T.C.
PAMUKKALE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

VERİ MADENCİLİĞİ YÖNTEMLERİ KULLANARAK MESLEK YÜKSEK OKULU ÖĞRENCİLERİNİN AKADEMİK BAŞARI TAHMİNİ

YÜKSEK LİSANS TEZİ

BURAK AYDEMİR

DENİZLİ, EKİM - 2017

T.C.
PAMUKKALE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

VERİ MADENCİLİĞİ YÖNTEMLERİ KULLANARAK MESLEK YÜKSEK OKULU ÖĞRENCİLERİNİN AKADEMİK BAŞARI TAHMİNİ

YÜKSEK LİSANS TEZİ

BURAK AYDEMİR

DENİZLİ, EKİM - 2017

KABUL VE ONAY SAYFASI

Burak AYDEMİR tarafından hazırlanan **"Veri Madenciliği Yöntemleri Kullanarak Meslek Yüksek Okulu Öğrencilerinin Akademik Başarı Tahmini"** adlı tez çalışmasının savunma sınavı 27.10.2017 tarihinde yapılmış olup aşağıda verilen jüri tarafından oy birliği / oy çokluğu ile Pamukkale Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Ana bilim dalı olarak kabul edilmiştir.

Jüri Üyeleri	İmza
Danışman Prof.Dr. Sezai TOKAT	.....
Üye Yrd.Doç.Dr. Elif HAYTAOĞLU	.....
Üye Yrd.Doç.Dr. Mahmut SİNECEN	.....

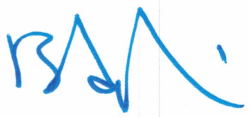
Pamukkale Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 06/12/2017 tarih ve ..48/07.. sayılı kararıyla onaylanmıştır.

.....

Prof. Dr. Uğur YÜCEL

Fen Bilimleri Enstitüsü Müdürü

Bu tezin tasarımı, hazırlanması, yürütülmesi, araştırmalarının yapılması ve bulgularının analizlerinde bilimsel etiğe ve akademik kurallara özenle riayet edildiğini; bu çalışmanın doğrudan birincil ürünü olmayan bulguların, verilerin ve materyallerin bilimsel etiğe uygun olarak kaynak gösterildiğini ve alıntı yapılan çalışmalara atfedildiğine beyan ederim.



Burak AYDEMİR

ÖZET

VERİ MADENCİLİĞİ YÖNTEMLERİ KULLANARAK MESLEK YÜKSEK OKULU ÖĞRENCİLERİNİN AKADEMİK BAŞARIM TAHMİNİ

YÜKSEK LİSANS TEZİ

BURAK AYDEMİR

PAMUKKALE ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

(TEZ DANIŞMANI:PROF. DR. SEZAİ TOKAT)

DENİZLİ, EKİM - 2017

Teknolojinin gelişmesiyle birlikte bilginin önemi her geçen gün artmaktadır. Veri madenciliği yöntemleri ile veriler üzerinde çok farklı analizler yapılabilmektedir. Bu araştırmanın amacı, veri madenciliği yöntemini kullanarak Meslek Yüksek Okulu öğrencilerinin akademik başarımlarını tahmin etmektir. Bu amaç doğrultusunda veri madenciliği sınıflama algoritmaları kullanılarak öğrencilerin başarılarını tahmin etmede en iyi başarımı gösteren sınıflama algoritması seçilmeye çalışılmıştır. Pamukkale Üniversitesi Meslek Yüksek Okullarına 2009 ile 2010 yılları arasında kayıt yaptıran 1387 öğrencinin bilgileri kullanılmıştır. Öğrencilerin akademik başarımlarını tahmin etmek için bağımlı değişken olarak öğrencilerin akademik not ortalamaları ile mezuniyet yılları kullanılmıştır. Akademik not ortalamasına göre başarım tahmininde en iyi sonucu Sıralı Minimum Optimizasyon(SMO) algoritması vermektedir. Mezuniyet yılına göre başarım tahmini yaptığımızda en iyi sonucu J4.8 ve NaiveBayes algoritmaları vermektedir.

