yeni yöntemlere ihtiyaç duyulmuş ve veri madenciliği kavramı ortaya çıkmıştır.

Veri madenciliğinde sık kullanılan tekniklerden biri de sınıflandırmadır. Sınıflandırma yöntemlerinin amacı, kavramlara ait niteliklerin aldığı değerler ile nitelikler arasındaki istatistiksel ve matematiksel örüntüden faydalanarak, araştırılan kavramların sınıf değeri tahminini yapmaktır.

Bu çalışmada veri, veritabanı, veri ambarı, veri madenciliği kavramlarıyla birlikte veri madenciliği yöntemleri açıklanmış, veri madenciliği yöntemlerinden sınıflandırma yöntemlerini incelenmiştir. Uygulama bölümünde, Spor Toto Süper Lig 2015/2016 futbol sezonunda, ligi ilk sırada tamamlayan Beşiktaş futbol takımının, ilgili sezondaki verileri kullanılmıştır. Maçların sonucu, Weka programıyla, naive bayes sınıflandırma algoritması ve C4.5 algoritması kullanılarak sınıflandırılmıştır. Uygulamanın sonucunda, naive bayes algoritması ile C4.5 algoritmasının sınıflandırma sonuçları karşılaştırılmış ve yorumlanmıştır.

Anahtar Kelimeler: Veri, Veri Madenciliği, Veri Madenciliği Yöntemleri, Sınıflandırma, Naive Bayes Algoritması, C4.5 Algoritması.

ABSTRACT
Master's Thesis
A Research On Data Mining Methods and Models Based On Optimization and
Its Application
Şemsettin ERKEN

Dokuz Eylül University
Graduate School of Social Sciences
Department of Econometrics
Econometrics Program

Improvements in information and computer technologies made it possible to collect and store in huge amounts of data. Parallel to these developments, new methods were needed in order to process large data stacks and obtain valuable information from within, as a result concepts of data mining emerged.

One of the frequently used techniques in data mining is classification. The purpose of classification methods is to make an estimate of the class values of concepts being explored by making use of statistical and mathematical pattern between attribute value of concepts and attributes.

In this study, data mining methods have been explained along with data, database, data warehouse and classification which is one of the data mining methods has been examined. As an application, the data of Beşiktaş football team which has completed the league on the first place in Spor Toto Super League 2015/2016 season were used. Results of matches were classified by using naive bayes and C4.5 classification algorithm via Weka data mining computer package. Classification results by naive bayes and C4.5 algorithms were also compared and interpreted.

Keywords: Data, Data Mining, Data Mining Methods, Classification, Naive Bayes Algorithm, C4.5 Algorithm.

**VERİ MADENCİLİĞİ YÖNTEMLERİ VE OPTİMİZASYONA DAYALI
MODELLER ÜZERİNE BİR ARAŞTIRMA VE UYGULAMA
İÇİNDEKİLER**

TEZ ONAY SAYFASI	ii
YEMİN METNİ	iii
ÖZET	iv
ABSTRACT	v
İÇİNDEKİLER	vi
KISALTMALAR	ix
TABLolar LİSTESİ	x
ŞEKİLLER LİSTESİ	xi

GİRİŞ	1
-------	---

**BİRİNCİ BÖLÜM
VERİ VE VERİ MADENCİLİĞİ**

1.1. VERİ KAVRAMI VE DOSYA YAPILARI	4
1.2. VERİTABANI KAVRAMI	5
1.3. İLİŞKİSEL VERİTABANI	6
1.4. VERİ AMBARI	6
1.4.1. Veri Ambarının Temel Özellikleri	6
1.4.2. Veri Ambarının İçerdiği Veriler	7
1.4.3. Veri Ambarı Mimarisi	8
1.4.4. Çevrimiçi Analitik İşleme ve Çevrimiçi Hareket İşleme	9
1.5. VERİTABANLARINDA BİLGİ KEŞFİ	10
1.6. VERİ MADENCİLİĞİ KAVRAMI	12
1.7. VERİ MADENCİLİĞİNİN KULLANIM ALANLARI	13
1.8. VERİ MADENCİLİĞİNDE KULLANILAN YAZILIMLAR	14
1.9. VERİ MADENCİLİĞİ SÜRECİ STANDARDI(CRISP-DM)	15

1.9.1. İŖi Tanımlama	16
1.9.2. Veriyi Tanımlama	16
1.9.3. Veriyi Hazırlama	17
1.9.4. Modelleme	18
1.9.5. Deęerlendirme	19
1.9.6. Uygulama	19
1.10. VERİ MADENCİLİęİ YÖNTEMLERİ	19
1.10.1. Sınıflandırma	20
1.10.2. Kümeleme	20
1.10.2.1. Uzaklık Ölçüleri	21
1.10.2.2. Hiyerarşik Kümeleme	22
1.10.2.3. Hiyerarşik Olmayan Kümeleme	22
1.10.3. Birliktelik Kuralları	23
1.10.3.1. Destek ve Güven Ölçütleri	24

İKİNCİ BÖLÜM

SINIFLANDIRMA YÖNTEMLERİ

2.1. KARAR AęAęLARI İLE SINIFLANDIRMA	25
2.1.1. ID3 Algoritması	30
2.1.2. C4.5 Algoritması	44
2.1.2.1. C4.5 Algoritması ve Bilinmeyen Nitelik Deęerleri	46
2.2. SINIFLANDIRMA VE REGRESYON AęAęLARI – GİNİ ALGORİTMASI	48
2.3. NAİVE BAYES SINIFLANDIRICISI	54
2.3.1. Naive Bayes Sınıflandırıcısında Nitelik Deęeri Türlerine Göre Olasılıkların Hesaplanması	58
2.3.1.1. Kategorik Nitelik Deęerleri	58
2.3.1.2. Sayısal Nitelik Deęerleri	61
2.3.2. Olasılık Deęerinin Sıfır Olması Sorunu	63
2.4. BAYES AęLARI	65
2.5. BELLEK TABANLI SINIFLANDIRMA	68

ÜÇÜNCÜ BÖLÜM

UYGULAMA

3.1. VERİ SETİNİN TANITILMASI	70
3.2. WEKA PROGRAMININ TANITILMASI	71
3.3. VERİ SETİNİN NAİVE BAYES ALGORİTMASI İLE SINIFLANDIRILMASI	77
3.4. VERİ SETİNİN C4.5 ALGORİTMASI İLE SINIFLANDIRILMASI	81
3.5. SINIFLANDIRMA ALGORİTMALARININ KARŞILAŞTIRILMASI İÇİN KULLANILAN PERFORMANS ÖLÇÜTLERİ	84
SONUÇ VE ÖNERİLER	87
KAYNAKÇA	89

KISALTMALAR

CRISP-DM	Cross-Industry Standard Process for Data Mining
ID3	Iterative Dichotomiser
OLAP	Online analytical Processing
OLTP	Online Transactional Processing
ROC	Receiver Operating Characteristic
SQL	Structured Query Language



TABLÖLAR LİSTESİ

Tablo 1: OLTP ve OLAP sistemler arasındaki farklar	s. 9
Tablo 2: Örnek Eğitim Kümesi	s. 28
Tablo 3: Aktivite Örneği İçin Eğitim Kümesi	s. 31
Tablo 4: Hava ve Sınıf Niteliği	s. 32
Tablo 5: Isı ve Sınıf Niteliği	s. 34
Tablo 6: Nem ve Sınıf Niteliği	s. 35
Tablo 7: Rüzgâr ve Sınıf Niteliği	s. 36
Tablo 8: Nitelikler ve Kazanç Ölçütleri	s. 37
Tablo 9: “HAVA” niteliği “Güneşli” Değerini Alan Kayıtlar	s. 38
Tablo 10: Nitelikler ve Kazanç Ölçütleri	s. 41
Tablo 11: “HAVA” Niteliği “Yağmurlu” Olan Kayıtlar	s. 42
Tablo 12: Sayısal Nitelik Değerleri	s. 45
Tablo 13: Eşik Değer ve Nitelik Değerleri	s. 45
Tablo 14: Eksik Veri	s. 47
Tablo 15: Eğitim Verisi	s. 49
Tablo 16: Gruplara Ayrılmış Eğitim Verisi	s. 50
Tablo 17: İkinci Dallanma İçin Eğitim Verisi	s. 52
Tablo 18: Gruplara Ayrılmış İkinci Düzüm Eğitim Verisi	s. 52
Tablo 19: Kategorik Veri Kümesi	s. 58
Tablo 20: Nitelik Frekansları ve Olasılıklar	s. 59
Tablo 21: Kategorik Veri Kümesi	s. 61
Tablo 22: Confusion Matrix Yapısı	s. 84
Tablo 23: Naive Bayes Algoritması Performans Ölçütü Değerleri	s. 86
Tablo 24: C4.5 Algoritması Performans Ölçütü Değerleri	s. 86
Tablo 25: Performans Kriterleri	s. 87

ŞEKİLLER LİSTESİ

Şekil 1: Veri Ambarı Mimarisi	s. 8
Şekil 2: Veritabanlarında Bilgi Keşfi Adımları	s. 10
Şekil 3: CRISP-DM Standardına Göre Veri Madenciliği Süreci	s. 15
Şekil 4: Basit Bir Karar Ağacı	s. 26
Şekil 5: Kök Düğümün Dallanması	s. 38
Şekil 6: ‘‘NEM’’ Düğümünün Dallanması	s. 41
Şekil 7: ‘‘Bulutlu’’ Dalı ve Yaprığı	s. 42
Şekil 8: ‘‘AKTİVİTE’’ Sınıf Niteliği İçin Karar Ağacı	s. 44
Şekil 9: İlk Dallanma	s. 51
Şekil 10: İkinci Dallanma	s. 53
Şekil 11: Karar Ağacının Son Hali	s. 54
Şekil 12: Bayes Ağı	s. 66
Şekil 13: Sigara, Radyasyon ve Kanser Ağı	s. 66
Şekil 14: 2015/2016 Sezonu Beşiktaş Futbol Takımı Verileri	s. 71
Şekil 15: Açılış Ekranı	s. 71
Şekil 16: Veri Seti Oluşturma	s. 73
Şekil 17: Veri Setinin Yüklenmesi	s. 74
Şekil 18: ‘‘Classify’’ Sekmesi	s. 75
Şekil 19: Sınıflandırma Algoritmaları	s. 76
Şekil 20: Veri Seti ve Sahip Olduğu Nitelikler	s. 78
Şekil 21: Niteliklerle İlgili Bilgiler	s. 79
Şekil 22: Sınıflandırma Sonucu	s. 80
Şekil 23: C4.5 Sınıflandırması	s. 81
Şekil 24: C4.5 Algoritması ile Oluşan Karar Ağacı	s. 82
Şekil 25: Karar Ağacı	s. 82
Şekil 26: C4.5 Sınıflandırması Sonuçları	s. 83

GİRİŞ

Bilgisayar kavramının hayatımıza girmesiyle birlikte, veri saklama ve işleme yöntemleri, ucu açık bir gelişmişlik içerisine girmiştir. Bilgisayar teknolojilerindeki gelişmeler, veritabanı kavramı ve teknolojileri ile birlikte çok büyük hacimlerdeki veri topluluklarını saklama konusundaki yetenekleri giderek geliştirmektedir. Veri miktarlarındaki bu artış, verilerin işlenmesinin, kontrolünün, yönetiminin ve analizinin zorlaşmasını da beraberinde getirmiştir. Böylelikle karar vericilere, çok büyük hacimlere ulaşılmış veri topluluklarından bilgi elde etmek konusunda bu teknolojiler yardımcı olmaktadır.

Veritabanlarındaki veri yığınlarından bilgiye ulaşmak için bir sürece ihtiyaç duyulmaktadır. Bu sürece veritabanlarında bilgi keşfi adı verilmektedir. Bu sürecin en önemli unsuru ve adımı veri madenciliğidir. Veri madenciliği, büyük veri yığınlarından gerçek, kullanışlı ve faydalı bilgiyi çıkarma süreci olarak tanımlanabilir. Veri madenciliği, sahip olduğu istatistik ve matematik gibi disiplinlerin bir arada kullanıldığı yöntemlerle verilerden birtakım örüntüler veya kural kalıpları elde ederek, karar destek sürecine katkı sağlayacak bilgiyi ortaya çıkardığı gibi araştırılan konuya yönelik tahminler de sunmaktadır.

Veri madenciliği yöntemleri ve algoritmalarının, ortaya koyulan yazılımlarla bilgisayar ortamında kolayca uygulanabilmesi, bu kavramın oldukça geniş bir kullanıcı ve araştırma alanına kavuşmasını sağlamıştır. Böylece bilginin artan önemine paralel olarak veri madenciliğinin önemi de artmıştır.

Çalışmanın ilk bölümünde, veri, veritabanı, veri ambarı ve veritabanlarında bilgi keşfi kavramlarının yanı sıra, veri madenciliği ve veri madenciliğinde kullanılan yöntemler açıklanmıştır.

İkinci bölümde ise birinci bölümde değinilmiş olan sınıflandırma kavramı çerçevesinde yer alan, oldukça sık bir kullanım alanına sahip olan algoritmalar kapsamlı olarak incelenmiştir.

Üçüncü bölüm olan uygulama bölümünde, Spor Toto Süper Lig 2015-2016 sezonunu şampiyon olarak tamamlayan Beşiktaş futbol takımının ilgili sezondaki, maç sonuçlarının ‘‘galibiyet’’, ‘‘beraberlik’’ ve ‘‘mağlubiyet’’ değerlerine sahip bir sınıf niteliği(değişkeni) olmak üzere toplam 11 niteliğe(değişkene) ait verileri

kullanılmıştır. İlgilenilen sezondaki maçların sonucu yani sınıf nitelik değeri, naive bayes algoritması ve C4.5 algoritması kullanılarak sınıflandırılmıştır. Uygulamanın sonucunda, bu iki algoritmanın sınıflandırma sonuçları karşılaştırılmış ve yorumlanmıştır.

LİTERATÜR TARAMASI

Veri madenciliğinde, sınıflandırma yöntemleri oldukça geniş bir uygulama alanına sahiptir. Sınıflandırma tekniklerinden, naive bayes sınıflandırma algoritmasının sahip olduğu kolaylık ve diğer avantajlı yönleri, farklı sınıflandırma tekniklerine üstünlük kurmuş ve tercih sebebi olmuştur.

Ling ve arkadaşlarının (2003), kullanılan sınıflandırma tekniğinin performans değerlendirmesinde eğri altında kalan alanın (Area Under The Curve=AUC), doğruluk değerinden daha iyi bir araç olduğunu gösterdiği çalışmada, AUC değeri açısından naive bayes sınıflandırıcısının, karar ağacı sınıflandırmasına göre çok daha iyi bir sonuç verdiği gösterilmiştir.

Zhang (2004), naive bayes sınıflandırıcısında, niteliklerin ve niteliklerin aldığı değerlerin koşullu bağımsızlığı varsayımının ihlali halinde yani güçlü bağımlılık durumunda da optimal çalıştığı ortaya koymuştur.

Panda ve Patra (2007), ağ tabanlı servislerde güvenlik ihlalleri tespiti için naive bayes sınıflandırıcısını kullanmış ve sinir ağı tabanlı yaklaşımı göre daha kısa sürede ve daha düşük faktör maliyetiyle daha yüksek tespit oranı elde etmişlerdir.

Bhargavi ve Jyothi (2009), naive bayes sınıflandırıcısını tarımsal alan topraklarını sınıflandırmak için kullanmıştır. Naive bayes yaklaşımının, diğer sınıflandırma tekniklerine göre sınıflandırma süresi, Kappa istatistiği, ortalama hata ve ortalama karesel hata değerleri açısından daha verimli bir sınıflandırıcı olduğu sonucuna ulaşmıştır.

Çatal ve arkadaşları (2011), java programlarında arıza tahmin sürecini kolaylaştırma amacıyla naive bayes algoritmasını kullanmışlar ve Eclipse tabanlı, yazılım arıza tahmin aracı geliştirmişlerdir.

Rajalakshmi ve Aravindan (2011), web sitelerin sınıflandırmasında naive bayes yaklaşımını kullanmışlardır.

Ting ve arkadaşları (2011), naive bayes sınıflandırıcısının doküman sınıflandırmasında, verimlilik ve doğruluk değerleri açısından sinir ağları, karar ağaçları ve destek vektör makineleri gibi sınıflandırıcılar arasından en iyi sınıflandırıcı olduğu sonucunu elde etmişlerdir.

Osmanbegovic ve Suljic (2012), öğrenci performanslarını tahminlemek için bir veri madenciliği çalışması yapmış ve naive bayes sınıflandırıcısı tahminlerinin, kullanılan diğer sınıflandırma metotlarının tahminlerinden daha üstün olduğu sonucuna ulaşmışlardır.

Balaji ve Srivatsa (2012), müşterilerin hayat sigortası ürünlerindeki tercihlerini tahminlemek için naive bayes algoritmasının kullanılmasının verimli sonuçlar verdiğini göstermişlerdir.

Ashari ve arkadaşları (2013), enerji simülasyon aracında alternatif bir dizayn araştırmasında, naive bayes algoritmasının, karar ağaçları ve en yakın k-komşu algoritmasından daha iyi sonuçlar verdiğini ortaya koymuşlardır.

Vembandasamy ve arkadaşları (2015), naive bayes algoritmasının kalp krizi teşhisinde kullanılmasının etkili sonuçlar verdiğini göstermiştir.

Rahangdale ve arkadaşları (2016), bankacılık ve sigorta alanında bir sınıflandırma çalışması yapmış ve en yakın k-komşu algoritmasına göre naive bayes algoritmasının daha doğru sonuçlar verdiğini ortaya koymuştur.

BİRİNCİ BÖLÜM

VERİ VE VERİ MADENCİLİĞİ

1.1. VERİ KAVRAMI VE DOSYA YAPILARI

Veri; ham, gerçek bilgi parçacığına verilen addır. Ölçüm, sayım, deney, gözlem ya da araştırma yolu gibi çeşitli yollarla elde edilmektedir. Sayısal bir değer bildiren verilere nicel veriler, sayısal bir değer bildirmeyen verilere de nitel veriler denir. Veriler, bir kavramı sembolize eden soyutlamalardır. Bir verinin tek başına bir anlamı yoktur ancak, belli bir mantıkla düzenlenerek elle ya da bilgisayarla işlenip bilgiye dönüştürüldüklerinde anlam kazanırlar. Böylece ilgili oldukları kavram için açıklama gücüne kavuşurlar. Elde edilen bilgiler de problem çözme ya da karar verme gibi bir amaç doğrultusunda kullanıma hazır hale gelmektedir.

Veri saklama birimlerinde depolanan veri topluluklarına dosya adı verilir. Dosyalar ise kendi içerisinde kayıtlara bölünmektedir(Özkan, 2008: 12). Örneğin; bir şirketin çalışanlarının listesi bir dosya, her bir çalışana ait bilgilerin toplamı bir kayıttır. Çalışan kayıtlarında bulunan çalışan adı, soyadı, birimi gibi bilgilerin her birine de alan denilmektedir.

Saklanan dosyalar ardışık ya da doğrudan erişim yöntemiyle kullanılmaktadır. Ardışık dosyalara ilk kayıttan son kayda kadar sıralı bir şekilde erişim sağlanabilirken, doğrudan erişim yönteminde ise istenilen kayıtlara direkt olarak ulaşılmaktadır.

Doğrudan erişim yöntemiyle tasarlanmış dosyalardan birisi dizinli dosyalardır. Bu tür dosyalarda, veri dosyası için bir dizin dosyası tasarlanır. Bir dosya için oluşturulan dizinde, ilgili dosyanın anahtarları ve anahtarlar için tanımlanan kayıtlar bulunur. Anahtar, erişim için seçilen alandır. Örneğin; bir öğrenci dosyasında, öğrenci numarası dizin alan ise, herhangi bir öğrencinin bilgilerine ulaşmak için bu numara dizin üzerinde bulunur ve bu dizin üzerindeki girilen öğrenci numarasına ilişkin kayıtlara erişilir(Özkan, 2008: 13).

Dosya yapılarından bir diğeri de hesaba dayalı dosyalardır. Bu tür dosyalarda bir hesaplama işlemi kullanılmaktadır. Dosyadaki istenilen bir kayda hesaplama işlemi yardımıyla doğrudan ulaşılır.

1.2. VERİTABANI KAVRAMI

Verilerin elde edilmesi, araştırma, deney ya da gözlem yapmak adına büyük önem taşımaktadır. Bu noktada, toplanmış olan verilerin kullanım amacına uygun bir şekilde saklanması da en az verilerin toplanması kadar önemlidir.

Birbiriyle ilişkili olan verilerin, kullanım amacına uygun olarak sistemli bir şekilde düzenlenmiş olduğu, yönetilebilir bilgi depolarına veritabanı adı verilmektedir.

Veritabanı kavramı, geleneksel veri saklama yöntemi olan dosya-işlem işleyişinin, artan veri boyutlarıyla birlikte çok sayıda dosya ilişkisi ve erişim konusunda zamanla verimini yitirmeye başlamasından sonra daha sistemli ve daha büyük boyutlarda veri saklama ve erişim anlamında daha etkin işlem yapma ihtiyacı ile birlikte ortaya çıkmıştır.

Veritabanı, birbiri ile ilişkili olan verilerin sahip oldukları ilişkisel ve boyutsal anlamları da içermektedir. Veritabanında depolanmış olan verilerin, ilişkisel yapısından faydalanılarak amaca uygun derlenmiş olması, büyük bir veri topluluğu içerisinde istenilen bilgiye doğrudan erişim olanağı sunmaktadır. Böylece veritabanı, veritabanı kullanıcıları için zaman ve işlem açısından etkin bir araç kimliği taşımaktadır.

Veritabanı sistemleri, klasik dosyalama sistemine göre gelişmiş özellikler taşımaktadır. Bu özellikler şu şekilde sıralanabilir;

-) Veri tekrarına izin vermez
-) Bilgiye erişim için yapılmış bazı tanımlamalar sayesinde verinin tutarlılığını sağlar
-) Birden fazla erişim talebi karşısında sorun yaşanmaz
-) Belirlenmiş bazı verilere, kısmen veya tamamen erişimin engellenmesi için oluşturulacak bazı kısıtlamalarla verinin güvenliğini sağlar.

1.3. İLİŞKİSEL VERİTABANI

İlişkisel veritabanı, her biri özel isimlere sahip tablolardan oluşur. Burada her tablo bir varlığa veya ilişkiye karşılık gelmektedir. Tablonun sütunları nitelikleri, satırları ise bu niteliklerin değerlerini ifade eder. Her bir satır kayıt olarak da düşünülebilir. Anahtar alan tablonun tanımlayıcısıdır(Özkan, 2008: 17).

Varlık-ilişki(entity-relationship) veri modeli gibi anlamsal bir veri modeli genellikle ilişkisel veritabanı olarak inşa edilir. Bir varlık-ilişki veri modeli, veritabanını, varlıklar ve varlıkların ilişkileri kümesi olarak gösterir(Han ve Kamber, 2006: 10).

1.4. VERİ AMBARI

Veri ambarı, veritabanı ile ilişkili bir kavram olmasının yanında, veri tabanlarının, barındırdıkları üstün özelliklere rağmen karar unsurlarına katkı sağlama konusunda yetersiz kalmaya başlamasının ve daha hızlı bir erişim ihtiyacının ardından ortaya çıkmış bir kavramdır.

Veri ambarı; her bir veri biriminin zamanın bir anıyla ilgili olduğu, karar destek fonksiyonlarına katkı yapmak amacıyla tasarlanmış, bütünleşik ve konu odaklı veritabanları toplamıdır(Kantardzic, 2011: 14). Tanımda belirtildiği üzere, oluşturulmasında farklı veritabanlarından faydalanılmasından dolayı farklı merkezlerinden elde edilen verilerin aynı ortamda analiz edilebilmesini de sağlamaktadır.

1.4.1. Veri Ambarının Temel Özellikleri

Veri ambarının ilk karakteristiği konuya yönelik olmasıdır. Bu özellik, veri ambarının işlem yerine konulara göre bölünebilmesi gerektiği anlamına gelmektedir(Giudici, 2003: 20). Örneğin; üretim yapan bir şirketin yaptığı ödemeleri, işleme yönelik bir sistem, ham madde ödemeleri, çalışan ödemeleri ve vergi ödemeleri şeklinde saklarken veri ambarı bu verileri ödemeler konusu altında saklamaktadır.

Veri ambarının en önemli karakteristiği bütünlük olmasıdır. Veri ambarının barındığı verilerin tanımlamalarındaki tutarlılık, bütünlük olma durumu olarak gösterilebilir. İşleme yönelik sistemlerde, uygulamalardaki aynı kavrama, aynı cins nesneye ait veriler için farklı tanımlamalar olabilir. Tanımlamama farklılıkları barındıran bu veriler, veritabanlarından veri ambarına aktarılmadan önce gerekli düzenlemelerle, tek tanımlama çatısı altında toplanmalı ve bunun sonrasında veri ambarına aktarılmalıdır. Örneğin; herhangi bir veritabanında, sınav sonucunu içeren bir alanda, sonuçlar “başarılı” veya “başarısız” olarak tanımlanmış olabilir. Başka bir veritabanında, aynı alanın başarı durumu için “pozitif”, başarısızlık durumu için “negatif” değeri kodlanmış olması muhtemeldir. Bahsedilen iki veritabanından sınav sonucu alanına ait veriler, veri ambarına aktarılmadan önce, veri ambarında sınav sonucu alanı nasıl tanımlanmış ise o tanımlama şekline dönüştürülmelidir.

Veri, geçmişe dayalı bir perspektiften bilgi sağlamak için depolanır (Örneğin; geçmiş 5-10 yıl). Veri ambarındaki her anahtar yapı, dolaylı veya doğrudan zamanın bir unsurunu içerir(Han ve Kamber, 2006: 106). İşlemsel sistemlerdeki verileri yeni verilerdir. Veri ambarının zaman boyutunun olmasından dolayı veri ambarındaki veriler ise geçmiş zamana ait verilerde içermektedir. Bu yüzden veri ambarı, işleme yönelik sistemlerinin aksine geçici olmamakla birlikte güncelleştirilmemiştir.

Veri ambarının temel fonksiyonu, karar destek unsurlarına katkı sağlamaktır. Bu amaçla tasarlanmış yapıdan dolayı, veriler üzerinde, işleme yönelik sistemlerin aksine sadece veriye erişim söz konusudur.

1.4.2. Veri Ambarının İçerdiği Veriler

Veri ambarı, farklı bir yapıya sahiptir. Veri ambarı yapısı, yeni ayrıntı veri, eski ayrıntı veri, düşük özetlenmiş veri, yüksek özetlenmiş veri ve meta-data bileşenlerini içerir(Ma ve diğerleri, 2000: 126).

Meta-data, veri ambarında önemli ve özel bir rol oynar. Basitçe, veri hakkında veri olarak tanımlanır. Meta-data verilerin nasıl oluşturulduğunu, verilere erişildiğini ve verilerin nasıl kullanıldığının yanı sıra işlem verisinin yapısını ve anlamını tanımlayan veridir(Ma ve diğerleri, 2000: 126). Veri ambarında meta-data olmadığında kullanıcı, hangi verinin olduğunu ya da olmadığını anlamak için bir

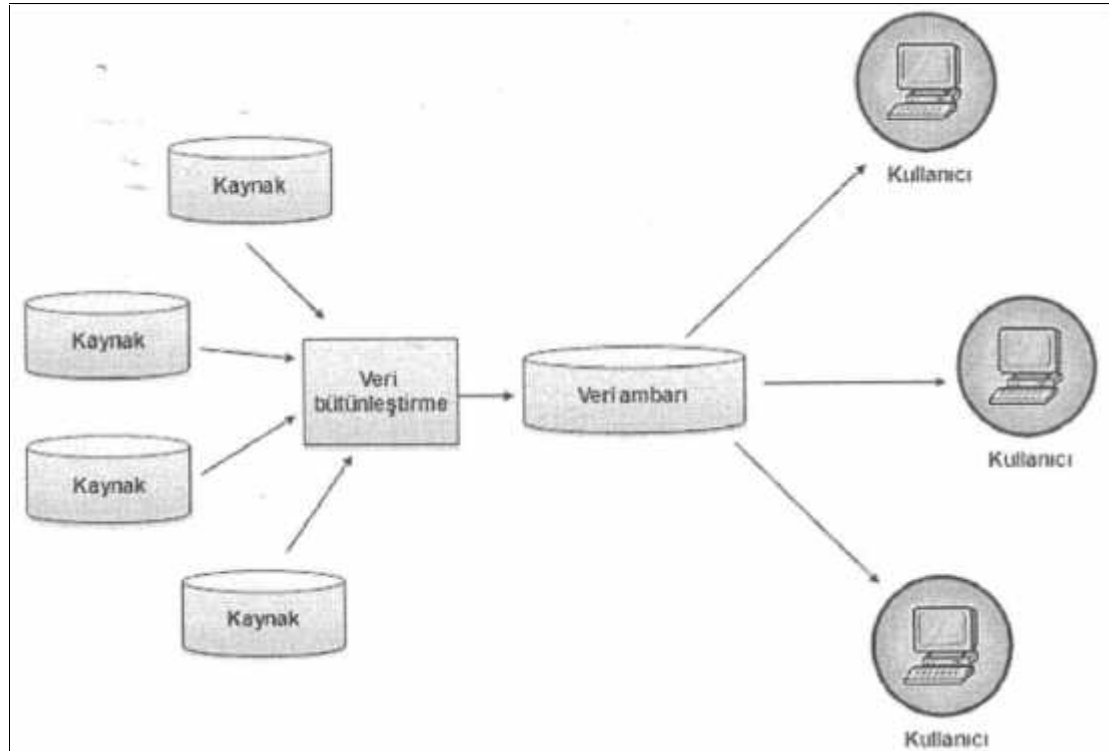
alıřma yapıp ardından, analizi gerekleřtirme ařamasına geebilir. Bu sorun, byk bir zaman kaybına yol amaktadır. Meta-data bu sorunu ortadan kaldırmaktadır(Inmon, 2002: 113).