ANAHTAR KELİMELER: Veri Madenciliği, akademik başarım, yapay sinir ağları, NaiveBayes, J4.8, SMO, IBk

ABSTRACT

PREDICTING ACADEMIC SUCCESS OF VOCATIONAL HIGH SCHOOL STUDENTS USING DATA MINING METHODS
GRADUATE THESİS
BURAK AYDEMİR
PAMUKKALE UNIVERSITY INSTITUTE OF SCIENCE
COMPUTER ENGINEERING

(SUPERVISOR:PROF.DR. SEZAİ TOKAT)

DENİZLİ, OCTOBER 2017

The importance of information increases each day with the developments in technology. Several different analysis can be applied on the related information using data mining methods. The aim of this research is to predict the academic success of vocational high school students of Pamukkale Universiy by using data mining methods. For this purpose, several data mining classification algorithms are used and their results are compared to find best suited algorithm. The information of 1387 students, who had registered to Pamukkale University Vocational High School between 2009 and 2010 are used. GPA and graduation year of the students are used as a dependent variable in order to predict academic success. Linear Minimum Optimization algorithm gives the best results when GPA is used, whereas J4.8 and Naïve Bayes algorithms are best suited if graduation date is used as a dependent variable.

KEYWORDS: Data mining, academic performance, artificial neural network, NaiveBayes, J4.8, SMO, IBk

İÇİNDEKİLER

Sayfa

ŞEKİL LİSTESİ

Sayfa

TABLO LİSTESİ

Sayfa

SEMBOL LİSTESİ

ÖSYM	**:**	Ölçme, Seçme ve Yerleştirme Merkezi
YGS	**:**	Yükseköğretime Geçiş Sınavı
LYS	**:**	Lisans Yerleştirme Sınavı
ÖSS	**:**	Öğrenci Seçme Sınavı
ABNO	**:**	Akademik başarı not ortalaması
SQL	**:**	Structured Query Language
YSA	**:**	Yapay Sinir Ağları
DVM	**:**	Destek Vektör Makineleri
WEKA	**:**	Waikato Environment for Knowledge Analysis
SMO	**:**	Sequential minimal optimization
IBk	**:**	Instance –base k

ÖNSÖZ

Bu çalışmada Meslek Yüksek Okulu öğrencilerinin akademik geçmiş ve ailevi durum bilgileri kullanılarak veri madenciliği yönteminin sınıflandırma teknikleriyle akademik başarımı tahmin etmeye çalışılmıştır. Bu amaç doğrultusunda veri madenciliği yöntemlerinden sınıflama algoritmaları kullanarak başarılarını tahmin etme de en iyi performansı gösteren sınıflama algoritması seçilmeye çalışılmıştır.

Çalışmanın uygulama kısmında Pamukkale Üniversitesi Meslek Yüksek Okullarına 2009 ile 2012 yılları arasında kayıt yaptıran öğrencilerin akademik bilgileri ile üniversiteye kayıt esnasında uygulanan anket verileri kullanılmıştır. Bu bilgileri bize sağlayan Pamukkale Üniversitesi Bilgi İşlem Dairesi'ne teşekkürlerimi sunarım.

Bu tez çalışmasının yürütülmesinde ilgi ve desteğini esirgemeyen, bilgi ve yönlendirmeleriyle bana destek olan hocam sayın Prof. Dr. Sezai TOKAT'a teşekkürlerimi sunarım.

Yüksek lisans eğitim sürecimin he aşamasında yakın ilgilerini gördüğüm, bana destek olan değerli bölüm hocalarıma ve teşekkürü bir borç bilirim.

Hayatımın her anında yanımda olan, bana maddi ve manevi desteklerini esirgemeyen annem, babam ve kardeşime tüm kalbimle teşekkür ederim.

1. GİRİŞ

Bilgi teknolojilerinde meydana gelen hızlı değişim ve gelişim toplumları bilgi üretmeye yöneltmiştir. Bilgi üretebilen veya bilgiyi kullanabilen toplumlar, teknolojiyi geliştirmekte ve kullanmaktadırlar. Teknolojinin kullanılması bireyleri ve toplumları geliştirmekte, onları olaylar karşısında daha hazırlıklı hale getirmekte ve hayatı kolaylaştırmaktadır. Teknolojik değişimlere ayak uydurabilen toplumlar teknolojinin sağladığı yararları yaşamlarıyla bütünleştirerek, bu gelişeme ayak uyduramayan toplumların her zaman önüne geçmektedir (Gündüz ve Odabaşı 2004).