Yeni ayrıntı veri, řu andaki ya da řu an ve belirli bir zaman periyodunu da iine alabilen dneme ait iřlenmemiř veridir. Eski ayrıntı veri, yeni ayrıntı verinin ilgilendiėi zaman periyodundan veya andan daha nceki zamanlara ait ayrıntı ieren enformasyonları barındırmaktadır. Yeni ayrıntı veriden, dřk derecede iřlenerek elde edilen verilere dřk zetlenmiř veri, yksek derecede zetlenmiř verileri de yksek zetlenmiř veri adı verilmektedir.

1.4.3. Veri Ambarı Mimarisi

Veri ambarı, oluřumunda bir mimariye sahiptir. Bu mimari řekil 1’de gsterilmiřtir. Farklı veri kaynaklarından elde edilen veriler, veri ambarının yapısına uygun olarak dnřtrlp veri ambarına yklenir. Ykleme iřleminin ardından, veri ambarı ile kullanıcıları arasında eriřim saėlanır.

řekil 1: Veri Ambarı Mimarisi



Kaynak: zkan, 2008: 31.

1.4.4. Çevrimiçi Analitik İşleme ve Çevrimiçi Hareket İşleme

OLAP, çevrimiçi analitik işleme, veri işleme süreci için önem taşıyan bir kavramdır. OLAP araçları, birçok boyutta hesaplama özetleri ve tanımlamaların SQL'den üstün olan, çok boyutlu veri analizi sağlamaya odaklanmışlardır(Koyuncugil ve Özgülbaş, 2009: 24). Diğer bir deyişle, OLAP, çeşitli kategoriler altında toplanmış verileri işleme ve çok boyutlu analiz etmeye olanak sağlayan fonksiyonlar kümesidir(Ma, ve diğerleri, 2000: 127).

Veri ambarları, kullanıcılarına karar verme noktasında yardım sağlama amacıyla tasarlandığı için OLAP araçları da aynı amaç doğrultusunda olan görselleştirme araçlarıdır.

OLTP, çevrimiçi hareket işleme, verilerin günlük olarak işlendiği sistemlerdir. Örneğin; Bir şirketin muhasebe, stok, kayıt ve satış gibi birçok günlük işlemlerini yürütebildiği bu tip sistemlerde, veri ekleme veya silme gibi işlemler yapılabilir.

Tablo 1: OLTP ve OLAP sistemler arasındaki farklar

Karakteristik	OLTP	OLAP
Değişkenlik	Dinamik veri	Statik veri
Zamansallık	Sadece yeni veri	Yeni ve eski veri
Öğe boyutu	Detaylandırılmış veri	Birikmiş ve birleştirilmiş veri
Güncelleme	Sürekli ve düzensiz	Periyodik ve düzenli
Hareketler	Tekrarlı	Tahmin edilmeyen
Esneklik	Düşük	Yüksek
Performans	Sorgu başına birkaç saniye	Karmaşık sorgularda düşebilir
Kullanıcılar	Çalışanlar	Karar vericiler
Kullanım amacı	İşlem	Karmaşık sorgu ve karar destek
Metrikler	İşlem oranı	Etkin karşılık

Kaynak: Vercellis, 2009:49.

Veri seçimi, çalışmanın veya organizasyonun amacına göre bilgi keşfi yapılacak verilere karar verip, bu verilerin, bir veritabanı içerisinde saklanmasını içeren aşamadır. Bilgi keşfinin sonuna kadar yapılacak tüm uygulamalar, bu veriler üzerinden yapılacağı ve uygun olmayan bir veri grubunun doğru sonucu veya doğru bilgiyi sağlayamayacağı için, veri seçimi aşaması çok önemli bir aşamadır.

Veri önışleme aşaması, bir önceki aşamada oluşturulan veritabanında depolanan verilere, birtakım işlemlerin yapıldığı ilk aşamadır. Bu aşamada, kullanılacak verilerin, farklı kaynaklardan elde edilmiş olması bazı sorunlara yol açabilmektedir. Bu sorunlardan bazıları, tanımlama ve kodlama uyumsuzlukları, verilerin farklı zamanlara ait olma durumlarıdır(Akpınar, 2000: 9). Bu aksaklıkların giderilmesinin ardından, veriler arasında, yanlış veri girişinden, bir olayın nadiren ortaya çıkmasından kaynaklı sapan verilerin olması ya da anlamsız bazı değişkenlerin var olması durumunda, nitelikli bir bilgi içermeyen verilerin temizlenmesi diğer bir deyişle silinmesi işlemi gerçekleştirilir(Akpınar, 2000: 10).

Veri indirgeme aşaması, görev hedefine bağlı veriyi temsil edecek faydalı özellikleri bulmaktır(Koyuncugil ve Özgülbaş, 2009: 23). Bu aşama, yapılacak olan işlemlerle, veri madenciliğine daha kaliteli veri hazırlama, boyut ve ele alınan değişkenlerin indirgenebildiği aşamadır(Gemici, 2012: 6). Ele alınan veritabanında çok fazla kayıt ve gerekli olmayan değişkenlerin olması durumunda veri indirgeme işlemi yapılır(Dondurmacı ve Çınar, 2014: 261).

Veriler hazır hale geldikten sonra, veri madenciliği adımına geçilmektedir. Veriler üzerinde anlamlı ve yararlı örüntüler çıkarılabilmek için kullanılacak olan modelin ve algoritmanın belirlenmesi bu aşamada gerçekleşmektedir(Erdoğan, 2004: 16). İlk olarak amaca uygun bir veri madenciliği yöntemine karar verilir. Veri madenciliği yöntemlerine örnek olarak birliktelik kuralları, kümeleme ve sınıflandırma verilebilir. Seçilen yöntemle ilgili ve amaca göre bir algoritma belirlenir. Veriler bu algoritma ile analiz edilir. Analiz işlemi, araştırılan konu ile ilgili istenen noktaya gelene kadar devam eder. Elde edilen bilgiler, geriye dönük kontrol veya gelecekte yapılacak çalışmalar için bir karşılaştırma unsuru olarak raporlanabilir, ayrı bir veri saklama ünitesinde depolanabilir.

1.6. VERİ MADENCİLİĞİ KAVRAMI

Veri ve veritabanı sistemlerindeki büyük gelişmeler sonucunda, eskiye oranla, devasa miktarlarda verinin depolanması ve erişimi oldukça kolaylaşmıştır. Daha büyük veri yığınlarının saklanabilmesi, veritabanındaki verilerin daha karmaşık yapıda olmasını beraberinde getirmiş, verilerin analizini ve bu veriler arasından anlamlı, kullanışlı bilgiler çıkarmak zorlaşmıştır. Bu noktada, büyük yığınlardan bilgi elde etmek için yeni yöntem ve araçlara ihtiyaç duyulmuştur. Böylece veri ambarlarında bulunan verileri analiz eden ve bu analizler sonucunda veriler arasındaki gizlenmiş olan örüntüleri, yapıları elde eden, yapıcı, kullanışlı ve değerli bilgileri sunan veri madenciliği kavramı ortaya çıkmıştır. Veri madenciliği kavramı, bu karakteristiği ile veritabanlarında bilgi keşfi sürecinin en önemli adımıdır.

Veri madenciliği kavramı için farklı tanımlamalar yapılmıştır. Bu tanımlamalardan bazıları aşağıdaki gibidir;

-) Veri madenciliği; veritabanı kullanıcısına, ilk başta bilinmeyen düzenlemeleri veya ilişkileri bulmak amacıyla net ve kullanışlı sonuçlar sağlamak için büyük miktarda verinin, seçme, keşfetme ve modelleme sürecidir(Giudici, 2003: 2).
-) Veri madenciliği; geniş hacimli verinin içerisinden, yeni, değerli ve çözülmesi zor olan bilgi sürecidir(Kantardzic, 2011: 2).
-) Veri madenciliği; büyük veritabanlarının analizi ve bilgi çıkarımı doğrultusunda, karar verme ve problem çözme ile ilgilenen unsurlara, kesin ve kullanışlı bilgiler sunan aşamalı süreçler kuran aktivitelerdir(Vercellis, 2009: 77).

Veri madenciliği, analiz ve bilgi elde etme işlevlerini ortaya koyarken istatistik, matematik, yapay zekâ ve makine öğrenmesi gibi disiplinlerden faydalanmaktadır. Bu açıklamadan yola çıkarak anlaşılabacağı üzere, veri madenciliği disiplinler arası bir alandır. Veri madenciliğinin faydalandığı ve benzerlikleri bulunan makine öğrenmesi gibi, eğitim kümesi olarak tanımlanan veri sistemlerinden faydalanılarak önceki durumlardan ve sonuçlardan, işlemlerin nasıl yapılacağı ve araştırılacak yeni durumlarla ilgili genellemeler yapılması sürecidir(Ay, 2009: 6).

1.7. VERİ MADENCİLİĞİNİN KULLANIM ALANLARI

Veri madenciliği, faydalandığı disiplinlerde ve bilgi, bilgisayar ve veritabanı teknolojilerinde yaşanan gelişmelerle birlikte daha geniş bir uygulama alanına ulaşmıştır. Günlük yaşamdan, kurumsal ve bilimsel yaşamın en küçük kısmına kadar her alanda, veri madenciliğine başvurulabilmekte ve etkin sonuçlar alınabilmektedir. Veri madenciliğinin kullanıldığı bu alanlardan bazıları, aşağıda belirtilmiştir;

-) Pazar yönetimi; hedef pazarlama, müşteri ilişkileri yönetimi, Pazar sepet analizi, çapraz satış, Pazar bölümlenmesi(Sumathi ve Sivanandam, 2006: 27).
-) Sigortacılık; yeni poliçe talep edebilecek müşterilerin tahmin edilmesi, riskli müşteri örüntülerinin tespiti, sigorta dolandırıcılıklarının tespiti(Akpınar, 2000: 5).
-) Borsa; hisse senedi fiyatlarının tahmin edilmesi, genel piyasa analizleri, hisse tespiti, alım ve satım stratejilerinin optimize edilmesi(Özçınar, 2006: 10).
-) Biyoloji, tıp, genetik ve ilaç: tıbbi teşhis, tedavi sürecinin belirlenmesi, ürün geliştirme, test sonuçlarının yorumlanması(Şık, 2014: 39).
-) Bankacılık; finansal göstergeler arasında gizli örüntülerin tespiti, kredi kartı dolandırıcılıklarının tespiti, kredi kartı kullanıcılarının harcamalarına göre gruplanması, kredi ve kredi kartı taleplerinin değerlendirilmesi(Akpınar, 2000: 4).
-) Spor; müsabaka sonuçlarının tahmini, antrenman tekniklerinin verimlilik değerlendirmesi ve belirlenmesi, spor dallarına eğilimin değerlendirilmesi, müsabaka sonuçlarına etki eden parametre ve değişkenler arasındaki gizli ilişkilerin belirlenmesi ya da belirlenmiş ilişkilerin etki gücünün ortaya konması.

Veri madenciliği 90'lı yıllardan itibaren önemini her geçen gün arttırmış ve daha geniş alanlarda uygulanmıştır. Yukarıda belirtilen alanların dışında, eğitim, perakendecilik, internet ve güvenlik, endüstri ve daha birçok alanda veri madenciliği uygulamaları yapılmaktadır.

1.8. VERİ MADENCİLİĞİNDE KULLANILAN YAZILIMLAR

Veri madenciliği için çeşitli özelliklere sahip yazılımlar mevcuttur. Bu yazılımlardan bazıları aşağıda belirtilmiştir;

-) SPSS clementine; İngiltere’ de geliştirilen bu yazılım güncellemelerle sürekli iyileştirilmektedir. Algoritma açısından güçlü yazılımlar arasında yer alırken, grafik özelliği açısından diğer yazılımlara göre daha zayıftır. Veri kalite kontrolü, zengin algoritma içeriği ve ortalama bir kullanım kolaylığına sahip olan bu yazılım, en çok kullanılan yazılımlardandır(Ay, 2009: 25).
-) SAS enterprise miner; veri madenciliği yazılımları arasında en çok bilinenlerdendir. Karar ağacı, birliktelik kuralları, regresyon, kümeleme gibi veri madenciliği sorgularına sahiptir. Grafik anlamında oldukça iyi olan bu yazılım, kullanım açısından ortalama üstü bir zorluk taşımaktadır(Ay, 2009: 26).
-) WEKA; Waikato Üniversitesi tarafından geliştirilmiş bu yazılım, geniş bir veri madenciliği sorgu yelpazesine sahiptir. Temel veri madenciliği yöntem ve algoritmalarının yanı sıra araştırma açısından daha özel yaklaşımlar, araştırma konularına özgü algoritmalar da barındırmaktadır. Veri önışleme ve görselleştirme açısından oldukça verimli bir programdır. Bu yazılım için geliştirilmiş arff formatında dosyalar ile çalışmaktadır. Anlaşılır olma ve kullanım kolaylığı açısından, en kolay yazılımlar arasındadır. Bahsedildiği gibi, kullanım açısından arayüzünün kolaylığı, yapılan işlemlerin görselleştirilmesi anlamında üstün olması ve geniş bir algoritma ve sorgu kapasitesine sahip olmasından dolayı bu çalışmada yapılacak uygulama için Weka programı kullanılacaktır.

Açıklanan bu yazılımlar dışında, taşıdığı özellikler, desteklediği dosya formatları, araştırma konusu özelinde verimlilik, kullanım kolaylığı gibi durumlara göre tercih edilen başka yazılımlar da mevcuttur. Bu yazılımlardan; Darwin, Excel, Oracle data mining, Angoss, Statistica data miner, RapidMiner, MATLAB ve R yazılımları örnek olarak verilebilir.

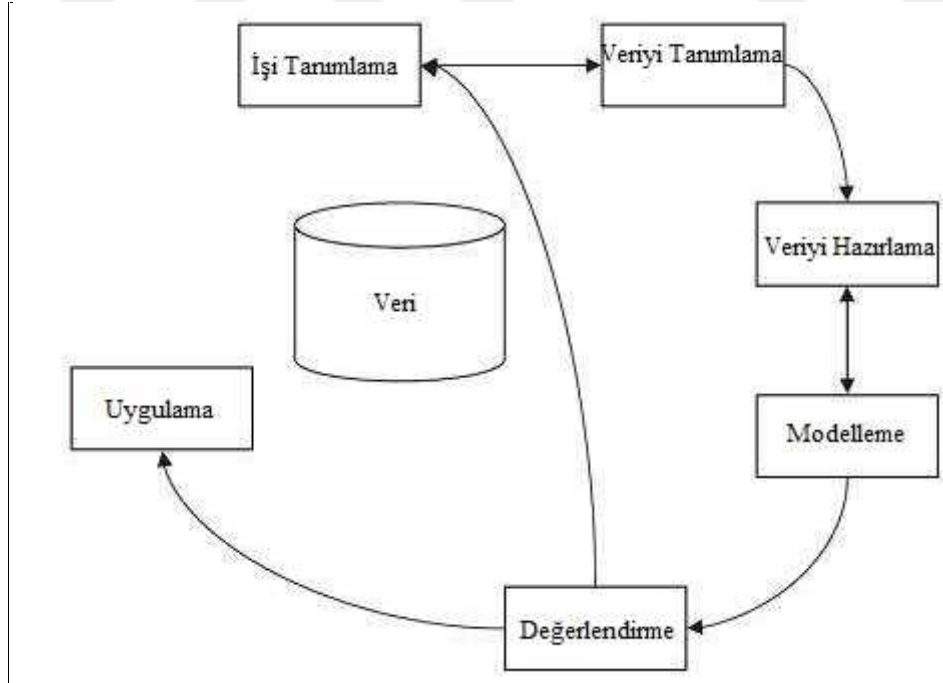
1.9. VERİ MADENCİLİĞİ SÜRECİ STANDARDI(CRISP-DM)

Veri madenciliği işlemi, veri üzerinde anlık olarak uygulanmayan, tekrarlanan bir işlemdir. Bundan dolayı veri madenciliği, bir süreç olarak ele alınmalıdır(Kocabaş, 2010: 13).

CRISP-DM, DaimlerCrysler, SPSS, NCR.CRISP kuruluşları tarafından 1996 yılında geliştirilmiş bir veri madenciliği süreci standardıdır. Bu standart, işlem ve araştırma ünitesi için veri madenciliğini, genel bir problem çözme stratejisi haline getirmektedir(Larose, 2005:5). Geliştirilen süreç standardı sayesinde, veri madenciliği projelerinin, başarısının ya da başarısızlığının, büyük oranda uygulayıcılara bağlı olması sorunu ortadan kalkmıştır. Böylece veri madenciliği süreci, daha güvenilir, daha az maliyetli, daha kolay tekrarlanabilir ve uygulanabilir bir süreç haline gelmiştir(Şık, 2014: 59).

CRISP-DM standardına göre veri madenciliği süreci, 6 aşamadan oluşmaktadır.

Şekil 3: CRISP-DM Standardına Göre Veri Madenciliği Süreci



Kaynak: Kocabaş, 2010: 14.

Şekil 3'te gösterildiği gibi aşamalar birbirleri ile etkileşim halindedir. Bundan dolayı bir aşamanın çıktısı, bir sonraki aşamanın girdisi durumundadır;

-) İşi tanımlama
-) Veriyi tanımlama
-) Veriyi hazırlama
-) Modelleme
-) Değerlendirme
-) Uygulama

1.9.1. İşi Tanımlama

Veri madenciliği çalışmalarında başarılı olmanın en önemli şartı, projelerin hangi amaç için yapılacağı ve elde edilecek sonuçların başarı düzeylerinin nasıl ölçüleceğinin tanımlanmasıdır(Savaş ve diğerleri, 2012: 8). Bu aşamada, projenin amaçları ve gereksinimleri açıkça belirtilmelidir. Amaçlar ve kısıtlamalar, veri madenciliği problem tanımına uygun olarak formüle edilmeli ve belirlenen amaçlara ulaşmak için stratejik bir hazırlık yapılmalıdır(Larose, 2005: 6).

1.9.2. Veriyi Tanımlama

Veri madenciliği projelerinde işin ve amacın tanımlanmasından sonra verinin tanımlaması adımına geçilir. Veriyi tanımlama aşaması, verinin toplanması, verinin ve veri kalitesinin anlaşılması için keşfedici veri analizi kullanımı ve grafiksel yöntemlerle verilerin genel yapısının incelenmesi faaliyetlerini içerir(Kıvrak, 2010: 11). Veriler toplandıktan sonra, toplanan veri sayısı, verilerin fiziksel ve niteliksel özellikleri belirlenir. Ardından, verilerin özellikleri arasındaki bağıntılar araştırılır ve bu araştırma için istatistiksel metotlardan faydalanabilir.

Bir veritabanında, veriler arasında yanlış ölçüm veya veri girişini yapan kullanıcın yanlış veri girmesi sonucu ortaya çıkan sorunlu verilere gürültülü veri, bir veri niteliğinin herhangi bir sebepten dolayı bir değere sahip olmadığı verilere ise kayıp veri denir. Veri tanımlama aşamasının son evresinde verinin sayısal ve

kavramsal olarak yeterliliği ve kalitesi hakkında bilgiler toplanır. Eğer kayıp veya gürültülü veri gibi sorun içeren veriler varsa ise bu veriler için belirlemeler yapılır.

1.9.3. Veriyi Hazırlama

Veri hazırlama aşaması, bu aşama sonucu oluşacak olan veriler, seçilecek araştırma modelinde kullanılacaktır. Bu yüzden, veri hazırlama, veri madenciliği süreci standardının çok önemli bir adımıdır.

Bu adımda verilerin proje amacına uygunluğu, kalitesi, hacmi ve türü gibi kısıtlar altında, araştırma kullanılacak olan veri grubu seçilir. Seçilen veri grubu içerisinde kayıp veri veya gürültülü veri gibi bir takım sorunlar var ise, uygun yöntemlerle kayıp verilerin tamamlanması ve veriler arasındaki gürültünün azaltılması veya ortadan kaldırılması işlemleri gibi veri temizlemesi işlemi yapılır (Şık, 2014: 64).

Veri temizleme işlemi yapıldıktan sonra aynı alana ilişkin verilerde, farklı veritabanlarından toplanarak bir veri ambarı oluşturulmasından dolayı, tanımlama ya da kodlama farklılıkları gibi bir takım tutarsızlıkları engellemek için veri bütünleştirilmesi yapılır.

Veri madenciliği uygulamalarında kullanılan veri depoları çok fazla değişken içerebilir ve değişkenlerden bazıları arasında doğrusal ilişkiler olabilir. Örneğin; müşteri numarası ile müşteri kimliği değişkenleri, kayıtlar için aynı sonuçları verir dolayısıyla bu iki değişken arasında doğrusal ilişki mevcuttur. Bu değişkenler, korelasyon analizi ile tespit edilip, bu tür bir ilişkiye sahip değişkenlerden biri silinir. Veri indirgeme adı verilen bu işlem, veri hazırlama aşamasında yapılır (Dondurmacı, 2011: 9).

Veri hazırlama aşamasında yapılan işlemlerinden birisi veri dönüştürmedir. Veri dönüştürme, normalleştirme ve genelleme gibi işlemlerinin en az birini içine alan evredir (Tüzüntürk, 2010: 69). Veri dönüştürme yöntemlerine, minimum-maksimum normalleştirilmesi, Z skor normalleştirilmesi ve ondalık ölçekleme örnek olarak verilebilir (Oğuzlar, 2003: 73). Dönüştürme işleminin amacı; farklı ölçek ya da tanımlamaların yol açtığı problemleri ortadan kaldırmaktır. Kaynak olarak kullanılan veritabanlarının birinde ağırlık ölçüsü kilogram olarak tanımlanmışken, diğer

veritabanında aynı kavramın pound olarak kaydedilmesi, bu konuya örnek olarak verilebilir (Dondurmacı ve Çınar, 2014: 262).

Veri hazırlama, ilgilenilen projeye yönelik analiz için kullanılacak modeli kurmak ve uygulamak kadar önemlidir. Veri madenciliği standardındaki adımların etkileşimli olması, dolayısıyla veri hazırlama aşamasında oluşturulan verilerin, kullanılacak modele aktarılacak olması, bu adımın ne kadar önemli olduğunu göstermektedir. Verilerin yapısında olabilecek ve bu başlıkta bahsedilmiş sorunlar çözüme ulaştırılamaması, kullanılacak modelin ve çalışma sonucunda keşfedilecek bilgilerin, doğruluğu ya da etkinliği anlamında gerçekçi olmayan hatta geçersiz ve yanlış sonuçlar ortaya çıkacaktır.

1.9.4. Modelleme

Belirlenen amacı ortaya koyacak en uygun modelin belirlenmesi için çok sayıda model denemesi gerekmektedir. Bundan dolayı, veri hazırlama ve modelleme adımları, en iyi modele ulaşana kadar tekrarlanan bir süreçtir(Akpınar, 2000: 10).

Model kurulumunda, denetimli ve denetimsiz öğrenmeye göre farklı adımlar izlenmektedir(Çelik, 2009: 11). Denetimli öğrenme; veriden yararlı bilgi çıkarmaya yönelikken, denetimsiz öğrenme veri tanımlamaya ve keşfetmeye yöneliktir(Koyuncugil ve Özgülbaş, 2009: 26). Denetimli öğrenmeye tahmin edici bir karakteristik taşırken, denetimsiz öğrenme, tanımlayıcı bir özellik barındırır. Öğrenme yöntemleri içerisindeki adımlar farklılaşsa da model kurma aşamasındaki genel kalıp aynıdır. Araştırma için hazır halde olan veri topluluğu, eğitim grubu ve test grubu olmak üzere iki alt sete bölünür(France ve diğerleri, 2002: 246). Eğitim grubu verileri kullanılarak model elde edilir. Elde edilen modelin doğruluğu, bir başka deyişle kalitesi, test grubu ile ölçülür. Farklı tekniklerle oluşturulan modeller arasından, amaca, projeye ya da ilgilenilen problemin yapısının en uygun model seçilir. Model kurulduktan sonra, gerekli ise veri hazırlama aşamasına dönülüp değişiklikler yapılabilir.

1.9.5. Değerlendirme

Oluşturulan modelin, uygulama safhasına geçmeden önceki, her anlamda değerlendirildiği son aşamadır. Modelin, belirlenmiş olan amaca uygunluğu ve amaca ulaşma noktasında etkinliği, ilgilenen problemi ele alırken, probleme tüm yönleriyle bir çözüm getirip getiremediği ve modelin doğruluk oranı, kalitesi, yeterliliği gibi konuların ele alınıp, değerlendirildiği aşamadır. Eğer model incelenen konularda yeterli bulunursa uygulama aşamasına geçilir(Ay, 2009: 13).

1.9.6. Uygulama

Bu safhada, kurulan model ilgilenilen proje veya problem doğrultusunda uygulanır. Geçerliliği kabul edilen modelin, bir uygulamanın alt uygulaması olarak kullanılması da mümkündür(Ay, 2009: 13). Seçilen modelin değişkenleri veya kaynak aldığı bazı kavramlarda zamanla değişimler olabilir. Bu sebepten dolayı, modelin izlenmesi de bu aşamada yapılan işlemlerden biri olarak düşünülmelidir.

1.10. VERİ MADENCİLİĞİ YÖNTEMLERİ

Veri madenciliği yöntemleri, temelde tanımlayıcı ve tahmin edici modeller olarak ikiye ayrılmaktadır. Modelleme başlığında açıklandığı gibi, tahmin edici modeller, verileri kullanarak bir model tasarlanıp, bu modelden faydalanarak ilgilenen probleme ilişkin verilerin sonuçlarını ortaya koyan modellerdir. Tanımlayıcı modeller ise veriler arasındaki gizli ilişki ve yapıları araştırmaya yönelik modellerdir(Akpınar, 2000: 5).

Veri madenciliği, birçok yöntem ve algoritma içermektedir. Kullanılan yöntemlerin birçoğu istatistiksel tabanlıdır. Yöntemlerden bazıları ise yapay zekâ kaynaklıdır(Dondurmacı ve Çınar, 2014: 263). Bu çalışmada, veri madenciliğinde kullanılan üç ana yöntem incelenmiştir. Bu yöntemler, aşağıda sıralanmıştır;

-) Sınıflandırma
-) Kümeleme
-) Birliktelik kuralları

1.10.1. Sınıflandırma

Sınıflandırma, veritabanındaki nesnelerin, niteliklerine bağlı olarak, sınıflarını belirleyen fonksiyon ya da modeldir. Bir nesne veya veri seti, eğitim grubu olarak ayrıldığında, eğitim grubunda, her nesne ait oldukları sınıf değerleri ile birlikte bir nitelik vektörü tarafından gösterilir. Sınıflama modeli veya fonksiyonu da, eğitim grubundaki her nesnenin sınıfları ile nitelikler arasındaki ilişkinin analizi ile inşa edilir. Böylelikle oluşturulan model ya da fonksiyon, araştırılacak yeni verileri sınıflandırabilecek, veritabanındaki verileri anlama kabiliyetini arttıracak ve daha da kolaylaştıracaktır(Sumathi ve Sivanandam, 2006: 205).

Literatürde, istatistiksel analiz yöntemlerinden regresyon yöntemi ile birlikte yer bulan bu yöntem, regresyondan farklıdır. Regresyon, nümerik niteliklerinin tahmini ile ilgilenirken sınıflandırma modelleri, kategorik hedef niteliklerin değerlerini tahmini ile ilgilenir. Diğer bir fark; sınıflandırma modellerinin denetimli öğrenme tekniklerinden biri olmasıdır. Sınıflandırma modelleri, geçmişte gözlenmiş ve sınıf nitelik değeri belli olan veri grubundan faydalananarak gelecekteki gözlemlerin sınıf nitelik değerlerini tahmin edebilmek adına sınıflandırma kuralları oluşturmak için kullanılır(Vercellis, 2009: 221).

Sınıflandırma yöntemi, veri madenciliği uygulamalarında yaygın olarak başvurulmuş bir yöntemdir. Bankacılık, sağlık, satış-pazarlama ve sigortacılık gibi gündelik, kurumsal veya bilimsel yaşamın neredeyse her evresinde kullanılabilmektedir.

1.10.2. Kümeleme

Kümeleme yöntemi, veri elemanlarının birbirine olan benzerliklerinden faydalanılarak, benzer verileri diğerlerinden ayırıp kategorize etme prensibine dayanmaktadır. Sınıflama modelinden farklı olarak kümelemede veriler veya veri gruplarının kesin bir sınıf üyeliği yoktur. Sınıflama işleminde verilerin hangi sınıfa dâhil olduğu bilinir fakat kümelemede bu bilgi kesin değildir(Çavuşlar, 2008: 26).

Kümeleme yönteminin sınıflama işleminden bir diğer farkı da kümeleme yönteminin amacı sınıflama yapmak değildir. Bu nedenle kümeleme yapmak için

hedef bir değişken yoktur. Böylece yöntem, heterojen halde bulunan tüm veriyi homojen alt kümelerine ayırır.

Kümeleme yönteminde, bellek tabanlı sınıflandırmada olduğu gibi uzaklıklardan faydalanılır. Veriler arasındaki uzaklıklardan yararlanarak, alt kümeler içi uzaklıkların minimizasyonu, alt kümeler arası uzaklıkların maksimizasyonu amaçlanır.

Veri madenciliğinin sık başvurulan yöntemlerinden olan kümeleme yöntemi, kendi içinde yöntemlerine ayrılmaktadır. Bu araştırma özelinde hiyerarşik ve hiyerarşik olmayan kümeleme yöntemlerinden bahsedilecektir.

1.10.2.1. Uzaklık Ölçüleri

Gözlemler arası uzaklıklara ilişkin yaklaşımlardan bazıları bellek tabanlı sınıflandırma başlığında bahsedilmişti. Yinelemek gerekirse uzaklık hesaplamak için kullanılan bağıntılar şu şekildedir.

Öklid uzaklığı; 2 boyutlu uzayda Pisagor teoreminin uygulanmasından ortaya çıkmaktadır(Özkan, 2008: 133).

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

Manhattan uzaklığı; gözlemler arasındaki uzaklığın mutlak değerinden faydalanılan uzaklık bağıntısıdır.

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

Minkowski uzaklığı; p sayıda değişken için gözlem değerleri arasındaki uzaklık hesabı için kullanılan bağıntıdır.

$$d(i, j) = \left[\sum_{k=1}^p (|x_{ik} - x_{jk}|^m) \right]^{\frac{1}{m}} \quad i, j=1, 2, \dots, n; k=1, 2, \dots, p$$

1.10.2.2. Hiyerarşik Kümeleme

Bu yöntemde, kümelerin temel bir küme olarak değerlendirilip sonrasında aşamalı olarak içerdiği alt kümelere ayrılması veya oluşturulmuş kümelerin yine aşamalı olarak tek bir küme oluncaya kadar birleştirme temeline dayanmaktadır.

Hiyerarşik kümeleme yöntemleri çerçevesinde kullanılan algoritmalarından biri en yakın komşu algoritmasıdır. Bu algoritmada, ilk olarak gözlem değerleri arasındaki uzaklıklar, istenen uzaklık bağıntısıyla hesaplanır. Hesaplanan uzaklıklardan en az olanı belirlenip, en az uzaklığa sahip olan gözlemler bir küme içinde birleştirilir. Birleştirme işleminin ardından, yeni durumdayken tekrar gözlemler arası uzaklık hesaplaması yapılır. Bu adımlar, gözlem değerlerinin tek bir küme altında birleşimine kadar tekrarlanır. Birkaç gözlemden oluşan iki küme arasındaki uzaklık hesaplanmasında ise iki kümenin birbirine en yakın gözlemleri arasındaki uzaklık, iki küme arası uzaklık olarak kabul edilmektedir.

1.10.2.3. Hiyerarşik Olmayan Kümeleme

Bu tür kümeleme yönteminde, bir küme merkezi seçilir. Merkezden daha önce öngörülmüş eşik değeriyle tüm veriler bir araya getirilerek bir grup oluşturulur. Ardından yeni bir küme merkezi seçilerek kümelenmiş noktalar için süreç tekrarlanır (Erdoğan, 2004: 39). Hiyerarşik olmayan kümeleme yöntemlerinin yaygın olarak kullanılan algoritmalarından biri, k-ortalamlar algoritmasıdır.

Bu algoritma, uygulaması basit ve yaygın, iteratif, gözlemleri kullanıcının belirlediği küme sayısı olan k adedince kümelere ayıran bir kümeleme algoritmasıdır (Wu ve Kumar, 2009: 22). Algoritmanın adımları aşağıdaki gibidir. İlk olarak küme sayısı olan k belirlenir. Ardından belirlenen $C_1, C_2, \dots, C_k$ kümelerine gözlemler atanır. Her kümenin merkezi aşağıdaki formülle belirlenir.

$$M_k = \frac{1}{n_k} \sum_{i=1}^{n_k} X_{ik} \quad n: \text{küme gözlem sayısı}$$

Kümelerin merkezleri bulunduğundan sonra küme içi değişimler olan $e_1, e_2, \dots, e_k$ ve küme içi değişimlerin toplamı olan E_k^2 , her kümenin gözlem değerleri ile kümenin merkezi arasındaki Öklid uzaklıkları toplamıdır ve kare hata şu şekilde bulunur.

$$e_i^2 = \sum_{i=1}^{n_k} (x_{ik} - M_k)^2$$

$$E_k^2 = \sum_{k=1}^K e_k^2$$

Kümeler bir önceki durumla aynı oluncaya kadar adımlar tekrarlanır. Bu yöntemin amacı, kare hatayı minimize eden k kümelerini bulmaktır (Özkan, 2008: 149).

1.10.3. Birliktelik Kuralları

Olayların birlikte gerçekleşme durumlarını analiz eden ve bu durumları belirli olasılıklarla ortaya koyan veri madenciliği yaklaşımına birliktelik kuralları adı verilmektedir (Alan, 2012: 167).

Bu yaklaşım, veriler arasındaki birliktelik ya da korelasyonları elde etmede kullanılmaktadır. Böylelikle veriler arasındaki ilk bakışta fark edilemeyen örüntüler bulunabilmektedir. Ortaya koyulan örüntü bilgisi özellikle pazarlama, satış, karar verme ve yönetim gibi alanlarda kullanılmaktadır.

Birliktelik kurallarının en çok kullanıldığı alanlardan biri de market ve satış sektörüdür. Pazar sepet analizi ismi verilen teknik, bu alanda oldukça sık kullanılmaktadır. Bu teknikte, birlikte satın alınmış ürünler arasındaki ilişki tespit edilip birlikte satılma durumları fazla olan ürünlerin yan yana olan raflarda

sunulması, stok kontrolü, satın alma alışkanlıklarının ortaya koyulması gibi amaçlarla düzenlemeler yapılarak satışların artması sağlanabilir.

Örneğin; bir marketten çay satın alan müşterilerin %60 olasılıkla şeker alıyor olması bilgisi, iki ürün arasındaki örüntü bilgisi ve birliktelik kuralıdır. Bu bilgiyle market yöneticileri, çay ve şeker reyonlarının yan yana olmaları yönünde karar verip satışların artırılması amaçlanabilir.

1.10.3.1. Destek ve Güven Ölçütleri

Ürünler arasındaki ilişkileri ortaya koyarken destek ve güven ölçütlerinden faydalanılmaktadır. Genel olarak çok sayıda veriden oluşan veri grupları ile çalışıldığı için birçok sayıda birliktelik kuralı elde edilebilir. Bundan dolayı elde edilen bağıntılar arasından bazıları doğru bir tahminleme konusunda yetersiz kalabilir. Bu aşamada elde edilen bağıntı sayısını azaltmak ve geçerliliği olan bağıntıları belirlemek için destek ve güven ölçütleri kullanılır.

Elde edilen bağıntıların destek ve güven ölçütleri hesaplanmadan önce minimum destek ve güven değerleri belirlenir. Hesaplanan destek ve güven değerleri belirlenen minimum değerlerden küçükse o bağıntı göz ardı edilir(Çoşkun, 2010: 18).

Bir bağıntının destek ölçütü, bağıntının tüm alışverişler içinde hangi oranda tekrarlandığını, güven ölçütü ise A ürünü satın alan müşterilerin B ürününü de satın alma olasılıklarını diğer deyişle birliktelik kuralını ortaya koymaktadır(Özkan 2008:157). Bu birliktelik kuralı $A \rightarrow B$ şeklinde gösterilmektedir. Buradan hareketle, N yapılan tüm alışveriş sayısı ve sayısı(A, B); A ve B ürünlerini birlikte içeren alışveriş sayısı olmak üzere; bağıntının destek ve güven ölçütleri şu şekilde hesaplanır.

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

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

İKİNCİ BÖLÜM

SINIFLANDIRMA YÖNTEMLERİ

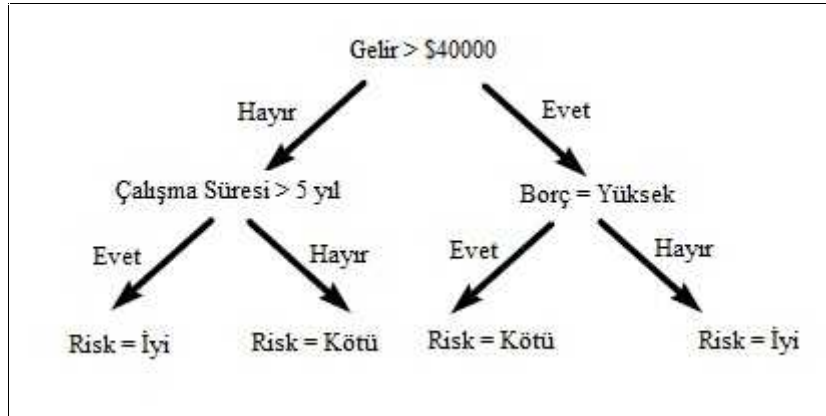
2.1. KARAR AĞAÇLARI İLE SINIFLANDIRMA

Veri madenciliğinde, sınıflandırma yapmak için yaygın olarak kullanılan tekniklerden birisi de karar ağaçlarıdır. Karar ağaçları; bir dizi karar kuralı uygulayarak, büyük miktarlarda veri topluluklarını, niteliklerinin aldığı değerlere ve bu değerlerin benzerliklerine ve yakınlıklarına göre daha küçük setler haline getirmeye yarayan yapılardır(Berry ve Linoff, 2004: 166).

Karar ağaçlarının yapısı; düğüm, dal ve yaprak denilen kavramlardan oluşmaktadır. Düğümler, ilgilen sınıflandırma konusunun niteliklerini temsil eder. Dallar ise bölünmelere uğrayan düğümleri birbirine bağlayan bağlantı elemanlarıdır. Bölünmeler tamamlandığında, her bir kaydın sınıf değerini veya sonucunu gösteren yapı elemanlarına ise yaprak adı verilir. Bir karar ağacının ilk düğümüne kök düğüm adı verilir. Karar ağaçlarının işleyişi bu düğümdeki bölünmeler ile başlamaktadır. Kök düğüm ile yapraklar arasında kalan düğümlere ara düğümler adı verilir(Rokach ve Maimon, 2010: 149). Bu yapıyla karar ağaçları akış şemalarıyla benzerlikler taşımaktadır.

Bu başlıkta bahsedildiği gibi karar ağaçları, sahip olduğu avantajlardan dolayı veri madenciliği ve sınıflandırma uygulamalarında sıkça kullanılan bir tekniktir. Karar ağaçlarının kurulumu ucuz ve kolaydır. Bu kolaylık, ağaç yapısının anlaşılmasını, yorumlanmasını ve işleyişini karmaşık olmaktan kurtarır. Başka veritabanı sistemlerine basit bir şekilde entegre edilebilmektedirler. Karar ağaçlarının, diğer sınıflandırma tekniklerine göre daha iyi olduğu en önemli özelliği ise görselleştirmedir. Karar ağaçları, hiçbir kısıtlama olmadan hem kategorik hem de nümerik verileri kullanabilir(Gorunescu, 2011: 183). Bu yöntem, genelde veriler için bir ön işleme sürecine ihtiyaç duymaz ve yüksek boyutlu veri yapılarıyla da çalışma imkânı sunmaktadır(Gargano ve Raggad, 1999: 85).

Şekil 4: Basit Bir Karar Ağacı



Kaynak: Edelstein, 1999: 15.

Şekil 4’te gösterilen karar ağacı, önceden edinilmiş olan bir miktar kaydın gelir, çalışma süresi, borç niteliklerinin ve risk sınıf niteliğinin aldığı değerlere göre eğitim yaparak oluşturulmuştur. Kredi isteyen bir kişinin, krediyi ödeme konusunda taşıdığı riskin tahmin edildiği bu karar ağacında gelir, çalışma süresi ve borç niteliği, ağacın düğümleridir. Düğümlere ait olası sonuçları ve düğümleri arasındaki bağlantıyı gösteren “hayır” ve “evet” değerleri ise karar ağacının dallarıdır. “Risk = İyi” ve “Risk = Kötü” sonuçları, sınıf değerleridir ve bu sonuçlar karar ağacının yapraklarını oluşturmaktadırlar. Daha önceden elde edilmiş veriler ışığında tasarlanan bu karar ağacı ile karar kuralları elde edilmiş ve bu sayede gelecekte aynı niteliklerle risk sınıfı değeri araştırılmak istenen bir kişinin sahip olduğu risk durumu tahmin edilebilecektir.

Şekil 4’teki karar ağacına göre; 40000 doların üzerinde geliri ve yüksek miktarda borcu olan kişiler ile 40000 dolardan daha az gelire ve 5 yıldan daha az çalışmış olan kişilerin risk değeri kötü olarak belirlenir. Öte yandan, 40000 doların üzerinde geliri ve düşük miktarda borcu olan kişiler ile 40000 dolardan daha az gelire ve 5 yıldan daha fazla çalışmış olan kişilerin risk değerleri de iyi olarak belirlenir.

Bu başlıkta anlatıldığı gibi karar ağaçları, daha önceden depolanan verilerle eğitim yaparak kök düğümden itibaren niteliklerin sahip olduğu olası sonuçlara göre bölünmesi ile oluşmaktadır. Karar ağaçları kullanılarak, ilgilenilen araştırma konusunda etkin tahminler veya sonuçlar elde etmek için bölünmenin ya dallanmanın

nasıl yapılacağı konusu, belirlenen her dallanma kuralı için farklı bir karar ağacı ortaya çıkacağı için çok önemlidir(Özkan, 2008: 54).

Entropi, karar ağaçlarında dallanma kuralının oluşturulmasında yararlanılan kavramlardan birisidir. Sınıflandırma yapmak için kullanılan karar ağacı algoritmalarının bazıları entropi tabanlıdır. Bir sistemdeki belirsizlik ölçüsü entropi olarak tanımlanabilir. X ; n adet mesaj üreten bir kaynak ve birbirinden bağımsız olarak üretilen her mesajın üretilme olasılığı P_i olmak üzere, X kaynağının entropisi aşağıdaki gibi hesaplanır(Özkan, 2008: 55).

$$H(X) = - \sum_{i=1}^n P_i \log_2 (P_i)$$

Eğitim grubu olarak belirlenmiş kayıt topluluğunu, sınıf niteliğinin aldığı değerlere göre ayırmak için sınıflarla ilgili bir miktar bilgiye ihtiyaç duyulur. Karar ağaçları söz konusu bu bilgi miktarını elde etmek için entropiden yararlanır.

Örneğin; kâr-zarar durumunu gösteren SONUÇ= {pozitif, pozitif, pozitif, negatif, negatif, negatif, negatif, negatif} kümesi, eğitim verisi olarak belirlenmiş bir veri topluluğunun sınıf niteliğidir. Sınıf niteliğinin aldığı değerler açısından eğitim grubundaki kayıtlar “pozitif” ve “negatif” olmak üzere iki sınıfa ayrılacaktır. P_1 ve P_2 sırasıyla “pozitif” ve “negatif” değerlerine ait olasılık değerleri olmak üzere; bu olasılıklar her bir sınıf değerinin, sınıf nitelik kümesindeki frekansının, sınıf niteliğindeki toplam eleman sayısına bölünmesi ile elde edilir(Quinlan,1993: 21).

$$P_i = \frac{\text{Sınıf kümesindeki ilgili sınıf değeri frekansı}}{\text{Sınıf Kümesi eleman sayısı}}$$

Sınıf 1; sınıf nitelik değeri “pozitif”, sınıf 2 sınıf nitelik değeri “negatif” olan sınıflar olmak üzere; ele alınan “Sonuç” sınıf kümesi sekiz elemanlı bir kümedir. Sınıf 1 frekansı, yani eleman sayısı 3’tür. Aynı şekilde sınıf 2 frekansı da 5’tir. Buradan hareketle, $P_1 = 3/8$, $P_2 = 5/8$ olarak hesaplanır. Olasılık değerleri hesapladığı için “Sonuç” kümesinin entropisi rahatlıkla aşağıdaki gibi hesaplanabilir.

$$H(\text{SONUÇ}) = - \left[\left(\frac{3}{8} \right) \log_2 \left(\frac{3}{8} \right) + \left(\frac{5}{8} \right) \log_2 \left(\frac{5}{8} \right) \right] = 0,9544$$

Eğitim grubunda, sınıf niteliğinin başka bir deyişle hedef niteliğin dışındaki nitelikler kendi içerisinde farklı değerler alırlar. Herhangi bir X niteliğinin değerlerine göre sınıf niteliği alt setlere ayrılırsa; hedef niteliğin bir elemanının sınıfını tayin etmek için gerekli bilgi miktarı, her bir alt setin bir elemanının sınıf değeri için gerekli bilgi miktarının ağırlıklı ortalamasıdır şeklinde tanımlanabilir(Quinlan, 1993: 22).

Sınıf niteliğin eleman sayısı T, sınıf nitelik dışındaki bir X niteliğinin aldığı değerler göre alt setlere bölündükten sonra her bir alt setin eleman sayısı T_i olmak üzere; herhangi bir X niteliğine göre entropi şu formül ile hesaplanır:

$$H(X, T) = \sum_{i=1}^n \frac{T_i}{T} \times H(T_i)$$

Tablo 2: Örnek Eğitim Kümesi

SATIŞ	SONUÇ
Çok	Pozitif
Çok	Pozitif
Az	Negatif
Az	Negatif
Çok	Pozitif
Az	Negatif
Çok	Negatif
Çok	Negatif

8 kayıttan oluşan eğitim kümesinde, “GELİR” niteliği “çok” ve “az” olmak üzere iki farklı değer almıştır. Bu nitelik aldığı değerlerin farklılığına göre iki alt gruba ayrılabilir. Söz konusu niteliğin alt gruplarının eleman sayısı;

$$|GELİR_{\text{çok}}| = 5$$

$$|GELİR_{\text{az}}| = 3 \text{ tür.}$$

“GELİR” niteliğinde “çok” değerini almış kayıtlarının sınıf nitelik değerlerinin 3’ü “pozitif”, 2’si “negatif”, aynı şekilde “az” değerini almış kayıtlarının sınıf nitelik değerlerinin ise hepsi “negatif” olarak karşılık bulmuştur. Böylece alt gruplara yönelik entropiler aşağıdaki gibi hesaplanır;

$$H(\text{GELİR}_{\text{çok}}) = - \left[\left(\frac{3}{5} \right) \log_2 \left(\frac{3}{5} \right) + \left(\frac{2}{5} \right) \log_2 \left(\frac{2}{5} \right) \right] = 0,97$$

$$H(\text{GELİR}_{\text{az}}) = - \left[\left(\frac{0}{3} \right) \log_2 \left(\frac{0}{3} \right) + \left(\frac{3}{3} \right) \log_2 \left(\frac{3}{3} \right) \right] = 0$$

“GELİR” niteliğinin, “çok” ve “az” dallanmalarından elde edilen bilgi miktarı yani entropisi hesaplamaların ardından, ilgili nitelik için entropi şu şekilde bulunur;

$$H(\text{GELİR, SONUÇ}) = \left(\frac{5}{8} \right) H(\text{GELİR}_{\text{çok}}) + \left(\frac{3}{8} \right) H(\text{GELİR}_{\text{az}})$$

$$H(\text{GELİR, SONUÇ}) = \left(\frac{5}{8} \right) \times 0,97 + \left(\frac{3}{8} \right) \times 0 = 0,60$$

T veritabanını herhangi bir X niteliğine göre ayrıştırarak elde edilen bilgi miktarlarını ölçüp karşılaştırmak için kazanç ölçütü kullanılır(Özkan, 2008: 58). Kazanç ölçütü sıradaki formülle hesaplanır:

$$\text{Kazanç}(X,T) = H(T) - H(X,T)$$

Tablo 2’deki “GELİR” değişkenin kazanç ölçütü, formül yardımıyla aşağıdaki gibi bulunur;

$$H(\text{SONUÇ}) = - \left[\left(\frac{3}{8} \right) \log_2 \left(\frac{3}{8} \right) + \left(\frac{5}{8} \right) \log_2 \left(\frac{5}{8} \right) \right] = 0,9544$$

$$\text{Kazanç}(\text{GELİR, SONUÇ}) = H(\text{SONUÇ}) - H(\text{GELİR, SONUÇ})$$

$$\text{Kazanç}(\text{GELİR, SONUÇ}) = 0,9544 - 0,60 = 0,3544$$

2.1.1. ID3 Algoritması

Entropiye dayalı dallanmada, kazanç ölçütü çok önemli bir yer tutmaktadır. Karar ağacı oluşturmak için kullanılan algoritmalarından biri olan ID3 algoritması, Quinlan tarafından geliştirilmiş olan entropi tabanlı algoritmalarından biridir. Bu algoritma, nominal verilerle çalışır ve tüm düğümler net bir şekilde belirlenene kadar veya dallanması yapılmamış değişken kalmayıncaya kadar devam eder(Duda ve diğerleri, 2001: 411). Karar ağaçları ile araştırılmak istenen problemlerin çözümünde oldukça sık kullanılmaktadır. Söz konusu algoritmayı geliştiren Quinlan, ID3'ten sonra sayısal verilerle de karar ağacı oluşturabilen daha gelişmiş algoritmalar da ortaya koymuştur.

Basit ve aşamalı bir işleyişi olan ID3 algoritmasının temel amacı; ağacın kök düğümünü ve kök düğümden itibaren tüm ara düğümler ile dallanmalarını ortaya koymaktır. ID3 algoritmasını kullanarak oluşturulan karar ağaçlarında dallanmanın başlayacağı düğüm, kazanç ölçütü ile saptanır. Sınıf niteliği dışındaki tüm niteliklerin kazanç ölçütü hesaplandıktan sonra, maksimum kazanç ölçütüne sahip nitelik belirlenir. En büyük kazanç ölçütüne sahip nitelik kök düğüm olarak seçilir ve ilk dallanma gerçekleştirilir. Ardından, aynı mantık çerçevesinde, sınıf niteliği ve ilk dallanmada seçilmiş olan nitelik dışında kalan nitelikler için tekrar kazanç ölçütü hesaplanır ve maksimum kazanç ölçütüne sahip nitelik seçilip sıradaki dallanma gerçekleştirilir. Eğitim verileri ile oluşturulmuş karar ağaçlarıyla karar kuralları elde edilir. Araştırma ile ilgili yeni verilere, karar kuralları uygulanarak ilgili kayıtların sınıf değerleri belirlenir.

Tablo 3: Aktivite Örneği İçin Eğitim Kümesi

HAVA	ISI	NEM	RÜZGÂR	AKTİVİTE
Güneşli	Sıcak	Yüksek	Yok	Olumsuz
Güneşli	Sıcak	Yüksek	Var	Olumsuz
Bulutlu	Sıcak	Yüksek	Yok	Olumlu
Yağmurlu	Ilık	Yüksek	Yok	Olumlu
Yağmurlu	Soğuk	Normal	Yok	Olumlu
Yağmurlu	Soğuk	Normal	Var	Olumsuz
Bulutlu	Soğuk	Normal	Var	Olumlu
Güneşli	Ilık	Yüksek	Yok	Olumsuz
Güneşli	Soğuk	Normal	Yok	Olumlu
Yağmurlu	Ilık	Normal	Yok	Olumlu
Güneşli	Ilık	Normal	Var	Olumlu
Bulutlu	Ilık	Yüksek	Var	Olumlu
Bulutlu	Sıcak	Normal	Yok	Olumlu
Yağmurlu	Ilık	Yüksek	Var	Olumsuz

Kaynak: Quinlan, 1986: 86.

Tablo 3’te Quinlan tarafından düzenlenmiş eğitim kümesiyle, ID3 algoritması kullanılarak karar ağacı uygulaması yapılacaktır.

Aktivite sınıf niteliği “olumlu” ve “olumsuz” olmak üzere iki farklı değer almıştır. Sınıf1 “olumlu”, sınıf2 “olumsuz” değerlere sahip sınıflar ve |Sınıf1|; sınıf1’in, |Sınıf2|; sınıf2’nin eleman sayısı ayrıca P_1 ve P_2 sırasıyla sınıflara ait olasılıklar olmak üzere; toplam 14 kayıt içerisinde, “olumlu” değeri 9, “olumsuz” değer 5 adettir.

$$|Sınıf1| = 9$$

$$|Sınıf2| = 5$$

Böylece $P_1 = 9/14$ ve $P_2 = 5/14$ ’tür. Buradan aktivite sınıf niteliği için entropi, $H(AKT)$; aşağıdaki gibi hesaplanır:

$$H(AKT) = - \left[\left(\frac{9}{14} \right) \log_2 \left(\frac{9}{14} \right) + \left(\frac{5}{14} \right) \log_2 \left(\frac{5}{14} \right) \right] = 0,94$$

Bu hesaplama sonra, algoritma gereği, “aktivite” sınıf niteliği dışındaki niteliklerin her biri için entropi ve ardından kazanç ölçütleri bulunacak, bulunan

kazanç ölçütleri karşılaştırılıp en büyüğü belirlenecektir. En büyük kazanç ölçütüne sahip olan nitelik, kök düğüm yani dallanmanın başlayacağı ilk düğüm olarak seçilecektir. Kök düğümün belirlenmesinden sonra tüm niteliklerin diğer niteliklerle olan ilişkileri ortaya çıkana kadar aynı aşamalar tekrarlanacaktır. Algoritma uygulamasında, konunun anlatımında kullanılan gösterimin aynısı kullanılacaktır.

“HAVA” niteliği işlemleri: ilk olarak söz konusu nitelik, aldığı değerlere göre üç alt gruplara bölünür.