Bilgi çağında meydana gelen bu hızlı değişim dünyadaki tüm toplumları bu teknolojik gelişimlere ayak uydurmaya itmektedir. Bu gelişeme ayak uydurmak için toplumların bu gelişime hazırlıklı olması gerekir. Yani bilgiye nasıl erişebileceğini bilen, ulaştığı bilgiyi kullanabilen, gerektiğinde yeni bilgileri kendi üretebilen bir toplumun yetişmesi gerekmektedir. Bu gelişime ayak uydurabilecek toplum oluşması ise ancak o toplumun eğitim sistemi sayesinde gerçekleştirilebilir.

Eğitim, uzamanlar tarafından, bireyin davranışlarında yaşantısı yoluyla istendik ve kasıtlı olarak değişme meydana getirme süreci olarak tanımlanmaktadır (Şimşek 2012). Uzmanlar tarafından yapılan bu tanımlarda anlatılmak istenen belli bir program ve plan dâhilinde öğrencilere istenilen, arzu edilen davranışları kazandırmak ve öğrencilerin bu davranışları sergilemesini beklemektir. Öğrencilerdeki tüm bu değişimin gerçekleşmesi belli bir plan dâhilinde olması gerekir. Bu plana da Eğitim ve Öğretim Programı denir.

Eğitim programı, Milli Eğitim'in amaçlarının gerçekleşmesine yönelik tüm faaliyetlerinin bir eğitim kurumunda öğrencilere sağlanmasıdır. Öğretim programı ise "bir derste öğrencilerin ulaşacağı hedefleri, hedeflerin kapsadığı davranışları, davranışları kazandırmak üzere düzenlenecek eğitim durumlarını ve davranışların ne derece kazandırıldığını ortaya koyabilecek sınama durumlarını kapsayan, gelişmeye açık ve çok yönlü etkileşim içinde olan öğeler bütünüdür (Hotaman 2010). Eğitim ve Öğretim Programlarının temelini öğrenciler oluştururlar. Bu programlarda öğrencilerin davranış olarak ulaşması gereken hedefler yer alır. Bu hedefler

doğrultusunda da öğrencilerin bu hedeflere ulaşmasını sağlayan eğitim-öğretim etkinlikleri oluşturulur. Öğrencilerin eğitim-öğretim etkinlikleri sayesinde oluşturulan hedeflerin ne kadarını gerçekleştirdiğini ne kadarını gerçekleştiremediğini belirleyen ölçme ve değerlendirme etkinlikleri gerçekleştirilir. Ölçme ve değerlendirme etkinlikleri programların sorunları hakkında uzmanlara bilgi veren önemli bir bölümüdür.

Türkiye'de eğitim programı okulöncesi, ilköğretim, ortaöğretim ve yükseköğretim olarak 4 kademe olarak planlanmıştır. Ortaöğretim yükseköğretime geçmeden önceki son kademedir. Öğrenciler bu kademede ilgi ve yeteneklerine uygun olan alanları seçerler. Bu seçim öğrencilerin ileride hangi tür meslekleri seçeceğini gösteren önemli bir seçimdir. Bu nedenle de ortaöğretim yükseköğretime geçişte önemli bir kademedir. Ancak öğrencilerin ileride hangi mesleği yapacağına karar verdiği ve sahip olacağı meslekle ilgili yeterlilikleri kazanacağı kademe yükseköğretimdir. Yükseköğretimin amacı; "Öğrencileri ilgi, istidat ve kabiliyetleri ölçüsünde ve doğrultusunda yurdumuzun bilim politikasına ve toplumun yüksek seviyede ve çeşitli kademelerdeki insan gücü ihtiyaçlarına göre yetiştirmek"[Milli Eğitim Temel Kanunu 1973]. Öğrenciler ortaöğretimin sonunda Ölçme, Seçme ve Yerleştirme Merkezi (ÖSYM) tarafından sınava tabi tutulmaktadırlar. ÖSYM öğrencilerin düzeyini belirlemek ve öğrencileri uygun yükseköğretim programlarına yerleştirmek için Yükseköğretime Geçiş Sınavı(YGS) ve Lisans Yerleştirme Sınavı(LYS) uygulamaktadır. Öğrenciler bir yükseköğretim programına yerleşmek için bu sınavlarda başarılı olmak zorundadırlar.