Tablo 4: Hava ve Sınıf Niteliği

HAVA	AKTİVİTE
Bulutlu	Olumlu
Bulutlu	Olumlu
Bulutlu	Olumlu
Bulutlu	Olumlu
Yağmurlu	Olumlu
Yağmurlu	Olumlu
Yağmurlu	Olumsuz
Yağmurlu	Olumlu
Yağmurlu	Olumsuz
Güneşli	Olumsuz
Güneşli	Olumsuz
Güneşli	Olumsuz
Güneşli	Olumlu
Güneşli	Olumlu

Kaynak: Özkan, 2008: 64.

$$|HAVA_{Bulutlu}|=4$$

$$|HAVA_{Yağmurlu}|=5$$

$$|HAVA_{Güneşli}|=5$$

$$H(HAVA_{Bulutlu}) = - \left[\left(\frac{4}{4} \right) \log_2 \left(\frac{4}{4} \right) \right] = 0$$

$$H(HAVA_{Yağmurlu}) = - \left[\left(\frac{3}{5} \right) \log_2 \left(\frac{3}{5} \right) + \left(\frac{2}{5} \right) \log_2 \left(\frac{2}{5} \right) \right] = 0,97$$

$$H(HAVA_{Güneşli}) = - \left[\left(\frac{2}{5} \right) \log_2 \left(\frac{2}{5} \right) + \left(\frac{3}{5} \right) \log_2 \left(\frac{3}{5} \right) \right] = 0,97$$

“HAVA” niteliği, aldığı değerlere göre alt gruplara ayrıldıktan sonra gerekli hesaplamalar yapılmıştır. Hesaplamaların ardından, söz konusu nitelik için entropi şu şekilde bulunur;

$$H(HAVA, AKT) = \left(\frac{4}{14} \right) H(HAVA_{Bulutlu}) + \left(\frac{5}{14} \right) H(HAVA_{Yağmurlu}) + \left(\frac{5}{14} \right) H(HAVA_{Güneşli})$$

$$H(HAVA, AKT) = \left(\frac{4}{14} \right) (0) + \left(\frac{5}{14} \right) (0,97) + \left(\frac{5}{14} \right) (0,97) = 0,69$$

“HAVA” niteliği için entropi hesaplamasının yapılmasının sonrasında, bu niteliğe ait kazanç ölçütü aşağıdaki gibi belirlenir;

$$Kazanç(HAVA, AKT) = H(AKT) - H(HAVA, AKT) = 0,94 - 0,69 = 0,25$$

“ISI” niteliği işlemleri: bu nitelik de “HAVA” niteliği gibi, aldığı değerlere göre üç alt gruba bölünür.

Tablo 5: ISI ve Sınıf Niteliği

ISI	AKTİVİTE
Sıcak	Olumsuz
Sıcak	Olumsuz
Sıcak	Olumlu
Sıcak	Olumlu
Ilık	Olumlu
Ilık	Olumsuz
Ilık	Olumlu
Ilık	Olumlu
Ilık	Olumlu
Ilık	Olumsuz
Soğuk	Olumlu
Soğuk	Olumsuz
Soğuk	Olumlu
Soğuk	Olumlu

Kaynak: Özkan, 2008: 62.

$$|ISI_{Sıcak}|=4$$

$$|ISI_{Ilık}|=6$$

$$|ISI_{Soğuk}|=4$$

$$H(ISI_{Sıcak}) = - \left[\left(\frac{2}{4} \right) \log_2 \left(\frac{2}{4} \right) + \left(\frac{2}{4} \right) \log_2 \left(\frac{2}{4} \right) \right] = 1$$

$$H(ISI_{Ilık}) = - \left[\left(\frac{4}{6} \right) \log_2 \left(\frac{4}{6} \right) + \left(\frac{2}{6} \right) \log_2 \left(\frac{2}{6} \right) \right] = 0,92$$

$$H(ISI_{Soğuk}) = - \left[\left(\frac{3}{4} \right) \log_2 \left(\frac{3}{4} \right) + \left(\frac{1}{4} \right) \log_2 \left(\frac{1}{4} \right) \right] = 0,8$$

“ISI” niteliği, aldığı değerlere göre alt gruplara ayrıldıktan sonra gerekli hesaplamalar yapılmıştır. Hesaplamaların ardından, “ISI” için entropi şu şekilde bulunur;

$$H(ISI, AKT) = \left(\frac{4}{14}\right)(1) + \left(\frac{6}{14}\right)(0,92) + \left(\frac{4}{14}\right)(0,8) = 0,9$$

Gerekli entropi hesaplarının ardından, son olarak ‘‘ISI’’ niteliğinin sahip olduğu kazanç ölçütü hesaplanır.

$$\text{Kazanç}(ISI, AKT) = H(AKT) - H(ISI, AKT) = 0,94 - 0,91 = 0,03$$

‘‘NEM’’ niteliği işlemleri: bu nitelik aldığı değerler açısından ‘‘normal’’ ve ‘‘yüksek’’ şeklinde iki farklı alt küme olarak düzenlenir.

Tablo 6: Nem ve Sınıf Niteliği

NEM	AKTİVİTE
Normal	Olumlu
Normal	Olumsuz
Normal	Olumlu
Normal	Olumlu
Normal	Olumlu
Normal	Olumlu
Normal	Olumlu
Yüksek	Olumsuz
Yüksek	Olumsuz
Yüksek	Olumlu
Yüksek	Olumlu
Yüksek	Olumsuz
Yüksek	Olumlu
Yüksek	Olumsuz

Kaynak: Özkan, 2008: 65.

$$|NEM_{Normal}| = 7$$

$$|NEM_{Yüksek}| = 7$$

$$H(NEM_{Normal}) = - \left[\left(\frac{6}{7}\right) \log_2 \left(\frac{6}{7}\right) + \left(\frac{1}{7}\right) \log_2 \left(\frac{1}{7}\right) \right] = 0,6$$

$$H(NEM_{Yüksek}) = - \left[\left(\frac{3}{7}\right) \log_2 \left(\frac{3}{7}\right) + \left(\frac{4}{7}\right) \log_2 \left(\frac{4}{7}\right) \right] = 0,99$$

Bu hesaplamalardan elde edilen sonuçlardan yola çıkarak ‘‘NEM’’ niteliği için entropi, aşağıdaki hesaplanır;

$$H(NEM, AKT) = \left(\frac{7}{14}\right) H(NEM_{Normal}) + \left(\frac{7}{14}\right) H(NEM_{Yüksek})$$

$$H(NEM, AKT) = \left(\frac{7}{14}\right) (0,6) + \left(\frac{7}{14}\right) (0,99) = 0,79$$

‘‘NEM’’ niteliğine ait kazanç ölçütü, aşağıdaki gibi son bir hesaplama ile elde edilir;

$$Kazanç(NEM, AKT) = H(AKT) - H(NEM, AKT)$$

$$Kazanç(NEM, AKT) = 0,94 - 0,79 = 0,12$$

İlk dallanmanın hangi nitelikten başlayacağını tespit etmek için analiz edilecek son nitelik, ‘‘Rüzgâr’’ niteliğidir. Bu nitelik aldığı değerler itibariyle iki farklı alt sete ayrılır.

Tablo 7: Rüzgâr ve Sınıf Niteliği

RÜZGÂR	AKTİVİTE
Yok	Olumsuz
Yok	Olumlu
Yok	Olumlu
Yok	Olumlu
Yok	Olumsuz
Yok	Olumlu
Yok	Olumlu
Yok	Olumlu
Var	Olumsuz
Var	Olumsuz
Var	Olumlu
Var	Olumlu
Var	Olumlu
Var	Olumsuz

Kaynak: Özkan, 2008: 66.

$$|Rüzgar_{Yok}|=8$$

$$|Rüzgar_{Var}|=6$$

$$H(Rüzgar_{Yok}) = - \left[\left(\frac{6}{8} \right) \log_2 \left(\frac{6}{8} \right) + \left(\frac{2}{8} \right) \log_2 \left(\frac{2}{8} \right) \right] = 0,81$$

$$H(Rüzgar_{Var}) = - \left[\left(\frac{3}{6} \right) \log_2 \left(\frac{3}{6} \right) + \left(\frac{3}{6} \right) \log_2 \left(\frac{3}{6} \right) \right] = 1$$

Bu adımların ardından, söz konusu nitelik için entropi hesabı şu şekilde yapılır;

$$H(RÜZGAR, AKT) = \left(\frac{8}{14} \right) H(Rüzgar_{Yok}) + \left(\frac{6}{14} \right) H(Rüzgar_{Var})$$

$$H(RÜZGAR, AKT) = \left(\frac{8}{14} \right) (0,81) + \left(\frac{6}{14} \right) (1) = 0,89$$

Son olarak ‘‘RÜZGÂR’’ niteliğinin entropisi hesaplandıktan sonra, niteliğin kazanç ölçütü hesaplanır.

$$Kazanç(RÜZGAR, AKT) = H(AKT) - H(RÜZGAR, AKT)$$

$$Kazanç(RÜZGAR, AKT) = 0,94 - 0,89 = 0,05$$

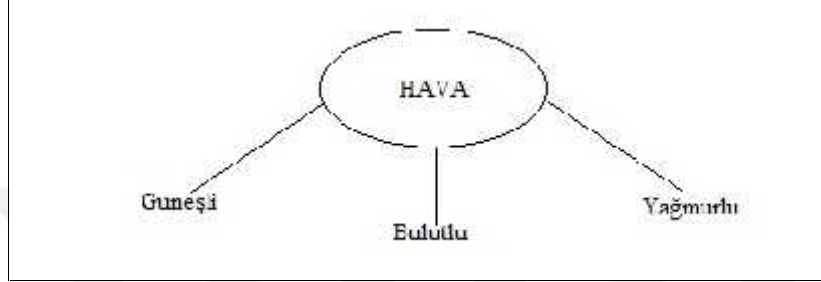
Tüm nitelikler için kazanç ölçütü hesaplanmıştır. Hesaplanan ölçütler karşılaştırılıp, aralarından en büyük olanı belirlenecektir. Böylece, ID3 algoritması kurallarına göre, karar ağacında dallanmanın başlayacağı ilk düğüm olan kök düğüm seçilir.

Tablo 8: Nitelikler ve Kazanç Ölçütleri

NİTELİKLER	KAZANÇ ÖLÇÜTÜ
HAVA	0,25
ISI	0,03
NEM	0,12
RÜZGÂR	0,05

Tablo 8’de gösterildiği gibi en büyük kazanç ölçütü “HAVA” niteliğine aittir. Bunun anlamı; karar ağacında dallanmanın başlayacağı niteliğin diğer bir deyişle kök düğümün, söz konusu nitelik olduğudur. Bundan dolayı “HAVA” niteliği, aldığı farklı değerlerce bölünmelere uğrar. Şekil 5’te açıklanan bölünme gösterilmiştir.

Şekil 5: Kök Düğümün Dallanması



Kaynak: Özkan, 2008: 68.

Kök düğümün dallanmasının ardından, “HAVA” niteliğine ait olan “Güneşli” değerinden başlayan dallanmalar tespit edilecektir. Bundan dolayı, eğitim kümesinde “HAVA” niteliği, “Güneşli” değerini alan kayıtlar kullanılacaktır. Bu kayıtlar üzerinden, aynı şekilde entropi ve diğer nitelikler için kazanç ölçütleri hesaplanacak sonrasında sıradaki dallanma gerçekleştirilecektir.

Tablo 9: “HAVA” niteliği “Güneşli” Değerini Alan Kayıtlar

ISI	NEM	RÜZGÂR	AKTİVİTE
Ilık	Normal	Var	Olumlu
Soğuk	Normal	Yok	Olumlu
Ilık	Yüksek	Yok	Olumsuz
Sıcak	Yüksek	Var	Olumsuz
Sıcak	Yüksek	Yok	Olumsuz

“ISI” niteliği işlemleri: Tablo 9’da görüldüğü gibi “ISI” niteliği, “Soğuk” değerini alan tek kayıt vardır. Bu kaydın sınıf niteliği değeri ise “Olumlu” değeridir. İlgili niteliği “Ilık” olan iki kayıttan birinin sınıf niteliği değeri, “Olumlu” diğerinin “Olumsuz” dur. Söz konusu nitelik değeri “Sıcak” olan iki kaydın da

sınıf niteliği değeri “Olumsuz” dur. Aşağıda “ISI” niteliği için entropi hesaplamaları yapılacak ve ardından kazanç ölçütü belirlenecektir.

İlk olarak, “HAVA” niteliğinin dallanması yapıldığı için bu niteliği içermeyen, yeni bir H(AKT) hesaplanmalıdır.

$$H(AKT) = - \left[\left(\frac{2}{5} \right) \log_2 \left(\frac{2}{5} \right) + \left(\frac{3}{5} \right) \log_2 \left(\frac{3}{5} \right) \right] = 0.970$$

$$H(ISI_{Soğuk}) = - \left[\left(\frac{1}{1} \right) \log_2 \left(\frac{1}{1} \right) \right] = 0$$

$$H(ISI_{Ilık}) = - \left[\left(\frac{1}{2} \right) \log_2 \left(\frac{1}{2} \right) + \left(\frac{1}{2} \right) \log_2 \left(\frac{1}{2} \right) \right] = 1$$

$$H(ISI_{Sıcak}) = - \left[\left(\frac{2}{2} \right) \log_2 \left(\frac{2}{2} \right) \right] = 0$$

$$H(ISI, OYUN) = \left(\frac{1}{5} \right) (ISI_{Soğuk}) + \left(\frac{2}{5} \right) (ISI_{Ilık}) + \left(\frac{2}{5} \right) (ISI_{Sıcak})$$

$$H(ISI, OYUN) = \left(\frac{1}{5} \right) (0) + \left(\frac{2}{5} \right) (1) + \left(\frac{2}{5} \right) (0) = 0,4$$

“ISI” niteliği için kazanç ölçütü;

$$\text{Kazanç}(ISI, AKT) = H(AKT) - H(ISI, AKT)$$

$$\text{Kazanç}(ISI, AKT) = 0,970 - 0,4 = 0,570$$

“NEM” niteliği için hesaplamalar: bu nitelikte kayıtların ikisi “Normal”, üç tanesi “Yüksek” değerini almıştır. “NEM” niteliği, “Normal” olan kayıtların, sınıf niteliği değerleri “Olumlu”, “Yüksek” olan kayıtlar ise “Olumsuz” değerini almıştır. Açıklanan bu ifadeler tablo 9’da gösterilmiştir.

$$H(NEM_{Normal}) = - \left[\left(\frac{2}{2} \right) \log_2 \left(\frac{2}{2} \right) \right] = 0$$

$$H(NEM_{Yüksek}) = - \left[\left(\frac{3}{3} \right) \log_2 \left(\frac{3}{3} \right) \right] = 0$$

$$H(NEM, AKT) = \left(\frac{2}{5}\right) (NEM_{Normal}) + \left(\frac{3}{5}\right) (NEM_{Yüksek})$$

$$H(NEM, AKT) = \left(\frac{2}{5}\right) (0) + \left(\frac{3}{5}\right) (0) = 0$$

$$Kazanç(NEM, AKT) = H(AKT) - H(NEM, AKT)$$

$$Kazanç(NEM, AKT) = 0,970 - 0 = 0,970$$

“RÜZGÂR” niteliği hesaplamaları: Tablo 9’da görüldüğü gibi, ilgili nitelik için kayıtların iki tanesi “Var”, kalan üç tanesi “Yok” değerini almıştır. Söz konusu nitelik için “Var” değerini alan kayıtların sınıf nitelik değerlerinden biri “Olumlu” iken, diğerinin değeri “Olumsuz” dur. Öte yandan “Yok” değerine sahip niteliklerin, bir tanesinin sınıf nitelik değeri “Olumlu”, kalan ikisinin sınıf nitelik değerleri ise “Olumsuz” dur. Bu verilere ilişki entropi ve kazanç ölçütü hesaplamaları şu şekilde yapılır;

$$H(RÜZGAR_{Yok}) = - \left[\left(\frac{1}{3}\right) \log_2 \left(\frac{1}{3}\right) + \left(\frac{2}{3}\right) \log_2 \left(\frac{2}{3}\right) \right] = 0,92$$

$$H(RÜZGAR_{Var}) = - \left[\left(\frac{1}{2}\right) \log_2 \left(\frac{1}{2}\right) + \left(\frac{1}{2}\right) \log_2 \left(\frac{1}{2}\right) \right] = 1$$

$$H(RÜZGAR, AKT) = \left(\frac{2}{5}\right) (RÜZGAR_{Var}) + \left(\frac{3}{5}\right) (RÜZGAR_{Yok})$$

$$H(RÜZGAR, AKT) = \left(\frac{2}{5}\right) (1) + \left(\frac{3}{5}\right) (0,92) = 0,95$$

$$Kazanç(RÜZGAR, OYUN) = H(OYUN) - H(RÜZGAR, OYUN)$$

$$Kazanç(RÜZGAR, OYUN) = 0,970 - 0,95 = 0,02$$

İlk dallanmadan sonra, “Güneşli” dalına bağlanacak düğümün seçimi için gerekli kazanç ölçütleri hesaplanmıştır.

Tablo 10: Nitelikler ve Kazanç Ölçütleri

Nitelik	ISI	NEM	RÜZGÂR
Kazanç Ölçütü	0,57	0,97	0,02

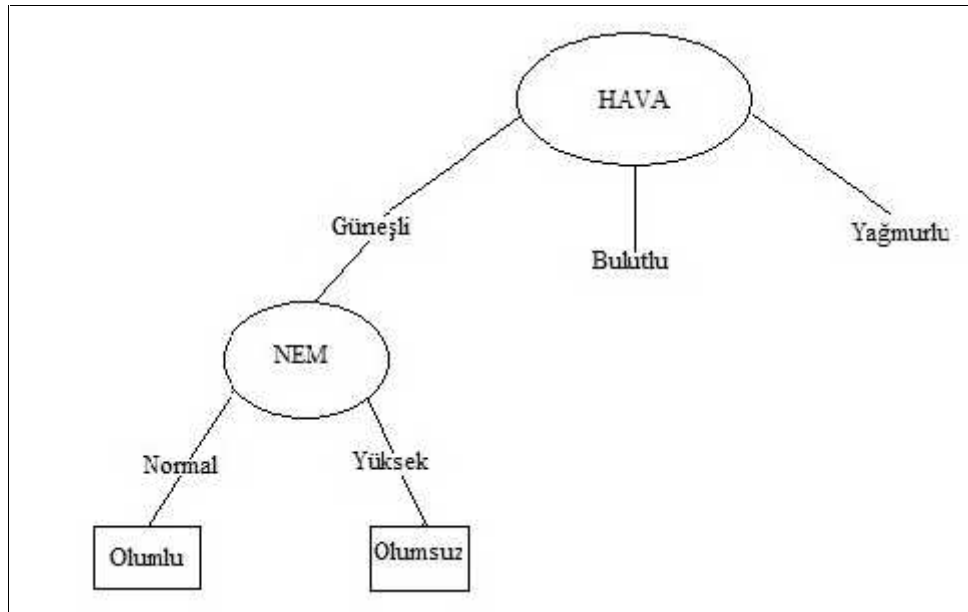
Tablo 10'daki kazanç ölçütlerine bakarak, sıradaki düğüm olarak “NEM” niteliğinin seçilmesi gerektiği anlaşılmaktadır.

“NEM” düğümü, niteliğin aldığı değerler itibarıyla “Yüksek” ve “Normal” dallarına ayrılacaktır. Söz konusu nitelikte “Yüksek” değerine sahip kayıtların, sınıf nitelik değerlerinin tamamı “Olumsuz”, “Normal” değerine sahip kayıtların ise tamamı “Olumlu” dur. Bu durumdan dolayı “NEM” düğümünün “Yüksek” ve “Normal” dallarının bağlandığı “Olumlu” ve “Olumsuz” değerleri, karar ağacının yapraklarından ikisini oluştururlar.

Bir karar ağacında, dalların yapraklara ulaşmasının anlamı; artık karar ağacının dallanmasının, yapraklara ulaşan dallara sahip nitelik dışındaki nitelikler için ağacın diğer yönlerine doğru olacağıdır. Bunun sebebi, daha önceki başlıklarda açıklandığı gibi, yaprakların herhangi bir karar ağacı akışındaki nihai eleman olmasıdır.

“NEM” düğümü için yapılan dallanma Şekil 6’da gösterilmiştir.

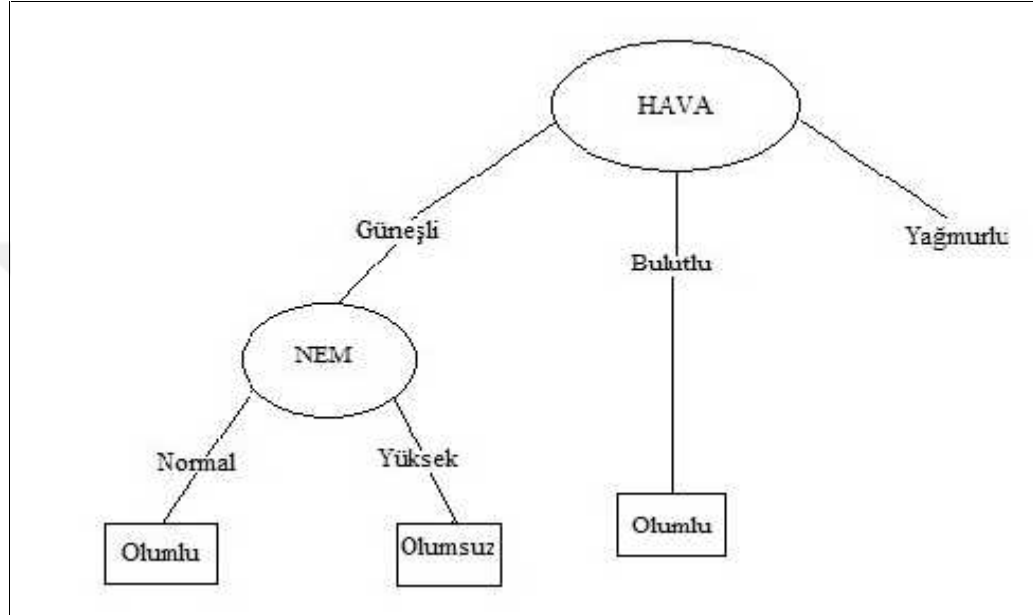
Şekil 6: “NEM” Düğümünün Dallanması



Kaynak: Özkan, 2008: 71.

Karar ağacındaki sıradaki dallanmalar, diğer dallar üzerinden devam edecektir. Sıradaki dal ‘‘Bulutlu’’ dalıdır. ‘‘HAVA’’ niteliği ‘‘Bulutlu’’ değerini alan kayıtların tüm sınıf nitelik değeri ‘‘Olumlu’’ dur. Böylece bu dala ilişkin dallanma sona ermiştir.

Şekil 7: ‘‘Bulutlu’’ Dalı ve Yaprakı



Kaynak: Özkan, 2008: 72.

‘‘Hava’’ niteliği , ‘‘Yağmurlu’’ olan kayıtlar için oluşturulacak dallanmalarla birlikte karar ağacı tamamlanacaktır. Bunun için niteliklerle ilgili entropi hesapları yapılmalıdır.

Tablo 11: ‘‘HAVA’’ Niteliği ‘‘Yağmurlu’’ Olan Kayıtlar

HAVA	ISI	NEM	RÜZGÂR	AKTİVİTE
Yağmurlu	Ilık	Yüksek	Yok	Olumlu
Yağmurlu	Soğuk	Normal	Yok	Olumlu
Yağmurlu	Soğuk	Normal	Var	Olumsuz
Yağmurlu	Ilık	Normal	Yok	Olumlu
Yağmurlu	Ilık	Yüksek	Var	Olumsuz

Kaynak: Özkan, 2008: 73.

$$H(AKT) = - \left[\left(\frac{3}{5} \right) \log_2 \left(\frac{3}{5} \right) + \left(\frac{2}{5} \right) \log_2 \left(\frac{2}{5} \right) \right] = 0,97$$

“ISI” niteliği işlemleri: bu niteliğe ait kayıtlar, aldıkları değerler bakımından 2 adet “Soğuk” ve 3 adet “Ilık” olarak iki farklı gruba ayrılmaktadır. “Soğuk” değerli kayıtların, sınıf nitelik değerlerinden biri “Olumlu”, biri “Olumsuz” iken “Ilık” değerli kayıtların sınıf nitelik değerlerinden iki tanesi “Olumlu”, bir tanesi “Olumsuz” dur. Bu iki değer için entropi ve ilgili nitelik için kazanç ölçütü aşağıdaki gibi hesaplanır.

$$H(ISI_{Soğuk}) = - \left[\left(\frac{1}{2} \right) \log_2 \left(\frac{1}{2} \right) + \left(\frac{1}{2} \right) \log_2 \left(\frac{1}{2} \right) \right] = 1$$

$$H(ISI_{Ilık}) = - \left[\left(\frac{2}{3} \right) \log_2 \left(\frac{2}{3} \right) + \left(\frac{1}{3} \right) \log_2 \left(\frac{1}{3} \right) \right] = 0,92$$

$$H(ISI, AKT) = \left(\frac{2}{5} \right) (ISI_{Soğuk}) + \left(\frac{3}{5} \right) (ISI_{Ilık})$$

$$H(ISI, AKT) = \left(\frac{2}{5} \right) (1) + \left(\frac{3}{5} \right) (0,92) = 0,95$$

$$Kazanç(ISI, AKT) = H(AKT) - H(ISI, AKT)$$

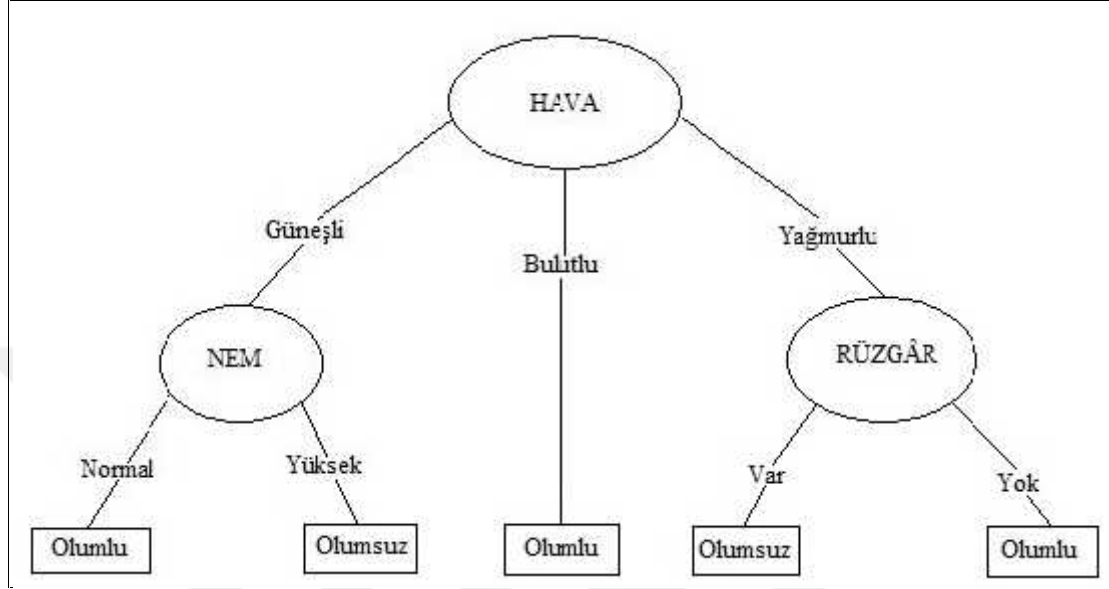
$$Kazanç(ISI, AKT) = 0,97 - 0,95 = 0,02$$

“RÜZGÂR” niteliği işlemleri: Tablo 11’de görüldüğü gibi bu niteliğe ilişkin kayıtların aldığı değerler, “Yok” ve “Var” değerleridir. “Yok” değerli kayıtların sınıf niteliği değerleri “Olumlu”, “Var” değerli kayıtların sınıf niteliği karşılığı ise “Olumsuz” dur. Bu durumda dallanmanın yapılacağı düğümün “RÜZGÂR” niteliği olduğuna karar verilir ve bu nitelik dışındaki başka bir nitelikten dallanma sağlanamayacağı anlamı çıkarılır. Böylece, son dallanma yapılır ve bu veri setine dayalı karar ağacı tamamlanır.

Oluşturulan karar ağacı, “AKTİVİTE” sınıf nitelik değeri araştırılan yeni bir kayıt için kullanılacak olan karar kurallarının bir görselidir. Bu sayede, aynı niteliklere sahip gelecekteki bir kaydın sınıf nitelik değeri, genelleştirilmiş olan ve

karar ağacı ile elde edilmiş karar kuralları ile kolayca tahmin edilebilir. Bu veriler için oluşturulmuş karar ağacı sıradaki şekildeki gibi elde edilir.

Şekil 8: “AKTİVİTE” Sınıf Niteliği İçin Karar Ağacı



Kaynak: Özkan, 2008: 75.

2.1.2. C4.5 Algoritması

C4.5 algoritması, önceki başlıkta anlatılmış olan ID3 algoritmasının gelişmiş bir halidir. Karar ağaçları oluşturarak sınıflandırma yapan bu algoritma, ID3 algoritması gibi entropi tabanlı bir algoritmadır. ID3 algoritmasından farklı olan en önemli özelliği, sayısal değerli nitelikler için de karar ağacı oluşturabilmesi ve sınıflandırma yapabilmesidir. Böylece, söz konusu algoritma, sayısal değer alan niteliklerle çalışıldığında ID3 algoritmasında ayrılmakla birlikte hesaplamalar ve karar ağacı oluşturma mantığı aynıdır.

Nitelik değerlerinin sayısal olması karar ağacı oluştururken bir zorluk oluşturur. C4.5 algoritması bu zorluğu, sayısal nitelik değerlerini sayısal olmayan değerlere dönüştürerek ortadan kaldırmaktadır. Böylelikle hem kategorik hem de sayısal verilerle çalışabilen bir algoritmadır.

C4.5 algoritmasıyla, sayısal nitelik değerlerini sayısal olmayan değerlere dönüştürmek için bir eşik değer belirlenir. Eşik değer rastgele belirlenebilir fakat en uygun olanı sayısal nitelik değerlerinin orta noktası olan değeri eşik değer olarak belirlemektir. Eşik değer belirlendikten sonra ilgili nitelik değerleri, eşik değerle karşılaştırılır ve ardından nitelik değerleri eşik değere göre “eşit veya küçük” ve “büyük” olarak kategorize edilerek sayısal olmayan değer haline getirilir.

Tablo 12: Sayısal Nitelik Değerleri

Nitelik 1	Nitelik 2	Sınıf
a	70	Sınıf1
a	90	Sınıf2
a	85	Sınıf2
b	78	Sınıf1
b	65	Sınıf2

Kaynak: Özkan, 2008: 78.

Eşik değeri belirlemek için değerler her biri bir defa kullanılmak üzere sıralanır ve orta noktasında kalan değer eşik değer olur. {65, 70, 78, 85, 90} sıralamasından dolayı 78 eşik değer olarak belirlenir. Eğer altı adet değer olsaydı ortada kalan iki değerın ortalaması eşik değer olarak belirlenirdi.

Eşik değerin belirlenmesinin ardından, sayısal nitelik değerleri bu değere göre düzenlenir. Örneğin; 70 değeri eşik değerden küçük olduğu için artık nitelik değeri olarak eşit veya küçük değerini alacaktır.

Tablo 13: Eşik Değer ve Nitelik Değerleri

Nitelik 1	Nitelik 2	Sınıf
a	eşit veya küçük	Sınıf1
a	büyük	Sınıf2
a	büyük	Sınıf1
b	eşit veya küçük	Sınıf1
b	eşit veya küçük	Sınıf2

Kaynak: Özkan, 2008: 79.

Düzenlemenin ardından ‘‘Nitelik 2’’ için entropi hesaplamaları, daha önceki bölümde anlatılmış olan ID3 algoritmasındaki yapıldığı şekilde yapılacaktır. Sınıf niteliği için;

$$H(\text{SINIF}) = - \left[\left(\frac{3}{5} \right) \log_2 \left(\frac{3}{5} \right) + \left(\frac{2}{5} \right) \log_2 \left(\frac{2}{5} \right) \right] = 0,97$$

$$|\text{NİTELİK2}_{\leq 78}| = 3$$

$$|\text{NİTELİK2}_{>78}| = 2$$

$$H(\text{NİTELİK2}_{\leq 78}) = - \left[\left(\frac{2}{3} \right) \log_2 \left(\frac{2}{3} \right) + \left(\frac{1}{3} \right) \log_2 \left(\frac{1}{3} \right) \right] = 0,92$$

$$H(\text{NİTELİK2}_{>78}) = - \left[\left(\frac{1}{2} \right) \log_2 \left(\frac{1}{2} \right) + \left(\frac{1}{2} \right) \log_2 \left(\frac{1}{2} \right) \right] = 1$$

$$H(\text{NİTELİK2}, \text{SINIF}) = \frac{3}{5} (0,92) + \frac{2}{5} (1) = 0,95$$

$$\text{KAZANÇ}(\text{NİTELİK2}, \text{SINIF}) = H(\text{SINIF}) - H(\text{NİTELİK2}, \text{SINIF})$$

$$\text{KAZANÇ}(\text{NİTELİK2}, \text{SINIF}) = 0,97 - 0,95 = 0,02$$

Tablo 13’teki verilen ve sayısal değerlere sahip nitelikleri olan örnek veri setinde, C4.5 algoritması kullanılmış ve sayısal nitelik değerleri eşik değer baz alınarak sayısal olmayan değerlere dönüştürülmüştür. Böylece, entropi ve kazanç hesaplamaları, kategorik nitelik değerleri ile çalışan ID3 algoritmasında olduğu gibi basit bir şekilde hesaplanabilir duruma getirilmiş ve bu hesaplamalar ışığında bir karar ağacı oluşturup sınıflandırma yapmak mümkün hale gelmiştir.

2.1.2.1. C4.5 Algoritması ve Bilinmeyen Nitelik Değerleri

Oluşturulan veritabanındaki bazı veriler, verinin kaybolması, verinin kaydedilmemesi gibi çeşitli sebeplerle eksik olabilirler. Bu eksiklik, beraberinde birtakım sorunlar yaratabilir. Bu sorunlardan kaçınmak ve sorunu ortadan kaldırmak için yapılacak işlemlerden biri, eksik veriye ilişkin değerlerin tümünün

veritabanından çıkarılmasıdır. Bu çözüm, sorunu basitçe ortadan kaldırabilir fakat kayıp ya da eksik veri sayısının çok fazla olduğu durumlarda sağlıklı bir çözüm olmayacaktır.

Bu durumda yapılacak işlem, eksik verilerle de çalışabilen yeni bir algoritmik yapı elde etmektir. C4.5 algoritması anlatıldığı gibi eksik verilen olduğu bir ortamda da karar ağacı oluşturabilen ve sınıflandırma yapabilen bir algoritmadır. ID3 algoritmasından önemli bir diğer farkı da bu özelliğidir. C4.5 algoritması, eksik verilerin olduğu durumda kazanç ölçütü değerini hesaplarken bir F düzeltme faktöründen faydalanır. F düzeltme faktörü; veritabanındaki değeri bilinen örnek sayısının, toplam örnek sayısına oranıdır. Böylece kazanç ölçütü şu şekilde hesaplanır;

$$\text{Kazanç}(X)=F[H(T) - H(X,T)]$$

Tablo 14: Eksik Veri

Nitelik	Sınıf
a	Sınıf1
a	Sınıf2
a	Sınıf2
a	Sınıf2
a	Sınıf1
?	Sınıf1
b	Sınıf1
b	Sınıf1
b	Sınıf1
c	Sınıf2
c	Sınıf2
c	Sınıf1
c	Sınıf1
c	Sınıf1

Kaynak: Özkan, 2008: 82.

$$H(\text{SINIF}) = - \left[\left(\frac{8}{13} \right) \log_2 \left(\frac{8}{13} \right) + \left(\frac{5}{13} \right) \log_2 \left(\frac{5}{13} \right) \right] = 0,96$$

$$H(\text{NİTELİK, SINIF}) = \frac{5}{13} \left[- \left(\frac{2}{5} \right) \log_2 \left(\frac{2}{5} \right) - \left(\frac{3}{5} \right) \log_2 \left(\frac{3}{5} \right) \right] + \frac{3}{13} \left[- \left(\frac{3}{3} \right) \log_2 \left(\frac{3}{3} \right) - \left(\frac{0}{3} \right) \log_2 \left(\frac{0}{3} \right) \right] \\ + \frac{5}{13} \left[- \left(\frac{3}{5} \right) \log_2 \left(\frac{3}{5} \right) - \left(\frac{2}{5} \right) \log_2 \left(\frac{2}{5} \right) \right] = 0,75$$

$$\text{Kazanç}(\text{NİTELİK, SINIF}) = \frac{13}{14} (0,96 - 0,75) = 0,20$$

C4.5 algoritmasının ID3 algoritmasından farklı olan özelliklerinden dolayı hesaplamalara yansıyan değişiklikler gösterildiği gibidir. Bunun dışındaki temel mantık ve işleyiş ID3 algoritması ile aynıdır. ID3 algoritmasında olduğu gibi bu algoritmada da kazanç ölçütüne göre dallanmalar yapılarak, karar ağacı ve sonucunda karar kuralları oluşturulacaktır. Karar kurallarının elde edilmesinin ardından sınıflandırma yapılabilecektir.

2.2. SINIFLANDIRMA VE REGRESYON AĞAÇLARI – GİNİ ALGORİTMASI

Veri madenciliği araştırmalarında sınıflandırma yapmak için başvuru yapılarından birisi de sınıflandırma ve regresyon ağaçları yapısıdır. Bu yaklaşımda her düğüm iki dala ayrılmaktadır (Emel ve Taşkın, 2015: 230). İkili dallanmalarla oluşturulacak olan karar ağacının hedefi, aynı veya benzer değerlere sahip olan alt gruplar elde etmektir (Oğuzlar, 2004: 83). Düğümlerin ikili dallanmaları, düğümlere farklı ölçütler uygulanarak yapılabilmektedir. Bu ölçütlerden biri gini indeksidir. Söz konusu indeks, gini algoritması olarak bu başlık altında anlatılacaktır.

Gini algoritmasında, her bir nitelik, değerleri iki grup olacak şekilde ayrılır. Gruplar “sağ” ve “sol” şeklinde isimlendirilir. Niteliklerin sağ ve sol grupları için, aşağıdaki formülle $Gini_{SAĞ}$ ve $Gini_{SOL}$ değerleri hesaplanır. Niteliklerin alt grupları için Gini değerlerinin hesaplanmasının ardından, her bir nitelik için ayrı bir Gini değeri diğer bir deyişle katsayısı hesaplanır. $Gini_{SAĞ}$, $Gini_{SOL}$ ve her bir j niteliği için $Gini_j$ katsayıları şu formüllerle hesaplanır:

k; sınıfların sayısı, T; bir düğümdeki örnekler, $|T_{SAĞ}|$; sağ taraftaki örneklerin sayısı, $|T_{SOL}|$; sol taraftaki örneklerin sayısı, L_i ; sol tarafta i kategorisindeki örneklerin sayısı, R_i ; sağ taraftaki i kategorisindeki örneklerin sayısı, n kayıt sayısı olmak üzere;

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

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

$$Gini_j = \frac{1}{n} (|T_{SOL}| \times Gini_{SOL} + |T_{SAĞ}| \times Gini_{SAĞ})$$

Her bir nitelik için elde edilen Gini katsayıları arasından en küçüğüne sahip olan nitelik, ikili dallanma yapılacak düğüm olarak seçilir. Ağaç oluşumu tamamlanana kadar bu aşamalar tekrarlanır (Özkan, 2008: 106). Aşağıdaki eğitim verisi grubuna Gini algoritması uygulanacaktır.

Tablo 15: Eğitim Verisi

Başvuru	Eğitim	Yaş	Cinsiyet	Kabul
1	Orta	Yaşlı	Erkek	Evet
2	İlk	Genç	Erkek	Hayır
3	Yüksek	Orta	Kadın	Hayır
4	Orta	Orta	Erkek	Evet
5	İlk	Orta	Erkek	Evet
6	Yüksek	Yaşlı	Kadın	Evet
7	İlk	Genç	Kadın	Hayır

Kaynak: Özkan, 2008: 108.

İlk olarak tüm nitelikler algoritmanın çalışma prensibi gereği, iki gruba ayrılır. Sınıf nitelik, “Kabul” niteliğidir. Bu nitelik, “Evet” ve “Hayır” olarak iki farklı değer almıştır. Sınıf niteliğin aldığı değerlere göre eğitim verisindeki nitelikler sağ ve sol olarak ikili gruplara ayrılacaktır. Gruplama işleminin ardından her bir nitelik için Gini katsayısı hesaplanır.

Niteliklerin aldığı değerler, sınıf nitelik değerine göre düzenlendiklerinde sıradaki tablodaki gibi bir grupta elde edilir.

Tablo 16: Gruplara Ayrılmış Eğitim Verisi

Kabul	Eğitim		Yaş		Cinsiyet	
	İlk	Orta Yüksek	Genç	Orta Yaşlı	Kadın	Erkek
Evet	1	3	0	4	1	3
Hayır	2	1	2	1	2	1

Kaynak: Özkan, 2008: 108.

Tablo 16’da her nitelik sağ ve sol olarak ikiye ayrılmıştır. Örneğin; “Eğitim” niteliği “İlk” ve sınıf nitelik değeri “Evet” olan kayıt sayısı 1’dir ve “Eğitim” niteliği “İlk” değerini alan kayıtlar sol, “Orta ve Yüksek” değerini alan kayıtlar sağ grubu oluşturmaktadır. Nitelikler için Gini katsayıları şu şekilde hesaplanır;

“Eğitim” niteliği:

$$Gini_{SOL} = 1 - \left[\left(\frac{1}{3} \right)^2 + \left(\frac{2}{3} \right)^2 \right] = 0,44$$

$$Gini_{SAĞ} = 1 - \left[\left(\frac{3}{4} \right)^2 + \left(\frac{1}{4} \right)^2 \right] = 0,38$$

$$Gini_{EĞİTİM} = \frac{3 \times (0,44) + 4 \times (0,38)}{7} = 0,41$$

“Yaş” niteliği:

$$Gini_{SOL} = 1 - \left[\left(\frac{0}{2} \right)^2 + \left(\frac{2}{2} \right)^2 \right] = 0$$

$$Gini_{SAĞ} = 1 - \left[\left(\frac{4}{5} \right)^2 + \left(\frac{1}{5} \right)^2 \right] = 0,32$$

$$Gini_{YAŞ} = \frac{2 \times (0) + 5 \times (0,32)}{7} = 0,23$$

“Cinsiyet” niteliği:

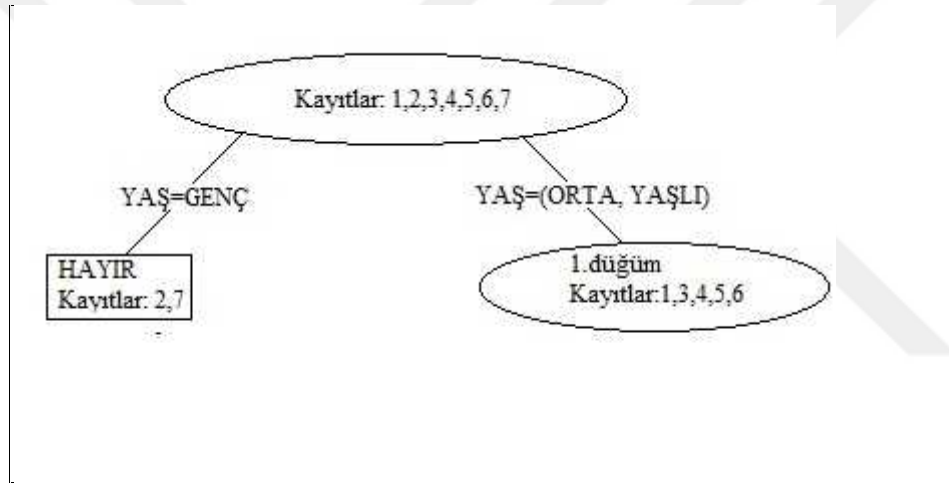
$$Gini_{SOL} = 1 - \left[\left(\frac{1}{3} \right)^2 + \left(\frac{2}{3} \right)^2 \right] = 0,44$$

$$Gini_{SAĞ} = 1 - \left[\left(\frac{3}{4} \right)^2 + \left(\frac{1}{4} \right)^2 \right] = 0,38$$

$$Gini_{CİNSİYET} = \frac{3 \times (0,44) + 4 \times (0,38)}{7} = 0,41$$

En küçük Gini değeri “Yaş” niteliğine aittir. Böylece ikili dallanmanın yapılacağı nitelik seçimi yapılmıştır. Dallanma “Genç” ve “Orta, Yaşlı” olarak yapılacaktır. İlgili nitelik değeri “genç” olanlar 2. ve 7. kayıtlardır. Bu kayıtlar dışındaki kayıtlar ise sağ dalda yer alacaktır.

Şekil 9: İlk Dallanma



Kaynak: Özkan, 2008: 110.

“YAŞ= GENÇ” değerini alan 2. ve 7.kayıtlar sol dalı oluştururlar. Söz konusu kayıtlar “Hayır” sınıf niteliğine sahiptir. Dolayısıyla bu kayıtlar dışındaki kayıtlar ile işleme devam edilir.

Tablo 17: İkinci Dallanma İçin Eğitim Verisi

Başvuru	Eğitim	Yaş	Cinsiyet	Kabul
1	Orta	Yaşlı	Erkek	Evet
3	Yüksek	Orta	Kadın	Hayır
4	Orta	Orta	Erkek	Evet
5	İlk	Orta	Erkek	Evet
6	Yüksek	Yaşlı	Kadın	Evet

Kaynak: Özkan, 2008: 110.

Kök düğüm için yapıldığı gibi, tablo 17’deki veriler gruplara ayrılarak niteliklerin aldığı değerlerin frekansları düzenlenir. Bu düzenleme yapıldığında, tablo 18’deki yapı elde edilir.

Tablo 18: Gruplara Ayrılmış İkinci Düğüm Eğitim Verisi

Kabul	Eğitim		Yaş		Cinsiyet	
	İlk	Orta Yüksek	Orta	Yaşlı	Kadın	Erkek
Evet	1	3	2	2	1	3
Hayır	0	1	1	0	1	0

Kaynak: Özkan, 2008: 111.

Tablo 18’deki frekanslardan yola çıkarak her nitelik için yeniden Gini katsayıları elde edilir.

“Eğitim” niteliği:

$$Gini_{SOL} = 1 - \left[\left(\frac{1}{1} \right)^2 + \left(\frac{0}{1} \right)^2 \right] = 0$$

$$Gini_{SAG} = 1 - \left[\left(\frac{3}{4} \right)^2 + \left(\frac{1}{4} \right)^2 \right] = 0,38$$

$$Gini_{EĞİTİM} = \frac{1 \times (0) + 4 \times (0,38)}{5} = 0,30$$

“Yaş” niteliği:

$$Gini_{SOL} = 1 - \left[\left(\frac{2}{3} \right)^2 + \left(\frac{1}{3} \right)^2 \right] = 0,44$$

$$\text{Gini}_{\text{SAĞ}} = 1 - \left[\left(\frac{2}{2} \right)^2 + \left(\frac{0}{2} \right)^2 \right] = 0$$

$$\text{Gini}_{\text{YAŞ}} = \frac{3 \times (0,44) + 2 \times (0)}{5} = 0,27$$

“Cinsiyet” niteliği:

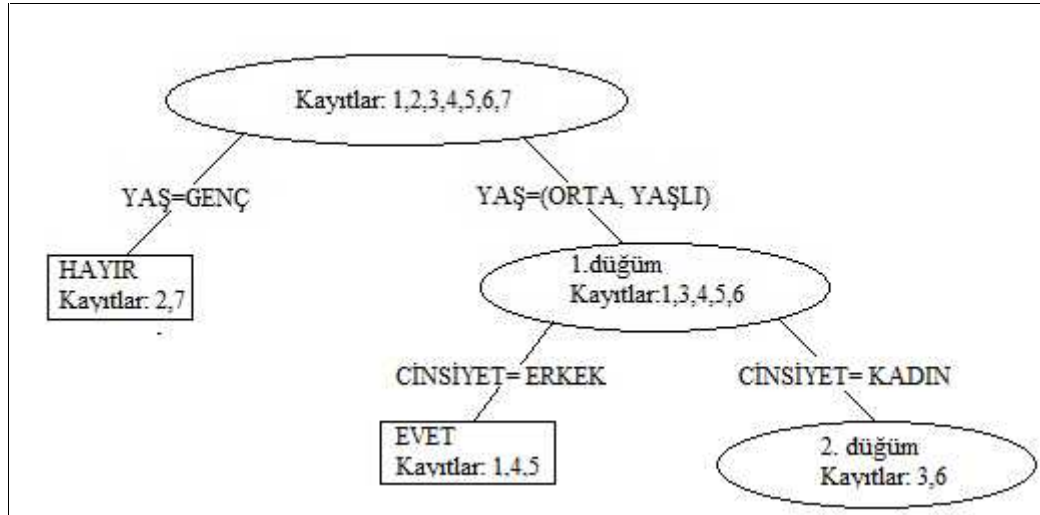
$$\text{Gini}_{\text{SOL}} = 1 - \left[\left(\frac{1}{2} \right)^2 + \left(\frac{1}{2} \right)^2 \right] = 0,50$$

$$\text{Gini}_{\text{SAĞ}} = 1 - \left[\left(\frac{3}{3} \right)^2 + \left(\frac{0}{3} \right)^2 \right] = 0$$

$$\text{Gini}_{\text{CİNSİYET}} = \frac{2 \times (0,50) + 3 \times (0)}{5} = 0,20$$

Niteliklerin Gini değerleri kıyaslandığında, en küçük değere sahip olan niteliğin “Cinsiyet” olduğu tespit edilir. İlgili nitelik “Erkek” ve “Kadın” olarak dallanacaktır. “Cinsiyet=Erkek” değerini almış kayıtların sınıf nitelik değerleri “Evet” değeridir. Bu kayıtlar 1. , 4. ve 5. kayıtlar olup, sol dalı oluşturacaklardır.

Şekil 10: İkinci Dallanma

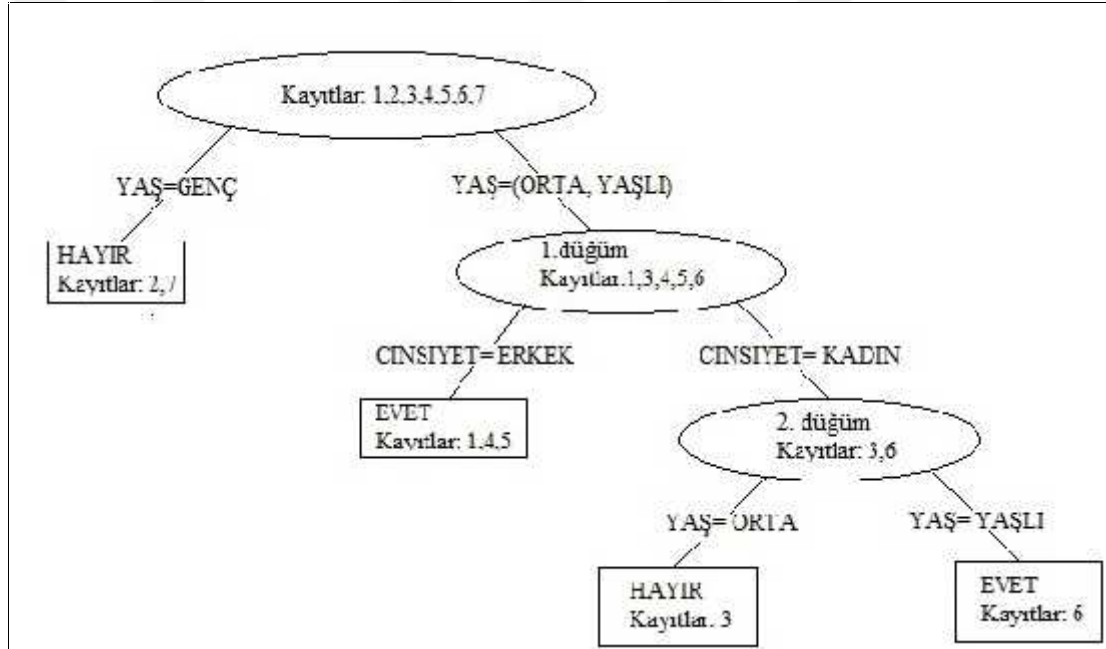


Kaynak: Özkan, 2008: 112.

İkinci dallanmanın ardından 1, 4 ve 5 numaraları kayıtlar dışarıda bırakılır. Böylece 3 ve 6 numaraları kayıtlar sıradaki dallanma için kullanılır. 3 ve 5 numaraları kayıtların ikisinin de, “Eğitim” niteliği “Yüksek”, “Cinsiyet” niteliği “Kadın” değerlerini almıştır. Söz konusu kayıtlar için “YAŞ=ORTA” ise sınıf nitelik değeri “Hayır”, “YAŞ=YAŞLI” ise sınıf nitelik değeri “Evet” olduğu görülmektedir.

Bu durumdan dolayı 2. düğüm için “Yaş” niteliğine göre son bir dallanma yapılabilir. Tüm işlemlerinin sonucunda tamamlanmış karar ağacı Şekil 11’de gösterilmiştir.

Şekil 11: Karar Ağacının Son Hali



Kaynak: Özkan, 2008: 113.

2.3. NAİVE BAYES SINIFLANDIRICISI

Bir olayın gerçekleşme bilgisi, başka bir olayın gerçekleşme olasılığını etkiliyorsa burada koşullu olasılıktan bahsedilir. İki olay arasında bu şekilde bir ilişkisinin bulunması, bir olayın meydana gelmesi diğer bir olayın meydana gelmesini engellemekle yani bağdaşan iki olay olmakla birlikte olayın tek başına meydana gelme olasılığı ile başka bir olayın gerçekleşme bilgisinin var olduğu

durumdaki meydana gelme olasılığının değişmesi, şarta bağlı bir olasılık hesabını gerektirmektedir. Bağıdaşan A ve B olayları ele alınırfa;

B olayının gerekleşme olasılığı A olayının gerekleşmesine bağıdır. Bu koşullu olasılık şu şekilde belirtilir:

$$P(B|A)=\frac{P(A\cap B)}{P(A)}$$

Bu denklemden faydalanılarak olayların kesişim olasılığı elde edilir.

$$P(A\cap B)=P(B|A).P(A)$$

Aynı mantıktan hareketle A olayına ilişkin olasılık ve sonucunda kesişim olasılığı aşığıdaki gibi elde edilir.

$$P(A|B)=\frac{P(A\cap B)}{P(B)}$$

$$P(A\cap B)=P(A|B).P(B)$$

Böylece B olayının koşullu olasılığı şu şekilde bulunur:

$$P(B|A)=\frac{P(A|B).P(B)}{P(A)}$$

Bayes teoremi, koşullu olasılıkların hesaplanmasında kullanılan, istatistikte ve olasılıkta büyük bir öneme sahip olan bir teoremdir. Bu teoremden faydalanılarak sınıflandırma işlemi yapılabilir. Teoriye göre koşullu olasılıklar aşığıdaki gibi hesaplanmaktadır.

$P(A)$: A olayının meydana gelme olasılığı

$P(B)$: B olayının meydana gelme olasılığı

$P(A|B)$: B olayından sonra A olayının meydana gelme olasılığı

$P(B|A)$: A olayından sonra B olayının meydana gelme olasılığı

A olayının önsel olasılığı, tüm ayrık parçalarının birleşimidir(Akkoç, 2012: 7).

$$P(A) = P(A \cap B) + P(A \cap \bar{B})$$

$$P(B|A) = \frac{P(A \cap B)}{P(A)}$$

$P(B|A)$ eşitliği, koşullu olasılık denklemini kullanarak açıldıktan sonra bayes formülü elde edilir. Elde edilen bayes formülü, iki ve daha fazla olay için aşağıdaki gibi genelleştirilebilir.

$$P(B|A) = \frac{P(A \cap B)}{P(A \cap B) + P(A \cap \bar{B})}$$

$$P(B|A) = \frac{P(A|B).P(B)}{P(A|B).P(B) + P(A|\bar{B}).P(\bar{B})}$$

$$P(B_i|A) = \frac{P(A|B_i).P(B_i)}{\sum_{j=1}^n P(A|B_j).P(B_j)}$$

İstatistiksel sınıflandırma araçlarından biri olan ve veri madenciliği uygulamalarında yaygın bir şekilde kullanılan naive bayes sınıflandırma algoritması, sınıf nitelik değeri araştırılan bir veri kümesinin, her bir sınıf niteliğine sahip olma olasılıklarının hesaplanıp, hesaplanan olasılıklardan en büyük değere sahip olan nitelik değerini, sınıf niteliği değeri olarak atayan sınıflandırma aracıdır. Naive bayes, veri madenciliği kapsamına giren ve literatürde metin madenciliği olarak adlandırılan analiz türünde de oldukça sık bir kullanım alanına sahiptir.

Bu algoritma, yeni bir veri ya da veri grubunun, mevcut sınıf değerlerinden herhangi birini alma olasılıklarını hesaplamak için elde hazır bir şekilde sınıflanmış olarak bulunan verilerden faydalanmaktadır(Dondurmacı, 2011: 28). Bu sınıflandırıcı ile yapılan tahminlerin teorik altyapısında bayes teoremi yer almaktadır. Veri kümelerinin içerdiği nitelik değerlerinin koşullu olarak birbirinden bağımsız olduğu

varsayımı, yapılacak sınıflandırma tahminlerinin analiz sürecini kolaylaştırmaktadır. Bu nedenle yaklaşım, “naive” ismini almıştır. Tekniğin teorik anlamda bayes teoremine dayanmasından dolayı sınıf nitelik değeri tahminlerinde, koşullu olasılık kavramından yararlanılmakta olup, bu olasılıklar aşağıdaki formülle hesaplanmaktadır.