Eğitimin, insan yetiştirmenin çok önemli olduğu bu çağda eğitim kurumlarında öğrencilerin derslerde gösterdikleri performans da önemli hale gelmektedir. Öğrencilerin derslerde gösterecekleri performansı arttırmak veya oluşabilecek kötü performansların önüne geçebilmek için ileriye yönelik yapılabilecek tahminler, rehberlik çalışmasını daha etkili kılacaktır. Bu amaçla veri madenciliği yöntemleri etkili bir şekilde kullanılabilir. Bu yöntemleri kullanan çok sayıda çalışma bulunmaktadır.

Bir insanın başarılı veya başarısız olmasını sağlayan birçok etkiden söz edilebilir. Ancak bu etkilerin kişinin başarılı olma olasılığını ne kadar etkilediği daha önemli bir konudur. Bir öğrencinin ileriye dönük başarılarını tahmin etmede bize en

çok yol gösterecek olan etmenler de başarılı olma olasılığını en çok etkileyen etmenlerdir. Bu doğrultuda yapılan çalışmalar şunu göstermiştir. Bir öğrencinin akademik başarısını tahmin etmede bize en çok yol gösteren etmenler öğrencinin akademik geçmişidir. Ankara Üniversitesi'nde yapılan bir çalışma da Ankara Üniversitesi'nin bazı fakülte ve lisans programlarında öğrenim gören 419 3.sınıf öğrencilerinin akademik başarılarını etkileyen faktörleri kullanarak öğrencilerin başarı durumlarına göre sınıflandırılmasında Yapay Sinir Ağları ve Lojistik regresyon yöntemleri kullanılmıştır. Lojistik regresyon analizi ve yapay sinir ağları analizinin öğrencilerin akademik başarısını en çok hangi değişkenlerin etkilediğine ilişkin yapılan karşılaştırmada "Ortaöğretim Mezuniyet Ortalaması, Mezun Olunan Lise ve Üniversiteye Giriş Puanı" ortak değişkenler olarak belirlenmiştir. Yapay sinir ağları analizi sonucu akademik başarının en önemli ilk belirleyicisi (%100) "Üniversiteye Giriş Puanı" olduğu görülmüştür (Çırak 2012).

Anadolu Üniversitesi Açık Öğretim Fakültesi'nde Bilgi Teknolojileri I Temelleri (BIL101U) dersini alan öğrencilerin final puanı Radyal Taban Fonksiyonu (RBF) ve Çok Katmanlı Perceptron (MLP) modeli kullanılarak öngörülmeye çalışılmıştır. 2014-2015 Güz döneminde BIL101U modülünün final sınav puanlarının tahmininde vize puanı, cinsiyet, milliyet, eğitim durumu, meslek okulu mezuniyeti, yabancı dil, engelli olup olmaması, mezuniyet derecesi, tercih sırası, doğum tarihi, yerleştirme puanı ve AÖF öğrencilerinin üniversite giriş sınavı puanları değişken olarak kullanılmıştır. Çok Katmanlı Perceptron için farklı parametrelerle 12 farklı ağ oluşturulmuş ve Radial Basis Fonksiyonu için farklı parametrelerle dört farklı ağ oluşturulmuş ve her biri için elde edilen sonuçlar karşılaştırılmış. Öğrencilerin nüfus bilgilerinin final puanlarını tahmin etmede çok önemli etkenler olduğu gözlenmemiştir. Nihai puanları tahmin etmede en önemli değişkenin vize puanları olduğu görülmüştür (Aybek ve Okur 2016).

Türkiye'de üniversiteye girişte yapılan sınavlarda yüksek puan alan ya da yüksek net yapan öğrencilerin girdikleri yükseköğretim programında da başarılı olacakları düşünülmektedir. Öğrencilerin üniversitede genel matematik dersindeki başarıları ile ÖSS başarıları arasındaki ilişkiyi araştıran bir çalışma da üniversite öğrencilerin genel matematik dersindeki başarıları ile ÖSS giriş puanları arasında pozitif bir ilişki olduğunu tespit etmişlerdir (Çetin ve Mahir 2006).