$X=\{x_1, x_2, \dots, x_n\}$ nitelik değerlerinden oluşan bir veri kümesi,
 $C=\{C_1, C_2, \dots, C_m\}$ değerlerinden oluşan sınıf niteliği olmak üzere;

$$P(C_i|X) = \frac{P(X|C_i) \cdot P(C_i)}{P(X)}$$

$$\text{Sonrasal olasılık} = \frac{\text{Koşullu Olasılık(Likelihood) x Sınıf Elemanının Önsel Olasılığı}}{\text{Tahmincinin Önsel Olasılığı}}$$

Örneğe ait x_i değerlerinin birbirinden bağımsız olması varsayımdan dolayı ve işlem hacmini azaltmak üzere $P(C_i|X)$ yerine aşağıdaki eşitlikten elde edilecek olasılıklar kullanılabilir(Han ve Kamber, 2006: 312).

$$P(X|C_i) = \prod_{k=1}^n P(x_k|C_i)$$

Böylece, her bir sınıf değerine ait olasılıkların hesaplanmasında payda kısmı eşit olacağı için pay değerleri arasından en büyük değere olasılığa sahip nitelik değeri, veri kümesinin sınıf niteliği olarak atanır.

$$\arg \max_{C_i} = \{P(X|C_i)P(C_i)\}$$

Yukarıda açıklanmış olan yöntemle, en büyük sonrasal sınıflandırma (Maximum A Posteriori classification=MAP) yöntemi adı verilmektedir. Böylece naive bayes sınıflandırıcısı için aşağıdaki bağıntı kullanılabilir.

$$C_{MAP} = \arg \max_{C_i} \left[\left(\prod_{k=1}^n P(x_k|C_i) \right) P(C_i) \right]$$

2.3.1. Naive Bayes Sınıflandırıcısında Nitelik Değeri Türlerine Göre Olasılıkların Hesaplanması

Sınıf nitelik değeri araştırılan bir veri kümesinin sahip olduğu nitelikler, farklı türlerde değer almaktadır. Bazı nitelikler, kategorik olarak adlandırılan değerleri alırken, bazıları ise sayısal değerler almaktadır. Örneğin; araştırmada kullanılacak veri kümesinin sahip olduğu niteliklerden ikisi “cinsiyet” ve “yaş” nitelikleri olsun. Bu niteliklerden “cinsiyet” niteliği, aldığı değerler itibariyle kategorik bir nitelik iken “yaş” niteliği, sayısal bir niteliktir. Niteliklerin aldığı değerler açısından sahip olduğu bu farklılık, doğal olarak, niteliklerle ilgili olasılık hesaplamalarında da ortaya çıkmaktadır.

2.3.1.1. Kategorik Nitelik Değerleri

Naive bayes sınıflandırıcısı ile sınıflandırma sürecinde koşullu olasılık kavramından yararlanılıp, X nitelik kümesi, C_i ($i= 1, 2, \dots, n$) sınıf nitelik değerleri olmak üzere, elde edilen $[P(X|C_i)P(C_i)]$ olasılıkları arasından en büyük olasılığa sahip sınıf niteliğinin, sınıf değeri araştırılan veri grubunun sınıflandırma tahmini olarak atandığı ve bu yöntemle en büyük sonrasal sınıflandırma adı verildiği önceki başlıklarda açıklanmıştı.

Olasılıkların hesaplanmasında kullanılan niteliklerin aldığı değerler, kategorik değerler ise bahsedilen olasılıklar, nitelik değerlerinin frekansları ile hesaplanır.

Tablo 19: Kategorik Veri Kümesi

Başvuru	Yaş	Eğitim	Cinsiyet	Kabul
1	Yaşlı	Orta	Erkek	Evet
2	Genç	İlk	Erkek	Hayır
3	Orta	Yüksek	Kadın	Hayır
4	Orta	Orta	Erkek	Evet
5	Orta	İlk	Erkek	Evet
6	Yaşlı	Yüksek	Kadın	Evet
7	Genç	İlk	Kadın	Hayır
8	Orta	Orta	Kadın	Evet

Kaynak: Özkan, 2008: 171.

Tablo 19’da verilen eğitim verisinde Kabul niteliği, sınıf niteliğidir. Bu verilerle naive bayes sınıflandırıcısı kullanarak, X ; x_1 : Yaş=Orta, x_2 : Eğitim=Yüksek, x_3 : Cinsiyet=Kadın veri grubunun sınıf nitelik değerini bulmak için kullanılacak nitelik frekansları ve olasılıklar Tablo 20’de verilmiştir.

Tablo 20: Nitelik Frekansları ve Olasılıklar

Nitelikler	Nitelik Değeri	Kabul			
		Evet		Hayır	
		Frekans	Olasılık	Frekans	Olasılık
Yaş	Genç	0	0	2	2/3
	Yaşlı	2	2/5	0	0
	Orta	3	3/5	1	1/3
Eğitim	İlk	1	1/5	2	2/3
	Orta	3	3/5	0	0
	Yüksek	1	1/5	1	1/3
Cinsiyet	Kadın	2	2/5	2	2/3
	Erkek	3	3/5	1	1/3

Tablo 20’deki olasılıklardan, sınıf üyeliği araştırılan X veri grubu niteliklerinin aldığı değerler için, hem “Hayır” hem de “Evet” değerine göre, matematiksel olarak $[P(X|C_i)P(C_i)]$ şeklinde ifade edilen koşullu olasılıklar elde edilir. İlk olarak veri grubunun sınıf nitelik değeri açısından “Evet” sınıfına ait olma olasılığı hesaplanacaktır. C_1 : Kabul=Evet, C_2 : Kabul=Hayır olmak üzere;

$$P(X|C_1)P(C_1) = P(X|Kabul=Evet)P(Kabul=Evet);$$

$$P(x_1|C_1) = P(\text{Yaş=Orta}|Kabul=Evet) = \frac{3}{5}$$

$$P(x_2|C_1) = P(\text{Eğitim=Yüksek}|Kabul=Evet) = \frac{1}{5}$$

$$P(x_3|C_1) = P(\text{Cinsiyet=Kadın}|Kabul=Evet) = \frac{2}{5}$$

$$P(X|C_1)=P(X|Kabul=Evet)=\frac{3}{5} \times \frac{1}{5} \times \frac{2}{5}=\frac{6}{125}$$

$$P(C_1)=P(Kabul=Evet)=\frac{5}{8}$$

$$P(X|Kabul=Evet)P(Kabul=Evet)=\frac{6}{125} \times \frac{5}{8}=0,03$$

Aynı şekilde, X veri grubunun “Hayır” sınıfı üyeliğine ait olma olasılığı hesaplanır.

$$P(X|C_2)P(C_2)= P(X|Kabul=Hayır)P(Kabul=Hayır);$$

$$P(x_1|C_2)=P(Yaş=Orta|Kabul=Hayır)=\frac{1}{3}$$

$$P(x_2|C_2)=P(Eğitim=Yüksek|Kabul=Hayır)=\frac{1}{3}$$

$$P(x_3|C_2)=P(Cinsiyet=Kadın|Kabul=Hayır)=\frac{2}{3}$$

$$P(X|C_2)=P(X|Kabul=Hayır)=\frac{1}{3} \times \frac{1}{3} \times \frac{2}{3}=\frac{2}{27}$$

$$P(C_2)=P(Kabul=Hayır)=\frac{3}{8}$$

$$P(X|Kabul=Hayır)P(Kabul=Hayır)=\frac{2}{27} \times \frac{3}{8}=0,02$$

X veri grubunun, sınıf nitelik değerlerinin her birine ait olma olasılıkları hesaplanmıştır. Sonraki adım olarak naive bayes sınıflandırıcısında kullanılan en büyük sonrasal sınıflandırma yaklaşımı ile sınıflandırma işlemi tamamlanır.

$$\arg \max_{C_i} \{P(X|C_i)P(C_i)\}=\{0,03 , 0,02\}=0,03$$

Böylece, X veri grubunun sınıf niteliğinin, hesaplanan sınıf niteliği olasılıklarından en büyüğüne sahip olan “Evet” değeri olduğuna karar verilir.

2.3.1.2. Sayısal Nitelik Değerleri

Veri grubu niteliklerinin aldığı değerler sayısal değerler ise ilgili olasılıklar, nitelik değerlerinin frekansları ile hesaplanamaz. Bu noktada, olasılıkların hesaplanması için sayısal değerlerin beklenen değeri(ortalaması) μ , standart sapması σ olan normal dağılıma uyduğu varsayımından faydalanılır(Han ve Kamber, 2006: 312).

Bu varsayımdan dolayı, sayısal değerlere sahip niteliklerin her bir sınıf niteliği değerine göre koşullu olasılıkları, normal dağılım olasılık yoğunluk fonksiyonu kullanılarak hesaplanır.

$$P(x_k|C_i)=f(x_k,\mu_{C_i},\sigma_{C_i})=\frac{1}{\sqrt{2\pi\sigma_{C_i}}}e^{-\frac{(x_k-\mu_{C_i})^2}{2\sigma_{C_i}^2}}$$

Tablo 21: Kategorik Veri Kümesi

Başvuru	Yaş	Eğitim	Cinsiyet	Kabul
1	60	Orta	Erkek	Evet
2	22	İlk	Erkek	Hayır
3	38	Yüksek	Kadın	Hayır
4	40	Orta	Erkek	Evet
5	40	İlk	Erkek	Evet
6	60	Yüksek	Kadın	Evet
7	20	İlk	Kadın	Hayır
8	42	Orta	Kadın	Evet

Kaynak: Özkan, 2008: 175.

Tablo 21'deki eğitim verilerinden faydalanarak, X ; x_1 : Yaş=44, x_2 : Eğitim=Yüksek, x_3 : Cinsiyet=Kadın veri grubunun sınıf nitelik değeri şu şekilde bulunur. C_1 : Kabul=Evet, C_2 : Kabul=Hayır olmak üzere;

$$P(X|C_1)=P(X|Kabul=Evet)=P(Yaş=44|Kabul=Evet) \times \frac{1}{5} \times \frac{2}{5}$$

$$P(X|C_2)=P(X|Kabul=Hayır)=P(Yaş=44|Kabul=Hayır) \times \frac{1}{3} \times \frac{2}{3}$$

$P(\text{Yaş}=44|\text{Kabul}=\text{Evet})$ olasılığını hesaplamak için normal dağılım olasılık yoğunluk fonksiyonu kullanılır. Bu sebeple ilk olarak, ‘‘Kabul’’ niteliği ‘‘Evet’’ değerine sahip olan veri satırlarının ‘‘Yaş’’ değerlerinin ortalaması(μ) ve standart sapması(σ) hesaplanır. Bu değerler; $\mu_{C_1}=48,4$, $\sigma_{C_1}=10,6$ olarak hesaplanır. Olasılık yoğunluk fonksiyonunda değerler yerine koyulduğunda sonuç elde edilir.

$$P(\text{Yaş}=44|\text{Kabul}=\text{Evet})=\frac{1}{\sqrt{2\pi(10,6)}}e^{-\frac{(44-48,4)^2}{2(10,6)}} \approx 0,03$$

Bu sonucun ardından, veri örneğinin ‘‘Evet’’ değeri için koşullu olasılığı şu şekilde bulunur.

$$P(X|C_1)=P(X|\text{Kabul}=\text{Evet})=0,03 \times \frac{1}{5} \times \frac{2}{5}=0,0024$$

Aynı mantıktan hareketle, tüm işlemler $P(\text{Yaş} = 44|\text{Kabul} = \text{Hayır})$ için de yapılır. ‘‘Kabul’’ niteliği ‘‘Hayır’’ değerine sahip olan veri satırlarının ‘‘Yaş’’ değerlerinin ortalaması(μ)=26,6 ve standart sapması(σ)= 9,9 olarak bulunur. Bu değerler olasılık yoğunluk fonksiyonunda yerine koyulduğunda sonuç 0,009 olarak bulunur. Veri örneğinin ‘‘Hayır’’ değeri için koşullu olasılık ise 0,002 olarak elde edilir. Koşullu olasılıkların hesaplanmasının ardından, veri örneğinin hangi sınıf değerini alacağı en büyük sonrasal sınıflandırma yaklaşımıyla tahminlenir.

$$P(C_1)=P(\text{Kabul}=\text{Evet})=\frac{5}{8}$$

$$P(C_2)=P(\text{Kabul}=\text{Hayır})=\frac{3}{8}$$

$$P(X|\text{Kabul}=\text{Evet})P(\text{Kabul}=\text{Evet})=0,0024 \times \frac{5}{8}=0,0015$$

$$P(X|\text{Kabul}=\text{Hayır})P(\text{Kabul}=\text{Hayır})=0,002 \times \frac{3}{8} \approx 0,0008$$

$$\arg \max_{C_i} \{P(X|C_i)P(C_i)\}=\{0,0015 , 0,0008\}=0,0015$$

Hesaplanan olasılık sonuçlarına göre veri örneğinin, sınıf nitelik değeri olarak “Evet” değerini alma olasılığı, “Hayır” değerini alma olasılığından fazladır. Bu nedenle, X veri örneğinin sınıf niteliği değeri “Evet” olarak tahminlenir.

2.3.2. Olasılık Değerinin Sıfır Olması Sorunu

Naive bayes sınıflandırıcısı ile sınıflandırma sürecinde, sınıf niteliği değeri araştırılan bir veri grubu için, veri grubunun sahip olduğu nitelik değerlerinden herhangi birinin, dâhil olunması muhtemel olan bir sınıf niteliği değeri frekansının sıfır olması bir sorun oluşturmaktadır. İlgili frekansın sıfır olması, niteliğin ilgili sınıf niteliği değerine göre koşullu olasılığının sıfır olması anlamına gelmektedir.

Daha önce kategorik nitelik değerleri başlığında açıklanmış olan örnekten, herhangi bir nitelik için koşullu olasılığın sıfır olmasını ve bu durumun ortaya çıkardığı sonuçları incelemek için tekrar yararlanılmıştır. İlk olarak, örnekteki X veri grubunun sınıf niteliğinin “Evet” değerini alma olasılığı hesaplanmıştı. X veri grubunun niteliklerinin “Evet” sınıf değeri için koşullu olasılıkları şu şekilde bulunmuştu;

$$P(x_1|C_1)=P(\text{Yaş=Orta}|\text{Kabul=Evet})=\frac{3}{5}$$

$$P(x_2|C_1)=P(\text{Eğitim=Yüksek}|\text{Kabul=Evet})=\frac{1}{5}$$

$$P(x_3|C_1)=P(\text{Cinsiyet=Kadın}|\text{Kabul=Evet})=\frac{2}{5}$$

Burada verilen koşulan olasılıklardan ilki ele alınırsa, “Kabul=Evet” olan 5 veri satırı içinde “Yaş=Orta” olan satır sayısı yani frekansı 3 olduğu için koşullu olasılık değeri 3/5 olarak hesaplanmıştı. Bu aşamadan sonra niteliklerin birbirinden koşullu olarak bağımsız olduğu varsayımından yararlanarak X veri grubunun “Evet” sınıf değeri için koşullu olasılığı şu şekilde hesaplanmıştı.

$$P(X|C_1)=P(X|\text{Kabul=Evet})=\frac{3}{5} \times \frac{1}{5} \times \frac{2}{5} = \frac{6}{125}$$

Eğer “Kabul=Evet” olan 5 veri satırı içinde “Yaş=Orta” olan veri satırı olmasaydı, yani frekans sıfır olsaydı, “Yaş=Orta” niteliği için koşullu olasılık da sıfır olurdu.

$$P(x_1|C_1)=P(\text{Yaş=Orta}|\text{Kabul=Evet})=\frac{0}{5}=0$$

X veri grubunun “Evet” sınıf değeri için koşullu olasılığı hesaplaması da şu şekilde elde edilirdi;

$$P(x_1|C_1)=P(\text{Yaş=Orta}|\text{Kabul=Evet})=0$$

$$P(x_2|C_1)=P(\text{Eğitim=Yüksek}|\text{Kabul=Evet})=\frac{1}{5}$$

$$P(x_3|C_1)=P(\text{Cinsiyet=Kadın}|\text{Kabul=Evet})=\frac{2}{5}$$

$$P(X|C_1)=P(X|\text{Kabul=Evet})=0 \times \frac{1}{5} \times \frac{2}{5}=0$$

Buradaki sorun, bir koşullu olasılığın sıfır olması, hesaplamada diğer olasılıkların etkisini ortadan kaldırmasıdır. Bu durum da sınıflama sürecinde sonuçların anlamsız olmasına sebep olmuştur.

Naive bayes sınıflandırıcısında böyle bir sorun gidermek için fransız matematikçi Pierre Laplace tarafından ortaya koyulmuş Laplace düzeltmesi yapıldıktan sonra ilgili olasılık hesaplanır. Bu düzeltmede, sadece hesaplamada, sıfır olasılığa sahip olan niteliğin, her bir değerinin frekansına 1 eklenir. Frekanslara 1 eklenmesinden dolayı, olasılıkların hesaplanmasındaki paydalara da 1 eklenmelidir(Han ve Kamber, 2006: 314).

Örneğin; “Kabul=Evet” olan 1000 tane veri olsun. 1000 veri içinden “Yaş=Genç” değerine sahip hiç veri olmasın yani 0 veri, “Yaş=Orta” değerine sahip 10 veri ve “Yaş=Yaşlı” değerine sahip 990 veri olsun. Verilen bu frekans bilgileri ışığında hesaplanacak “Yaş” niteliği için koşullu olasılıklar sırasıyla; $(0/1000)=0$, $(10/1000)=0,010$ ve $(990/1000)=0,990$ ’dur. Buradaki sıfır olasılık sorununu gidermek için Laplace düzeltmesi yapıldığında; her bir “Yaş” niteliği

değerinin frekansına bir eklenir ve olasılıklar sırasıyla; $(1/1003) \square 0,001, (11/1003) \square 0,011$ ve $(991/1003) \square 0,988$ olarak hesaplanır.

Hesaplanan olasılıkların, düzeltmeden önceki ve düzeltmeden sonraki değerlerine bakıldığında birbirlerine oldukça yakın olduğu fark edilir. Değerlerin yakın olması, yapılan düzeltmenin sınıflama işleminin sonucunu değiştirecek bir işlem olmadığını göstermektedir. Fakat düzeltmenin esas anlamı ve sebebi, sıfır değerinin yutan eleman olmasından dolayı diğer olasılık değerlerini ve etkisini yok etmesinin önüne geçmesidir.

2.4. BAYES AĞLARI

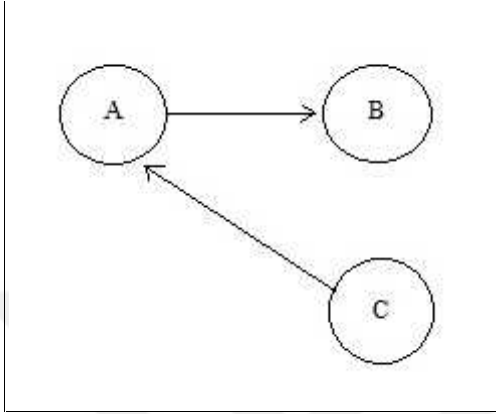
İnanç ağları olarak da bilinen bayes ağları, yapı itibarıyla yönlendirilmiş çevrimsiz grafik modelleridir. Bu ağlar, değişkenlerin, bir diğer deyişle niteliklerin alt kümeleri arasındaki koşullu bağımsızlıkları ortaya koyarak sınıflandırma amacıyla kullanılabilen yöntemlerdendir. Bayes ağları, yönlendirilmiş çevrimsiz bir grafik ve değişkenler arasındaki koşullu olasılık değerler olmak üzere iki yapıdan oluşmaktadır. Bu yöntemde nitelikler düğümlerle, nitelikler arasındaki olasılıksal ilişkiler de dallarla sembolize edilmektedir. Bahsedilen olasılıklar istatistiksel hesaplama yöntemleri ile elde edilebilirler(Akkoç, 2012: 8).

Bayes ağlarda, başka bir düğümü etkileyen düğümlere ebeveyn düğüm adı verilmektedir. Düğümlerin ağdaki pozisyonu ve diğer düğümlerle olan etkileşim yönü çok önemlidir. Eksik ya da yanlış oluşturulmuş bir ağ, hem düğümler arası ilişkilerden hesaplanması gereken olasılıkların yanlış bulunmasına dolayısıyla ağın kullanım amacına hizmet etmemesine sebep olacaktır.

Bir bayes ağının yapısını ortaya koymak için kullanılan yaklaşımlardan birisi skor yaklaşımıdır. Bu yaklaşımda veri yapısını ifade edebilme yeteneğine göre, oluşturulan alternatif ağlar, skor elde eder ve çeşitli algoritmalarla ağın skoru mümkün olduğunca artırmaya çalışılarak optimum ağ yapısı elde edilmeye çalışılır. Uygun bir ağ yapısının elde edilebilmesi için Cooper ve Herskovit tarafından geliştirilen K2 algoritması, skor tabanlı algoritmalarla örnek olarak gösterilebilir. Diğer bir yaklaşım da nitelikler arası koşullu bağımsızlık ilişkilerinin belirlenmesi ile ağ yapısının oluşturulmasıdır(Akkoç, 2012: 18).

Araştırma konusu ile ilgili ağ yapısı ve olasılık tablosu oluşturulduktan sonra karar verme, sınıflandırma işlevi yerine getirilebilir. Şekil 12’ de basit bir bayes ağı örneği verilmiştir.

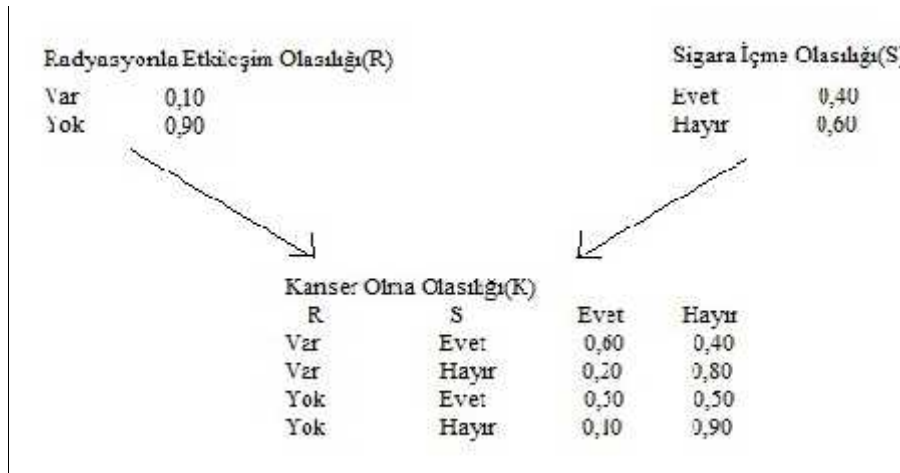
Şekil 12: Bayes Ağı



Şekil 12’deki ağda C düğümü A düğümünün, A düğümü de B düğümün ebeveynidir. Bu ilişkiden yola çıkarak ağın koşullu olasılık dağılımı; aşağıdaki gibi elde edilir. Aşağıdaki denklemden de anlaşılabacağı gibi bir alt düğüm olasılıkları sadece kendi ebeveynlerine bağlıdır. Naive bayes algoritmasından farklı olarak ağdaki herhangi bir düğüm sınıf nitelik olarak seçilebilir.

$$P(A,B,C)=P(C) \times P(A|C) \times P(B|A)$$

Şekil 13: Sigara, Radyasyon ve Kanser Ağı



Kaynak: Özkan, 2008: 179.

Şekil 13'teki bayes ağında, “Sigara” ve “Radyasyon” düğümü, “Kanser” düğümünün ebeveyni konumundadır. Düğümler arası koşullu ilişkilerle ilgili olasılıklar da şekilde verilmiştir. Bu ağ özelinde, kanser olduğu bilinen bir hastanın sigara içen biri olma olasılığı şu şekilde elde edilir.

$P(S=Evete|K=Evete)$ ve $P(S=Hayır|K=Evete)$ olasılıkları elde edildikten sonra en büyük sonrasal sınıflandırma yaklaşımıyla aranan olasılık bir diğer deyişle kanserli bir hastanın sigara niteliğinin hangi sınıfa ait olduğu bulunacaktır. İlk olarak $P(K=Evete)$ önsel olasılığı elde edilsin.

$$\begin{aligned} P(K=Evete) &= P(K=Evete|R=Var, S=Evete)P(R=Var, S=Evete) \\ &+ P(K=Evete|R=Var, S=Hayır)P(R=Var, S=Hayır) \\ &+ P(K=Evete|R=Yok, S=Evete)P(R=Yok, S=Evete) \\ &+ P(K=Evete|R=Yok, S=Hayır)P(R=Yok, S=Hayır) \end{aligned}$$

Önsel olasılık denkleminde yer alan koşullu olasılık ifadelerinin değerleri Şekil 13'te belirtilmiştir. Denklemden geriye kalan “Radyasyon” ve “Sigara” düğümleri ile ilgili olasılık değerleri, bu iki düğümün bağımsız olmasından dolayı ilgili düğümlerin aldığı değerlere ait olasılıkların çarpımı ile bulunur. Örneğin;

$$P(R=Var, S=Evete) = P(R=Var)P(S=Evete) = 0,10 \times 0,40 = 0,04$$

Bu mantıktan hareketle; bulunan değerler denklemden yerine koyulduğunda $P(K=Evete)$ önsel olasılığı 0,27 olarak elde edilir. Bu işlemin ardından, $P(S=Evete|K=Evete)$ ve $P(S=Hayır|K=Evete)$ önsel olasılıkları hesaplınsın.

$$P(S=Evete|K=Evete) = \frac{P(S=Evete, K=Evete)}{P(K=Evete)}$$

$$P(S=Evete, K=Evete) = P(S=Evete, R=Var, K=Evete) + P(S=Evete, R=Yok, K=Evete)$$

$$\begin{aligned} P(S=Evete, R=Var, K=Evete) &= P(S=Evete)P(R=Var)P(K=Evete | S=Evete, R=Var) \\ &= 0,40 \times 0,10 \times 0,60 = 0,024 \end{aligned}$$

$$P(S=Evete, R=Yok, K=Evete) = P(S=Evete)P(R=Yok)P(K=Evete | S=Evete, R=Yok)$$

$$=0,40 \times 0,90 \times 0,50 = 0,18$$

$$P(S=\text{Evet}, K=\text{Evet}) = 0,024 + 0,18 = 0,204$$

$$P(S=\text{Evet} | K=\text{Evet}) = \frac{P(S=\text{Evet}, K=\text{Evet})}{P(K=\text{Evet})} = \frac{0,204}{0,27} = 0,76$$

$$P(S=\text{Hayır} | K=\text{Evet}) = \frac{P(S=\text{Hayır}, K=\text{Evet})}{P(K=\text{Evet})}$$

$$P(S=\text{Hayır}, K=\text{Evet}) = P(S=\text{Hayır}, R=\text{Var}, K=\text{Evet}) + P(S=\text{Hayır}, R=\text{Yok}, K=\text{Evet})$$

$$P(S=\text{Hayır}, R=\text{Var}, K=\text{Evet}) = P(S=\text{Hayır})P(R=\text{Var})P(K=\text{Evet} | S=\text{Hayır}, R=\text{Var})$$

$$= 0,60 \times 0,10 \times 0,20 = 0,012$$

$$P(S=\text{Hayır}, R=\text{Yok}, K=\text{Evet}) = P(S=\text{Hayır})P(R=\text{Yok})P(K=\text{Evet} | S=\text{Hayır}, R=\text{Yok})$$

$$= 0,60 \times 0,90 \times 0,10 = 0,054$$

$$P(S=\text{Hayır}, K=\text{Evet}) = 0,012 + 0,054 = 0,066$$

$$P(S=\text{Hayır} | K=\text{Evet}) = \frac{P(S=\text{Hayır}, K=\text{Evet})}{P(K=\text{Evet})} = \frac{0,066}{0,27} = 0,24$$

$$\arg \max_{C_i} \{P(X|C_i)P(C_i)\} = \max\{0,76; 0,24\} = 0,76$$

Böylece kanser hastası olduğu bilinen bir kişinin, sigara içen ve sigara içmeyen bir kişi olması olasılığı hesaplanmış yanı sıra en büyük sonrasal sınıflandırma yaklaşımı ile kanser hastası olan bir kişi için, “Sigara” nitelik değeri sigara içme olasılığı daha fazla olduğundan “Evet” sınıfının tahminlenmiştir.

2.5. BELLEK TABANLI SINIFLANDIRMA

Veri madenciliğinde yaygın bir kullanım alanına sahip olan yöntemlerden birisi de bellek tabanlı sınıflandırmadır. Bu yöntemde kullanılan ve yöntemin felsefesini en iyi şekilde anlatan algoritma en yakın k-komşu algoritmasıdır. Algoritma, bir noktanın en yakınındaki k tane komşusuyla aynı sınıf değerine sahip

olması prensibine dayanmaktadır(Şentürk, 2011: 25). Sınıf değeri belli olan bir örnek setinin gözlem değerlerinden faydalanılarak, yeni bir gözlemin sınıf değerini tahmin etmek için yeni gözlem ile örnek setindeki gözlemler arasındaki uzaklıklar hesaplanır. Hesaplanan uzaklıklar arasında en küçük değere sahip k adet örnek gözlemin sınıf değeri, yeni gözlem değerinin sınıf değeri olarak atanır.

Uzaklıkların hesaplanması ile ilgili Chebyshev, Manhattan, Minkowski ve Öklid uzaklığı gibi farklı yaklaşımlar mevcuttur(Witten ve diğerleri, 2011: 472). Bu yaklaşımlardan en çok kullanılan uzaklık bağıntısı Öklid uzaklığıdır. Öklid uzaklığı aşağıdaki formülle hesaplanmaktadır.

$$d(i, j) = \sqrt{\sum_{k=1}^p (x_{ij} - x_{jk})^2}$$

Sınıf değeri araştırılan gözleme en yakın komşu sayısını gösteren k belirlenir. Araştırma konusu olan gözlem ile diğer tüm örnek gözlem değerleri arasındaki uzaklıklar, Öklid uzaklığı formülü bulunur. Hesaplanan uzaklıklar arasından en küçük k tanesi seçilir. Seçilen k tane gözlemin sınıf değerleri kontrol edilir ve en fazla yinelenmiş olan sınıf değeri, araştırılan gözlemin sınıf değeri olarak tahmin edilir.

ÜÇÜNCÜ BÖLÜM

UYGULAMA

Çalışmanın bu bölümünde, Beşiktaş futbol takımının 2015/2016 futbol sezondaki verilerinden ve Weka veri madenciliği programından faydalanarak, naive bayes ve C4.5 algoritması ile maç sonuçları sınıflandırılacaktır. Sınıflandırma aşamasından önce, ilgili veriler ve Weka paket programı kısaca tanıtılacaktır.

3.1. VERİ SETİNİN TANITILMASI

2015/2016 sezonundan elde edilmiş veriler bir tanesi sınıf nitelik olmak üzere toplam 11 adet nitelik için düzenlenmiştir. Bu nitelikler; “isabetli pas”, “isabetli orta”, “isabetli şut”, “atılan gol”, “takıma yapılan faul”, “top kapma”, “takımın yaptığı faul”, “top kaybı”, “topa sahip olma oranı”, “maç sahası” ve sınıf nitelik olarak “maç sonucu” şeklinde belirlenmiştir.

Şekil 14: 2015/2016 Sezonu Beşiktaş Futbol Takımı Verileri

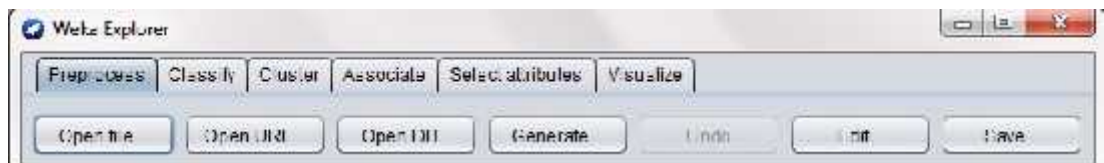
Hafta	IsabetliPas	IsabetliCrtı	IsabetliSut	AtılanGol	TakimaYapılanFaul	TopKapma	TakiminYaptigiFaul	TopKaybi	TopaSahipOlmaOrani (%)	MacSahasi	MacSonucu
1	377	1	10	5	21	13	12	5	54	DIS	GALIBIYET
2	445	4	3	1	18	13	15	19	56	IC	MAGLUBIYET
3	385	4	5	4	22	15	16	13	62	DIS	GALIBIYET
4	346	4	7	2	20	14	14	14	52	IC	GALIBIYET
5	441	13	7	1	27	9	11	16	68	DIS	BERABERLIK
6	255	6	2	3	21	12	19	13	44	IC	GALIBIYET
7	418	1	4	2	18	23	16	20	58	DIS	GALIBIYET
8	527	5	5	1	20	13	10	16	66	IC	GALIBIYET
9	405	3	10	5	17	19	18	13	55	DIS	GALIBIYET
10	375	1	7	3	19	18	15	12	63	IC	BERABERLIK
11	243	2	2	1	20	10	20	14	48	DIS	GALIBIYET
12	316	8	7	2	22	21	15	11	53	IC	GALIBIYET
13	505	16	5	0	21	8	10	16	67	IC	MAGLUBIYET
14	330	1	3	2	15	17	14	14	48	DIS	GALIBIYET
15	324	6	9	2	25	15	17	18	49	IC	GALIBIYET
16	358	3	3	3	27	9	10	14	56	DIS	GALIBIYET
17	376	6	9	4	14	10	14	15	52	IC	GALIBIYET
18	463	6	9	4	14	19	17	19	61	IC	GALIBIYET
19	449	7	6	2	22	11	10	11	63	DIS	BERABERLIK
20	416	6	4	1	28	10	8	15	64	IC	GALIBIYET
21	447	10	7	1	19	13	14	8	64	IC	GALIBIYET
22	326	7	4	0	17	10	19	17	57	DIS	MAGLUBIYET
23	533	7	9	3	11	12	16	11	63	IC	GALIBIYET
24	429	10	10	2	16	11	7	18	56	DIS	GALIBIYET
25	440	9	7	2	15	7	9	16	61	DIS	GALIBIYET
26	406	3	2	1	17	13	13	16	53	IC	GALIBIYET
27	443	9	4	1	23	13	16	12	68	DIS	MAGLUBIYET
28	409	7	9	3	25	16	12	17	63	IC	GALIBIYET
29	424	1	2	2	19	12	8	8	61	DIS	GALIBIYET
30	464	4	6	3	24	15	18	13	62	DIS	BERABERLIK
31	527	10	11	4	16	13	15	7	65	IC	GALIBIYET
32	336	4	10	1	13	5	13	15	43	DIS	GALIBIYET
33	412	7	4	3	20	21	17	17	59	IC	GALIBIYET
34	431	4	5	1	16	11	9	12	56	DIS	MAGLUBIYET

Belirlenen niteliklerin ilgili sezonda aldığı değerler ve tüm veri seti şekil 14’te gösterilmiştir.

3.2. WEKA PROGRAMININ TANITILMASI

Weka programı “.arff” uzantılı dosyalarla çalışmaktadır. Açılış ekranının ilk başında programın fonksiyonları gösterilmektedir.

Şekil 15: Açılış Ekranı

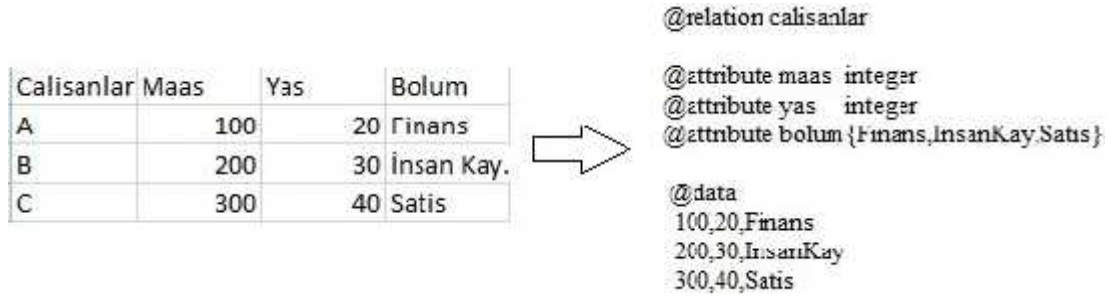


“Preprocess” fonksiyonu ile yüklenen veri setindeki niteliklerle ilgili değişiklikler ya da düzenlemeler yapılırken, “Classify” fonksiyonu ile sınıflandırma işlemleri, “Cluster” ile kümeleme işlemleri, “Associate” ile birliktelik kuralları işlemleri yapılabilmektedir. “Open file” sekmesi ile kaydedilmiş bir veri seti programa yüklenir. “Open URL” sekmesi ile bir internet sayfasına bağlanarak veri seti yüklenebilir. “Open DB” sekmesi bir veritabanından veri yüklemesi yapmak için kullanılırken “Generate” sekmesi ile program açıkken veri seti oluşturulabilir. “Edit” sekmesi ile yüklenmiş olan veri setinin tamamı görüntülenebilir ve istenirse veriler üstünde değişiklik yapılabilir. “Save” sekmesi, yapılan değişiklik veya yeni oluşturulmuş bir veri seti yapısını kaydetme işlemi için kullanılır.

Weka’da tanımlanan veri türleri; “numeric”, “date”, “string” ve “nominal” olarak sıralanabilir. “numeric” veriler, reel sayılardan oluşan veriler için “real”, tamsayılardan oluşan veriler için “integer” olarak ikiye ayrılır. “date” tarih verileri için, “string” ise dizgi olarak dilimizde karşılık bulmuş, metin verisi tanımlamak için kullanılan nitelik türüdür. “nominal” nitelik türü ise bir kümenin elemanları şeklinde tanımlanan verileri tanımlamak için kullanılır.

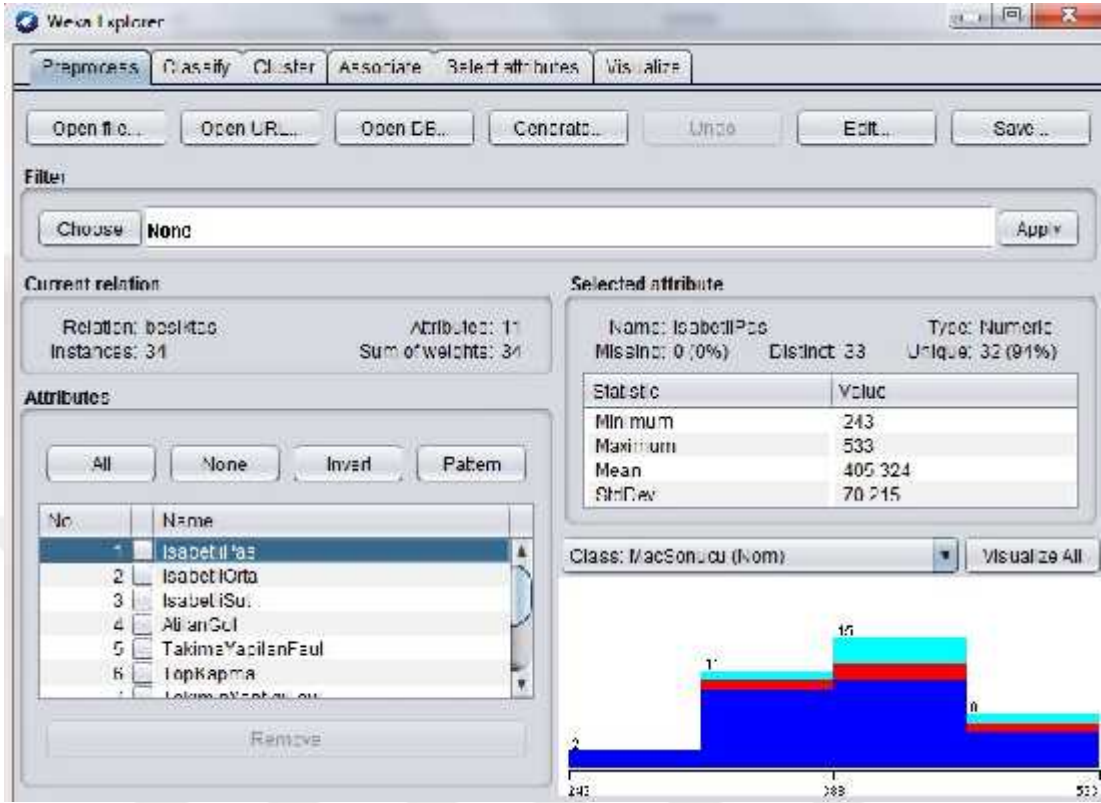
Weka programı, veri seti oluşturulurken komut satırlarını benzer bir mantıkta çalışan ifadelerle çalışır. İlk olarak “@relation” ifadesi ile birlikte veri setine bir isim verilir. Ardından veri setinin sahip olduğu nitelik sayısınca, her satıra bir nitelik gelecek şekilde “@attribute” ifadesi kullanılır ve niteliğe isim verilir. İsim verilmesinin ardından her bir niteliğin türü de kendi satırında verilir. Son olarak “@data” komutuyla her bir niteliği ait veriler virgül ile ayrılarak niteliklere ait sütunlara yazılır. Bu şekilde hazırlanan dosya “.arff” uzantısıyla kaydedilerek Weka programında kullanılabilecek bir veri seti haline getirilir. Burada bahsedilen yöntemle şekil 16’da bir çalışanlar örnek veri seti oluşturulmuştur. Bunun dışında yukarıda bahsedilen sekmeler ile veri seti yüklenebilir.

Şekil 16: Veri Seti Oluşturma



Bir veri seti yüklendikten sonra açılış ekranında yüklenen veri seti ile ilgili bilgiler görsel olarak program tarafından kullanıcıya sunulmaktadır. Kullanıcı bu aşamada veri setinde bulunan niteliklerle alakalı bilgileri elde edebilir. Weka programında her veri seti “relation” kısmındaki açıklamayla isimlendirilir. “Current relation” çerçevesinde “instances” kayıt sayısını “attribute” ise veri setinin kaç adet nitelikten oluştuğunu göstermektedir. “Attributes” çerçevesinde var olan nitelikler, “Selected Attribute” çerçevesinde ise seçili olan nitelikle alakalı istatistiksel özet bilgiler bulunur. Yüklenmiş olan veri setindeki niteliklerden hangisinin sınıf nitelik olarak belirleneceği karar verildikten sonra “Class” sekmesinde kararlaştırılan nitelik seçilerek, sınıf niteliği belirlenir.

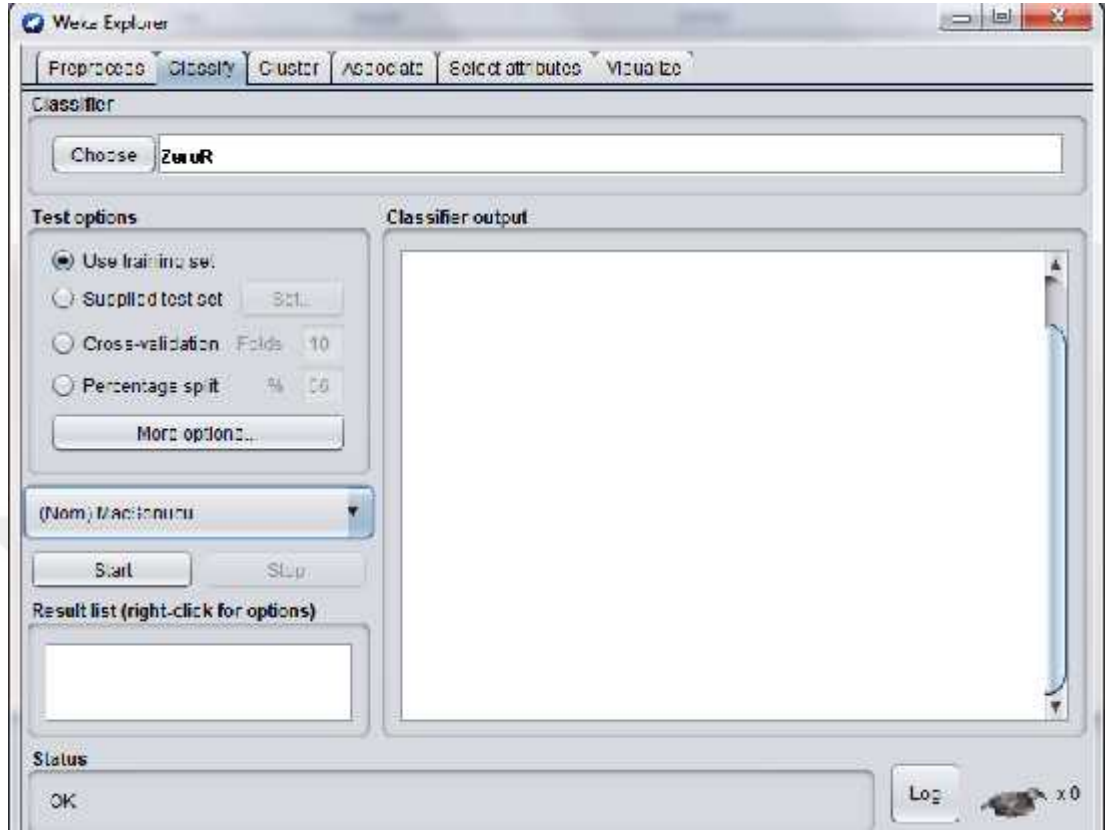
Şekil 17: Veri Setinin Yüklmesi



Weka programına Beşiktaş futbol takımının 2015/2016 sezonuna ait bu bölüm başında bahsedilen 11 nitelik ve niteliklere ait verilerden oluşturulmuş veri seti yüklenmiştir. Şekil 17’deki “Class” sekmesinin aldığı değerden de anlaşılacağı gibi veri集中的 sınıf niteliği “MacSonucu” niteliğidir.

Uygulamada da sınıflandırma işlemi gerçekleştirileceği için, “Classify” fonksiyonuna ait pencere daha detaylı olarak ele alınacaktır. İlgili sekme tıklandığında Şekil 18’deki ekran ile karşılaşılacaktır.

Şekil 18: “Classify” Sekmesi

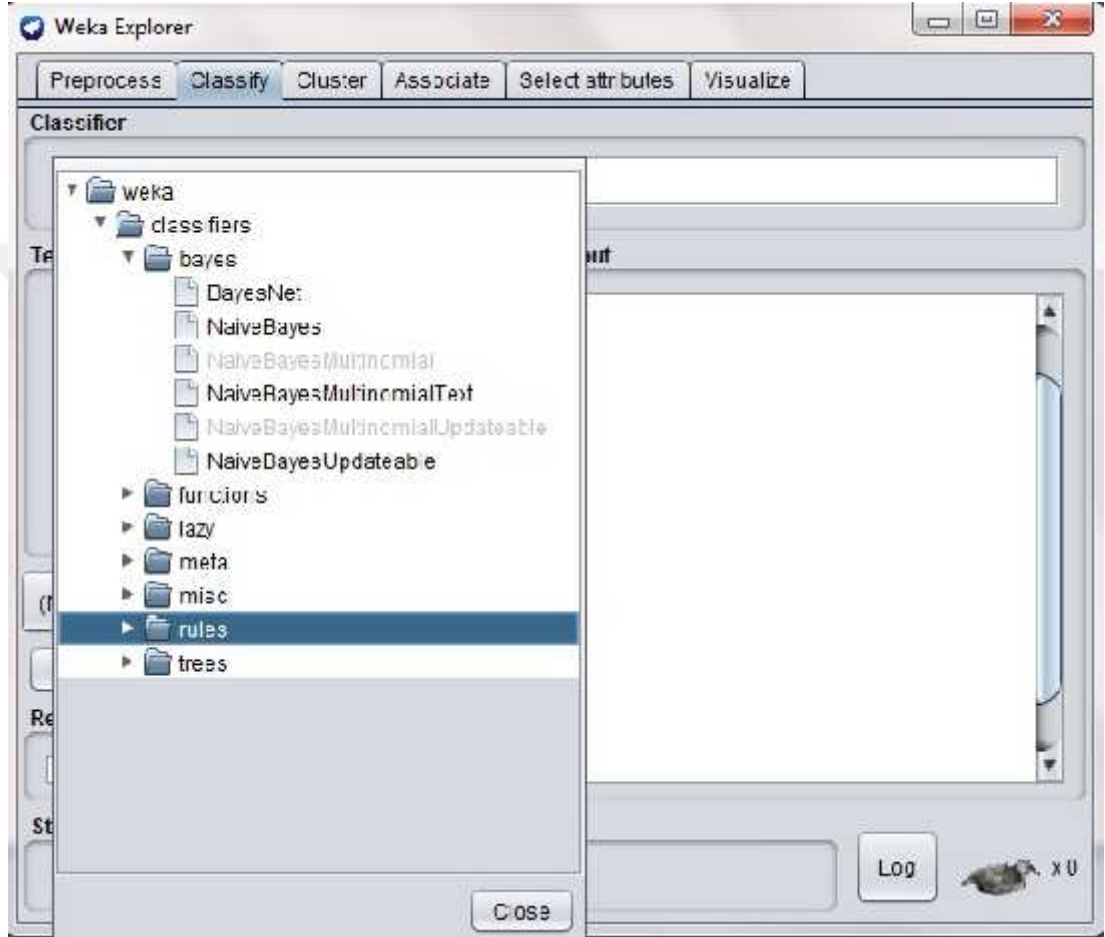


“Test Options” çerçevesi yüklenen veri setinin, eğitim ve test verisi bakımından nasıl bir dağılıma sahip olacağının belirlendiği kısımdır. “Use training set” seçeneği, veri setinin tamamının hem eğitim verisi hem test verisi olarak kullanılmasını sağlamaktadır. “Supplied test set”, kullanıcı tarafından test verisi başka bir dosyadan sağlamak istenirse kullanılması gereken seçenektir. “Cross-validation” bu seçeneğe girilen “Fold” değeri kadar veri seti bölünür. Örneğin Fold=10 için veri seti 10 parçaya bölünür 9 parça eğitim verisi olarak 1 parçası da test verisi olarak kullanılır ve parçalar sürekli değiştirilir. “Percentage split” ise veri setinin girilen yüzde değer kadarı eğitim verisi, geriye kalan kısmın test verisi olarak kullanılmasını sağlayan seçenektir.

“Classifier” çerçevesindeki “Choose” sekmesi, sınıflama işlemini yaparken kullanılacak olan algoritma belirlenmektedir. Şekil 19’da Weka programının 3.8.0. versiyonunda yer alan bazı algoritmalar gösterilmiştir. Veri seti yüklendikten sonra algoritma seçme aşamasında koyu bir yazı ile gösterilen algoritmalar, yüklenen veri setinin yapısına uygun olanları koyu olmayanlar ise veri setinin yapısına uygun

olmayan algoritmaları simgelemektedir. Program içerisinde algoritmalar çeşitli başlıklar altında toplanmıştır. Örneğin; naive bayes algoritması “bayes” klasörü içinde yer almaktadır. Weka’da bazı algoritmalar için farklı isimler kullanılmıştır. Örneğin; C4.5 algoritması “trees” klasöründe “J48” adıyla yer almaktadır.

Şekil 19: Sınıflandırma Algoritmaları



Kullanılacak olan sınıflandırma algoritması seçildikten sonra “Start” komutu ile sınıflandırma işlemi gerçekleştirilir. Sınıflandırma işlemi gerçekleştirildikten sonra yapılan işlemin sonuçları “Classifier Output” penceresinde gösterilir. Aynı adımlar kümeleme ve birliktelik kuralları işlemleri yapmak için de geçerlidir. Sol altta bulunan “Result list” çerçevesinde yapılan işlemlerin zamanı ve hangi algoritmanın kullanıldığı bilgisi yer almaktadır. Yapılan tüm işlemlerin sonuçları program kapatılmadığı sürece o pencerede kayıtlı olmakla beraber istenilen işlem tıklandığında, tıklanan işlemle ilgili sonuçlar “output” penceresinde gösterilir.

Ayrıca “Result list” de yer alan bir işlem sonucunu sağ tıklayarak sonuçla ilgili grafik görseller başta olmak üzere modeli kaydetmek, sonuçları başka bir pencerede açmak gibi farklı fonksiyonlara da erişim sağlanabilmektedir.

Beşiktaş futbol takımının 2015/2016 sezonundaki verilerinden oluşmuş 11 niteliğe sahip veri seti Weka programına yüklenmiştir. “Test Option” seçeneği olarak “Cross-Validation” seçeneği kullanılmış ve “Fold” değeri 10 olarak girilmiştir. Böylece tüm veri seti 10 parçaya bölünmüş 9 parçası eğitim, kalan 1 parçası da test verisi olarak değerlendirilmiştir. Weka programı, seçeneğin sahip olduğu fonksiyon gereği her bir 10 parçayı birbirinden farklı olacak şekilde sürekli değiştirerek eğitim yapmıştır.

Özellikle futbol gibi sportif branşlarda, takım ve bireysel sporcu psikolojisi, teknik kafiye kalitesi ve motivasyonu, yönetim, medya gibi branşla ilgili veya branştan bağımsız birçok faktör, takımların müsabaka performanslarını hem olumlu hem de olumsuz yönde etkilemektedir. Böylece takımların aldıkları iyi ve kötü sonuçların altındaki sebeplerin tümünün tespiti homojen olarak yapılamamaktadır. Takımların yukarıda bahsedilen sebeplerden kaynaklı, performans ve sonuç anlamında sadece geçici iyi veya sadece geçici kötü oldukları dönemlere ait verilerin eğitim verisi olarak değerlendirilmesi sonuçlar açısından sağlıklı olmayacaktır. Bundan dolayı, veri setini her seferinde farklı bölümlere ayırıp, farklı parçaları kullanarak eğitim yapması tespitler yapılması daha makuldür.

3.3. VERİ SETİNİN NAİVE BAYES ALGORİTMASI İLE SINIFLANDIRILMASI

İlk önce naive bayes algoritmasıyla sonrasında C4.5 algoritmasıyla sınıflandırma yapılacaktır. Naive bayes algoritması seçilip, sınıflandırma işlemi yapılmış ve sonuçlar Şekil 20, Şekil 21 ve Şekil 22’de gösterilmiştir.

Şekil 20: Veri Seti ve Sahip Olduğu Nitelikler

```
=== Run information ===

Scheme:      weka.classifiers.bayes.NaiveBayes
Relation:    besiktas
Instances:   34
Attributes:  11
              IsabetliPas
              IsabetliOrta
              IsabetliSut
              AtilanGol
              TakimaYapilanFaul
              TopKapma
              TakiminYaptigiFaul
              TopKaybi
              TopaSahipOlmaOrani
              MacSahasi
              MacSonucu
Test mode:   10-fold cross-validation

=== Classifier model (full training set) ===

Naive Bayes Classifier
```

Önceki başlıklarda bahsedildiği gibi, Şekil 20’de Beşiktaş futbol takımına ait veri seti, yapılan test ve testi yapmak üzere kullanılan algoritma ile ilgili bazı bilgiler gösterilmiştir. “besiktas” veri seti 34 kayıttan ve şekilde listelenmiş olan 11 nitelikten oluşmaktadır. “Fold” değeri 10 olan “Cross-Validation” seçeneği, test modu olarak belirlenmiştir. “Scheme” satırındaki açıklamada, sınıflandırma fonksiyonu ve sınıflandırma algoritması olarak “bayes” klasöründeki naive bayes algoritmasının kullanıldığı belirtilmiştir.

Şekil 21: Niteliklerle İlgili Bilgiler

Attribute	Class			Attribute	Class		
	GALIBİYET (0.7)	BERABERLİK (0.14)	MAĞLUBİYET (0.16)		GALIBİYET (0.7)	BERABERLİK (0.14)	MAĞLUBİYET (0.16)
IsBelirliPas				TakimeYapilanFaul			
mean	398.85	432.7344	431.375	mean	19.04	22.95	19.04
std. dev.	72.849	36.8948	58.6197	std. dev.	4.3952	2.9308	2.6239
weight sum	25	4	5	weight sum	25	4	5
precision	9.0625	9.0625	9.0625	precision	1.1333	1.1333	1.1333
IsBelirliOrta				TopKapsa			
mean	5.0727	6.4773	6.1010	mean	13.728	13.2	11.04
std. dev.	2.7414	4.5611	4.5636	std. dev.	4.4204	0.2663	1.92
weight sum	25	4	5	weight sum	25	4	5
precision	1.3636	1.3636	1.3636	precision	1.2	1.2	1.2
IsBelirliSut				TakimeYapilanFaul			
mean	5.48	6.1875	4.275	mean	13.76	13.5	13.8
std. dev.	2.9656	0.8628	0.46	std. dev.	3.4903	3.2016	3.763
weight sum	25	4	5	weight sum	25	4	5
precision	1.125	1.125	1.125	precision	1	1	1
AtilanGol				TopKaybi			
mean	2.52	2.22	0.6	mean	18.5	13.125	15.6
std. dev.	1.2368	0.8282	0.4899	std. dev.	3.7637	1.875	2.5739
weight sum	25	4	5	weight sum	25	4	5
precision	1	1	1	precision	1.25	1.25	1.25
TopaSahipOlmaOrani				Macbando			
mean	0.5637	0.6414	0.7195	DTB	12.0	4.0	4.0
std. dev.	0.066	0.0253	0.0549	IC	15.0	2.0	3.0
weight sum	25	4	5	[total]	27.0	6.0	7.0
precision	0.0132	0.0132	0.0132				

Şekil 20'deki niteliklere ait “mean=ortalama”, “std. dev.= standart sapma” ve “weight sum= ağırlık toplamı” gibi istatistiksel özetler, sınıf niteliğın aldığı değerlere göre gruplandırılmıştır. Böylece, maçların sonuçlarına göre niteliklerle ilgili genel bir fikir elde edilebilmektedir. Sınıf niteliğının aldığı değerlerin altında parantez içinde belirtilen sayılar, sınıf niteliğı değerlerine ait olasılıklardır.

Şekil 22: Sınıflandırma Sonucu

```
Time taken to build model: 0 seconds

== Stratified cross-validation ==
-- Summary --

Correctly Classified Instances      24          70.5882 %
Kappa statistic                    0.1237
Mean absolute error                 0.1527
Root mean squared error            0.403
Negative absolute error             64.1249 %
Root relative squared error        105.6204 %
Total Number of Instances         34

-- Detailed Accuracy By Class --

          TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
          0.920    0.075    0.767     0.920    0.836      0.195    0.795    0.923    GALIBIYET
          0.250    0.033    0.100     0.250    0.233      0.297    0.142    0.285    BERABERLIK
          0.000    0.065    0.000     0.000    0.000     -0.154    0.290    0.476    MAGLUBIYET
Weighted Avg.    0.706    0.588    0.423     0.706    0.554      0.163    0.780    0.782

-- Confusion Matrix --

a b c < classified as
23 0 2 | a = GALIBIYET
3 1 0 | b = BERABERLIK
4 1 0 | c = MAGLUBIYET
```

Şekil 22’de, belirlenmiş kriterlerle sınıflandırma işleminin sonucu gösterilmiştir. Gösterilen sonuçta en önemli kısım, “Correctly Classified Instances=doğru sınıflandırmış örnek sayısı” ve “Confusion Matrix”tir. “Correctly Classified Instances” 24’tür. Bunun anlamı, 34 tane maçın 24 tanesinin sonucu doğru sınıflandırılmış olduğudur. Aynı satırdaki %70,5882 oranı, doğru sınıflandırılmış örnek oranıdır.

“Confusion Matrix”, yapılan doğru ve yanlış sınıflandırmaların daha detaylı okunmasını sağlayacak ve gerçek sonuçlar ile sınıflandırma işlemiyle elde edilmiş sonuçların karşılaştırılabileceği kısımdır. Şekil 21’den daha rahat görülebildiği gibi 34 maçın 25 tanesinin galibiyetle, 4’ünün beraberlikle ve 5 tanesinin mağlubiyet ile sonuçlanmıştır. Matriste “a” ile galibiyet, “b” ile beraberlik ve “c” ile mağlubiyet simgelenmiştir. “Galibiyet” satırındaki 25 maçın, 23 tanesinin sonucu “a” yani galibiyet ve 2 tanesinin sonucu “c” yani mağlubiyet olarak sınıflandırılmıştır. Aynı mantıkla, naive bayes sınıflandırma algoritması, beraberlikle bitmiş 4 maçın sonucunu; 3 galibiyet, 1 beraberlik ve mağlubiyetle sona ermiş 5 maçın sonucunu da; 4 galibiyet, 1 beraberlik olarak sınıflandırmıştır.

Sonuç çıktısında, doğru sınıflandırılmış örnek sayısı ve oranı, “Confusion Matrix” sonuçlarının yanı sıra bazı istatistiksel değerler ve özellikle makine öğrenmesinde kullanılabilir, sınıf niteliği değerlerine göre düzenlenmiş istatistiksel bilgiler yer almaktadır.

3.4. VERİ SETİNİN C4.5 ALGORİTMASI İLE SINIFLANDIRILMASI

Naive bayes algoritması ile yapılan sınıflandırmanın ardından C4.5 algoritması ile sınıflandırma yapılmış, ilgili işlemin sonuçları ve ayrıca C4.5 algoritması bir karar ağacı algoritması olduğu için, sınıflandırma işlemiyle ortaya çıkan karar ağacı da verilmiştir.

Şekil 23: C4.5 Sınıflandırması

```
=== Run information ===

Scheme:      weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:    besiktas
Instances:   34
Attributes:  11
             IsabetliPas
             IsabetliOrta
             IsabetliSut
             AtılanGol
             TakimaYapılanFaul
             TopKapma
             TakiminYaptigiFaul
             TopKaybi
             TopaSahipOlmaOrani
             MacSahasi
             MacSonucu
Test mode:   10-fold cross-validation
```

Şekil 23'te naive bayes algoritmasında olduğu gibi, C4.5 algoritmasının uygulandığı veri seti ve bu veri setinin sahip olduğu nitelikler verilmiştir. “Scheme” kısmında bu veri seti için “J48” algoritmasının kullanıldığı bilgisi yer almıştır. Daha önce bahsedildiği gibi Weka’da C4.5 algoritması için J48 ismi kullanılmakta olduğu için bu şekilde belirtilmiştir.

Şekil 24: C4.5 Algoritması ile Oluşan Karar Ağacı

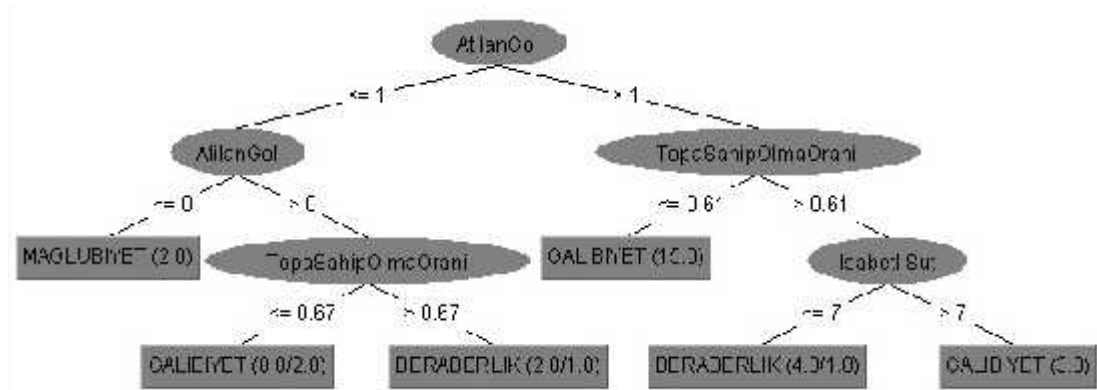
```
J48 pruned tree
-----

AtilanGol <= 1
|   AtilanGol <= 0: MAGLUBIYET (2.0)
|   AtilanGol > 0
|   |   TopaSahipOlmaOrani <= 0.67: GALIBIYET (8.0/2.0)
|   |   TopaSahipOlmaOrani > 0.67: BERABERLIK (2.0/1.0)
AtilanGol > 1
|   TopaSahipOlmaOrani <= 0.61: GALIBIYET (15.0)
|   TopaSahipOlmaOrani > 0.61
|   |   IsabetliSut <= 7: BERABERLIK (4.0/1.0)
|   |   IsabetliSut > 7: GALIBIYET (3.0)

Number of Leaves      :      6
Size of the tree      :     11
```

Şekil 24'te C4.5 algoritması ile veri setinin sınıflandırılmasının ardından oluşan karar kuralları belirtilmiştir. Toplam 11 elemandan oluşan karar ağacında 6 adet sonuç elemanı, diğer deyişle 6 adet yaprak bulunduğu bilgisi de şeklin son 2 satırında gösterilmiştir.

Şekil 25: Karar Ağacı



Şekil 25'de, C4.5 algoritmasıyla yapılan sınıflandırma işlemiyle oluşturulmuş karar ağacı gösterilmektedir. Oluşturulan karar ağacı ile üretilmiş karar kuralları şu şekildedir.

Eğer AtılanGol>1 ve TopaSahipOlmaOranı>0.61 ve İsabetliSut>7 ise
MacSonucu=Galibiyet

Eğer AtılanGol>1 ve TopaSahipOlmaOranı>0.61 ve İsabetliSut≤7 ise
MacSonucu=Beraberlik

Eğer AtılanGol>1 ve TopaSahipOlmaOranı≤0.61 ise MacSonucu=Galibiyet

Eğer AtılanGol=0 ise MacSonucu=Mağlubiyet

Eğer AtılanGol=1 ve TopaSahipOlmaOranı>0.67 ise MacSonucu=Beraberlik

Eğer AtılanGol=1 ve TopaSahipOlmaOranı≤0.67 ise MacSonucu=Galibiyet.

Şekil 26: C4.5 Sınıflandırması Sonuçları

```
Time taken to build model: 0.03 seconds

--- Stratified cross-validation ---
--- Summary ---

Correctly Classified Instances      17          50      %
Kappa statistic                    0.1333
Mean absolute error                 0.3486
Root mean squared error            0.5272
Relative absolute error            115.9535 %
Root relative squared error        138.1747 %
Total Number of Instances         34

=== Detailed Accuracy By Class ===

      TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
      0.640    0.840    0.667    0.640    0.653     0.241    0.350    0.280    GALIBIYET
      0.250    0.100    0.250    0.250    0.250     0.150    0.533    0.175    BERABERLIK
      0.000    0.200    0.000    0.000    0.000     0.792    0.444    0.154    MAGLUBIYET
Weighted Avg.    0.500    0.696    0.520    0.500    0.510     -0.188    0.400    0.514

--- Confusion Matrix ---

a b c <-- Classified as
10 3 6 | a - GALIBIYET
  3 1 1 | b - BERABERLIK
  5 0 0 | c - MAGLUBIYET
```

Şekil 26'da C4.5 algoritması ile sınıflandırılmış olan veri setine ait sınıflandırma sonuçları gösterilmiştir. Sonuçtan da anlaşılacağı gibi doğru sınıflandırılmış örnek sayısı 17'dir bu sayısı da veri setindeki kayıtların yarısına karşılık gelmektedir.

Confusion matrix'te galibiyetle sonuçlanmış 25 maçın, 16 tanesini galibiyet, 3 tanesini beraberlik ve 6 tanesini de mağlubiyet olarak sınıflandırmıştır. Beraberlikle sonuçlanmış 4 maçın, 3 tanesini galibiyet ve 1 tanesini beraberlik olarak sınıflandırmıştır. Mağlubiyetle sonuçlanmış 5 maçın, 5 tanesini de galibiyet olarak sınıflandırmıştır.

3.5. SINIFLANDIRMA ALGORİTMALARININ KARŞILAŞTIRILMASI İÇİN KULLANILAN PERFORMANS ÖLÇÜTLERİ

Veri madenciliğinde, algoritmaların sınıflandırma performanslarını değerlendirmek ve birbirleriyle kıyaslamak için bazı performans ölçümlerine başvurulmaktadır. Uygulamada kullanılan algoritmaların performanslarının karşılaştırılacağı kriterler; sınıflandırma süresi, doğruluk, kesinlik, duyarlılık, F-ölçütü ve ROC(Receiver Operating Characteristic) alanı değerleridir.

Bir sınıflama tekniğinin diğer bir deyişle modelin başarılı olma durumu doğru sınıfa ve yanlış sınıf atanmış örnek sayısı ile doğrudan ilgilidir. Uygulama bölümünde anlatılan “Confusion Matrix”, başarılı olma durumuna ilişkin bilgileri içeren sonuç çıktısıdır. Bahsedilen performans kriterleri de bu matristeki bilgilerin ışığında hesaplanmaktadır.

Tablo 22: Confusion Matrix Yapısı

		Tahminlenen Sınıf	
		Sınıf 1	Sınıf 0
Gerçek Sınıf	Sınıf 1	X	Y
	Sınıf 0	Z	K

Tablo 22’de gösterilen matriste x; TP= True Pozitif, y; FN= False negatif, z; FP= False Pozitif ve k; TN=True Negatif değerlerini göstermektedir. Örneğin x=10 değeri, gerçekte Sınıf 1’e üye olan ve tahminleme sonucunda yine Sınıf 1 atanmış olan 10 tane örnek olduğu anlamına gelir.

Sınıflandırma süresi kriteri; algoritmanın sınıflandırma işlemi için kullandığı süreyi belirtmektedir. Doğruluk ya da doğruluk oranı; doğru sınıflandırılmış örnek sayısının toplam örnek sayısına oranlanmasıyla elde edilir. Bu kriter, sınıflandırma sonucunun gerçek değerlere ne kadar yakın olduğunu gösteren bir orandır.

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

Kesinlik, sınıf 1 olarak sınıflandırılmış TP örnek adedinin, sınıf 1 olarak sınıflandırılmış tüm TP ve FP örnek sayısına bölünmesiyle bulunur. Kesinlik oranının araştırılmasının sebebi, aynı yöntemle elde edilmiş analiz sonuçlarının birbirine yakınlığının tespit edilmeye çalışılmasıdır.

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

Duyarlılık; TP olarak sınıflandırılmış örnek sayısının toplam pozitif TP ve FN toplamına oranıdır. Sınıflandırma işlemi sonuçlarının birbirine olan yakınlığının bir göstergesidir.

$$\text{Duyarlılık} = \frac{TP}{TP+FN}$$

F-ölçütü; duyarlılık ve kesinlik oranlarının harmonik ortalaması şeklinde oluşmaktadır. Duyarlılık ve kesinlik geçerli ve değerlendirme anlamında önemli kriterler olsa da tek başlarına yeterli kıyaslama kalitesinde olmayabilirler. Duyarlılık ve kesinlik oranları dengede olduğunda F-ölçütü yüksek değerlere sahip olur.

$$\text{F-ölçütü} = \frac{2 \times \text{Duyarlılık} \times \text{Kesinlik}}{\text{Duyarlılık} + \text{Kesinlik}}$$

ROC alanı; TP oranının Y ekseninde ve FP oranının X ekseninde karşılaştırıldığı 2-boyutlu bir grafiğin oluşturduğu alandır. Bu alan, veri madenciliği algoritmalarının karşılaştırmasında genel bir kullanıma sahip bir kriterdir. Bahsedilen bu kriterleri elle hesaplamak mümkün olduğu gibi, Weka programı bu kriterlerin aldığı değerleri hazır olarak vermektedir. Tablo 23'te naive bayes algoritması için hesaplanan performans ölçütü değerleri gösterilmiştir.

Tablo 23: Naive Bayes Algoritması Performans Ölçütü Değerleri

Naive Bayes Algoritması					
Sınıflandırması Süresi(Saniye)	Doğruluk	Kesinlik	Duyarlılık	F-Ölçütü	ROC Alanı
0.00	% 70,58	% 62,3	% 70,6	% 65,4	% 78

Tablo 24: C4.5 Algoritması Performans Ölçütü Değerleri

C4.5 Algoritması					
Sınıflandırması Süresi(Saniye)	Doğruluk	Kesinlik	Duyarlılık	F-Ölçütü	ROC Alanı
0.03 saniye	% 50	% 52	% 50	% 51	% 40

Tablo 24’te, C4.5 algoritması için, anlatılan performans ölçütlerinin sonuçları belirtilmiştir. Tablo 23 ve tablo 24’teki performans değerleri, algoritmaların sonuç çıktısında ağırlık ortalama yani “weighted avg.” satırında hazır hesaplanmış olarak da yer almaktadır.

SONUÇ VE ÖNERİLER

Yapılan tez çalışmasında, veri ve veri madenciliği kavramlarından bahsedilmesinin ardından, veri madenciliği yöntemleri anlatılmıştır. Veri madenciliği yöntemlerinden, sınıflandırma yöntemleri detaylı bir şekilde ele alınmıştır. Sınıflandırma algoritmalarından, karar ağacı algoritması olan C4.5 algoritması ve istatistiksel sınıflandırma araçlarından naive bayes algoritması, Beşiktaş futbol takımının 2015/2016 sezonundaki verileriyle oluşturulmuş veri setini sınıflandırmak ve sonuçlarını karşılaştırmak için kullanılmıştır. Veri seti, daha önceki başlıklarda bahsedilen, “isabetli pas”, “isabetli orta”, “isabetli şut”, “atılan gol”, “takıma yapılan faul”, “top kapma”, “takımın yaptığı faul”, “top kaybı”, “topa sahip olma oranı”, “maç sahası” ve sınıf nitelik olarak “maç sonucu” nitelikleri çerçevesinde düzenlenmiştir. Ardından veri seti, Weka veri madenciliği programına yüklenip naive bayes algoritması ve C4.5 algoritmasıyla sınıflandırılmıştır. İki algoritma için sınıflandırma sonuçları, bahsedilen performans ölçütlerinin aldığı değerler ışığında karşılaştırılmış ve naive bayes algoritmasının C4.5 algoritmasına göre, daha iyi sonuçlar ortaya koyduğu saptanmıştır.

Tablo 25’te naive bayes ve C4.5 algoritmalarına ait sınıflandırma sonuçlarına ait performans ölçütleri değerleri karşılaştırmalı olarak verilmiştir.

Tablo 25: Performans Kriterleri

Naive Bayes Algoritması					
0.00 saniye	% 70,58	% 62,3	% 70,6	% 65,4	% 78
Sınıflandırma Süresi	Doğruluk	Kesinlik	Duyarlılık	F-Ölçütü	ROC Alanı
0.03 saniye	% 50	% 52	% 50	% 51	% 40
C4.5 Algoritması					

Karşılaştırma kriterlerinin aldığı değerlere bakıldığında, C4.5 algoritması herhangi bir kriterde naive bayes algoritmasının daha üstün bir skor elde edememiştir. Çok basit ama önemli bir kriter olan doğruluk oranı başta olmak üzere, tüm kriterlerin aldığı değerler arasında naive bayes algoritması lehine büyük farklar

vardır. Açık bir şekilde, naive bayes algoritmasının, C4.5 algoritmasına göre minimum zamanda daha iyi bir sınıflandırma yaptığı görülmektedir.

Bu konuda, yapılacak çalışmalar için öneriler aşağıdaki gibidir;

-) Durağan(stabil) ve daha geniş hacimli bir veri setinin daha fazla sınıflandırma algoritmasıyla sınıflandırılıp sonuçların karşılaştırılması
-) Aynı sezonda ligde mücadele etmiş tüm takımların verilerinin sınıflandırma sonuçları ile takımların gerçek sonuçların karşılaştırılması
-) Bayes ağlar gibi nitelikler arası ilişkileri ortaya koyan bir yaklaşımla, nitelikler arasındaki ilişkiler ışığında, sınıf değeri bilinen veri kayıtlarının sınıf niteliği dışında herhangi bir nitelik için, tanımlı olduğu türdeki değerleri alma olasılıklarının ortaya koyularak sonuçların yorumlanması.

KAYNAKÇA

Akkoç, B. (2012). *Sosyal Ağ Analizi için Bayes Ağlarının Kullanımı*. (Yayınlanmamış Yüksek Lisans Tezi). Konya: Selçuk Üniversitesi Fen Bilimleri Enstitüsü.

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

Alan, M.A. (2012). Veri Madenciliği ve Lisansüstü Öğrenci Verileri Üzerine Bir Uygulama. *Dumlupınar Üniversitesi Sosyal Bilimler Dergisi*. 9(33): 165-174.

Ashari, A., Paryudi, I. ve Tjao, A.M. (2013). Performance Comparison Between Naive Bayes, Decision Tree and K-Nearest Neighbor in Searching Alternative Design in an Energy Simulation Tool. *International Journal of Advanced Computer Science and Applications*. 4(11): 33-39.

Ay, D. (2009). *Veri Madenciliği ve Apriori Algoritması ile Süpermarket Analizi*. (Yayınlanmamış Yüksek Lisans Tezi). Sakarya: Sakarya Üniversitesi Fen Bilimleri Enstitüsü.

Balaji, S. ve Srivatsa, S.K. (2012). Naive Bayes Classification Approach for Mining Life Insurance Databases for Effective Prediction of Customer Preferences over Life Insurance Products. *International Journal of Computer Applications*. 51(3): 22-26.

Berry, M.J.A. ve Linoff, G.S. (2004). *Data Mining Techniques: for Marketing, Sales and Customer Relationship Management*. Indiana: Wiley.

Bhargavi, P. ve Jiyothi, S. (2009). Applying Data Mining Techniques for Classification of Agricultural Land Soils. *International Journal of Computer Science and Network Security*. 9(8): 117-122.

Coşkun, C. (2010). *Veri Madenciliği Algoritmaları Karşılaştırılması*. (Yayınlanmamış Yüksek Lisans Tezi). Diyarbakır: Dicle Üniversitesi Fen Bilimleri Enstitüsü.

Çatal, Ç., Sevim, U. ve Diri, B. (2011). Practical Development of an Eclipse-based Software Fault Prediction Tool Using Naive Bayes Algorithm. *Expert Systems with Applications*. 38(2011): 2347-2353.

Çavuşlar, İ. (2008). *Medical Data Mining Using Matlab/Sas*. (Unpublished Master Dissertation). İzmir: Graduate School of Natural and Applied Sciences.

Çelik, M. (2009). *Veri Madenciliğinde Kullanılan Sınıflandırma Yöntemleri ve Bir Uygulama*. (Yayınlanmamış Yüksek Lisans Tezi). İstanbul: İstanbul Üniversitesi Sosyal Bilimler Enstitüsü.

Dondurmacı, G. (2011). *Veri Madenciliğinde Regresyon Ağaçları ile Sınıflandırma ve Bir Uygulama*. (Yayınlanmamış Doktora Tezi). İstanbul: Mimar Sinan Güzel Sanatlar Üniversitesi Fen Bilimleri Enstitüsü.

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

Duda, R.O., Hart, P.E. ve Stork D.G. (2000). *Pattern Classification*. New York: Wiley-Interscience.

Edelstein, A.H. (1999). *Introduction to Data Mining and Knowledge Discovery*. Potomac: Two Crops Corporation.

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

Erdoğan, Ş.Z. (2004). *Veri Madenciliği ve Veri Madenciliğinde kullanılan K-Means Algoritmasının Öğrenci Veri Tabanında Uygulanılması*. (Yayınlanmamış Yüksek Lisans Tezi). İstanbul: İstanbul Üniversitesi Sosyal Bilimler Enstitüsü.

France, T., Yen, D., Wang, J.C. ve Chang, C.M. (2002). Integrating Search Engines With Data Mining For Customer-Oriented Information Search. *Information Management & Computer Security*. 10(5): 242-254.

Frawley, W.J., Piatetsky-Shapiro, G. ve Matheus, C.J. (1992). Knowledge Discovery in Databases: An Overview. *AI Magazine*. 13(3): 57-70.

Gargano, M.L. ve Raggad, B.G. (1999). Data Mining- A Powerful Information Creating Tool. *OCLC Systems & Services*. 15(2): 81-90.

Gemici, B. (2012). *Veri Madenciliği ve Bir Uygulaması*. (Yayınlanmamış Yüksek Lisans Tezi). İzmir: Dokuz Eylül Üniversitesi Sosyal Bilimler Enstitüsü.

Giudici, P. (2003). *Applied Data Mining: Statistical Methods for Business and Industry*. Chichester: Wiley.

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

Han, J. ve Kamber, M. (2006). *Data Mining: Concepts and Techniques*. San Francisco: Morgan Kaufmann.

Inmon, W.H. (2002). *Building The Data Warehouse*. New York: Wiley.

Kantardzic, M. (2011). *Data Mining: Concepts, Models, Methods and Algorithms*. New Jersey: Wiley.

Kıvrak, M. (2010). *Veri Madenciliği Yöntemleri ve Bir Uygulama*. (Yayınlanmamış Yüksek Lisans Tezi). Van: Yüzüncü Yıl Üniversitesi Sağlık Bilimleri Enstitüsü.

Kocabaş, F.M. (2010). *Veri Madenciliği Süreci ve Gerçek Bir Veri Seti Üzerinde Uygulanması*. (Yayınlanmamış Yüksek Lisans Tezi). Ankara: Hacettepe Üniversitesi Fen Bilimleri Enstitüsü.

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

Larose, D.T. (2005). *Discovering Knowledge In Data: An Introduction to Data Mining*. New Jersey: Wiley-Interscience.

Ling, C.X., Huang, J. ve Zhang, H. (2003). AUC: A Better Measure Than Accuracy in Comparing Learning Algorithms. *Advances in Artificial Intelligence* (ss.329-341). Düzenleyen Yang Xiang ve Brahim Chaib-Draa. Canada. 11-13 Haziran 2003.

Ma, C., Chou, D.C. ve Yen, D.C. (2000). Data Warehousing, Technology Assessment and Management. *Industrial Management & Data Systems*. 100(3): 125-134.

Maçkolik, (2017). <http://www.mackolik.com/> (13.07.2017)

Maimon, O. ve Rokach, L. (2010). Introduction to Knowledge Discovery and Data Mining. *Data Mining and Knowledge Discovery Handbook* (ss.1-15). New York: Springer.

Oğuzlar, A. (2003). Veri Ön İşleme. *Erciyes Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*. 17(21): 67-76.

Oğuzlar, A. (2004). CART Analizi ile Hane Halkı İşgücü Anketi Sonuçlarının Özetlenmesi. *Uludağ Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*. 18(3): 79-90.

Osmanbegovic, E. ve Suljic, M. (2012). Data Mining Approach for Predicting Student Performance. *Economic Review- Journal of Economics and Business*. 10(1): 3-12.

Özçınar, H. (2006). *KPSS Sonuçlarının Veri Madenciliği Yöntemleriyle Tahmin Edilmesi*. (Yayınlanmamış Yüksek Lisans Tezi). Denizli: Pamukkale Üniversitesi Fen Bilimleri Enstitüsü.

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

Panda, M. ve Patra, M.R. (2007). Network Intrusion Detection Using Naive Bayes. *International Journal of Computer Science and Network Security*. 7(12): 258-263.

Quinlan, J.R. (1986). Induction of Decision Trees. *Kluwer Academic Publishers*. 1(1): 81-106.

Quinlan, J.R. (1993). *C4.5: Programs for Machine Learning*. California: Morgan Kaufmann.

Rahangdale, G., Ahirwar, M. ve Motwani, M. (2016). Application of K-NN and Naive Bayes Algorithm in Banking and Insurance Domain. *International Journal of Computer Science Issues*. 13(5): 69-75.

Rajalakshmi, R. ve Aravindan, C. (2011). Naive Bayes Approach for Website Classification. *Information Technology and Mobile Communication* (ss.323-326). Düzenleyen Vinu V Das, Gylson Thomas ve Ford Lumban Gaol. India: 21-22 Nisan 2011.

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

Sumathi, S. ve Sivanandam, S.N. (2006). *Introduction to Data Mining and Its Applications*. Berlin: Springer.

Şentürk, Z.K. (2011). *Veri Madenciliği ile Kanser Tanısı*. (Yayınlanmamış Yüksek Lisans Tezi). Düzce: Düzce Üniversitesi Fen Bilimleri Enstitüsü.

Şık, M.Ş. (2014). *Veri Madenciliği ve Kanser Erken Teşhisinde Kullanımı*. (Yayınlanmamış Yüksek Lisans Tezi). Malatya: İnönü Üniversitesi Sosyal Bilimler Enstitüsü.

Ting, S.L., Ip, W.H. ve Tsang, A.H.C. (2011). Is Naive Bayes a Good Classifier for Document Classification?. *International Journal of Software Engineering and Its Applications*. 5(3): 37-46.

Tüzüntürk, S. (2010). Veri Madenciliği ve İstatistik. *Uludağ Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*. 29(1): 65-90.

Vembandasamy, K., Sasipriya, R. ve Deepa, E. (2015). Heart Diseases Detection Using Naive Bayes Algorithm. *International Journal of Innovative Science, Engineering & Technology*. 2(9): 441-444.

Vercellis, C. (2009). *Business Intelligence: Data Mining and Optimization for Decision Making*. Chichester: Wiley.

Witten, I.H., Frank, E. ve Hall, M.A. (2011). *Data Mining: Practical Machine Learning Tools and Techniques*. New Jersey: Morgan Kaufmann.

Wu, X. ve Kumar, V. (2009). *Top Ten Algorithms in Data Mining*. Florida: CRC Press.

Zhang, H. (2004). The Optimality of Naive Bayes. *FLAIRS Conference* (ss.562-567). Düzenleyen Valerie Barr ve Zdravko Markov. Florida. Ocak 2004.