Başka bir çalışma da Atatürk Üniversitesi öğrencilerinin mezun oldukları lise türleri ve lise mezuniyet dereceleri ile kazandıkları fakülteler arasındaki ilişki, veri madenciliği teknikleri kullanılarak incelenmiştir. Yapılan çalışmada lise mezuniyet notları yüksek olan öğrencilerin daha çok Tıp, Diş ve Eczacılık gibi yükseköğretim giriş puanı yüksek olan yerleri tercih ettikleri ve bu bölümlere yerleştikleri görülmüştür (Ayık 2007).

Veri madenciliği yönteminin kullanıldığı bir çalışma da öğrencilerin üniversite giriş sınavındaki başarı durumlarını tahmin eden bir erken uyarı sisteminin geliştirilmesini amaçlanmıştır. Araştırmada üniversite giriş sınavında başarıyı etkileyen faktörlerin başında, öğrencilerin ortaöğretimdeki not bilgileri ve ilköğretim diploma not bilgisi olduğu görülmüştür. Özellikle öğrencilerin 11. ve 12. sınıf notlarının üniversite giriş sınavındaki başarılarında diğer notlarına göre daha önemli olduğu sonucuna varılmıştır (Göker 2012).

Meslek yüksek okuluna yeni kayıt olan öğrencilerin akademik başarılarını ve mezuniyet sürelerini yapay zeka tekniklerinden biri olan destek vektör makinelerini kullanarak tahmin etmeye çalışmışlar. Girdi verileri olarak öğrencilerin cinsiyeti, yaşı, öğrencinin geldiği coğrafi bölge, öğrencinin mezun olduğu lise türü, mezun olduğu liseden aldığı diploma notu ve meslek yüksek okuluna sınavla mı yoksa sınavsız mı giriş yaptığı bilgileri kullanılmıştır. Yapılan çalışma sonucunda öğrencinin akademik geçmişinin meslek yüksek okulundaki başarısını ve mezun olma süresin önemli derecede etkilediği görülmüştür. Ayrıca sınava girmeden meslek yüksek okuluna kayıt yaptıran öğrencilerin akademik başarısı ve mezun olma süresini olumsuz etkilediği gözlenmiştir (Tokat ve diğ. 2014).

Öğrencilerin akademik başarılarını tahmin etmek için birçok veri madenciliği yöntemine başvurulmuş. Buradaki amaç her zaman en iyi tahmin sonuçlarını veren yöntemi bulmak olmuş. Bu doğrultu da yapılan bazı çalışmalar aşağıdaki gibidir;

Hindistan'da yapılan bir çalışmada, veri madenciliği yöntemi aracılığıyla İleri Orta Öğretim öğrencilerin akademik başarılarını etkileyen demokrafik, psikolojik ve sosyo ekonomik özellikleri kullanılarak öğrencilerin akademik başarıları tahmin edilmeye çalışılmış. Gerekli bilgiler anket aracılığıyla ve eğitim kurumlarından alınarak bir veri tabanı oluşturulmuş. Çalışma da Karar ağacı algoritması olan

CHAID algoritması kullanılarak bir model oluşturulmuş. Bu modelde öğrencilerin başarıları 7 sınıfa ayrılmıştır. Model uygulandığında öğrencilerin başarıları %44.69 oranında doğru olarak tahmin edilmiştir (Ramaswami ve Bhaskaran 2010).

Belçika'da yapılan bir çalışmada 533 üniversite birinci sınıf öğrencisini sınav sonuçlarına göre düşük riskli, orta riskli ve yüksek riskli grup olarak 3 gruba ayırmışlar. Düşük riskli grup öğrenciler; başarılı olma olasılığı yüksek öğrencilerden oluşmaktadır. Orta riskli grup öğrenciler; üniversite tarafından alınan önlemler sayesinde başarılı olabilecek öğrencilerden oluşmaktadır. Yüksek riskli grup öğrenciler; başarısız olma olasılığı yüksek veya okuldan ayrılma olasılığı olan öğrencilerden oluşmaktadır. Daha sonra öğrencilere uygulanan anket ve sınav sonuçlarına göre öğrencilerin hangi grupta yer alabileceğini tahmin etmeye çalışmışlar. Bunun için de karar ağacı algoritmaları olan ID3 ve CART algoritmaları ile yapay sinir ağları ve doğrusal diskriminant analizi kullanmışlar. Bu yöntemlerle %40.63 ile %57.35 oranları arasında doğru tahmin yüzdesine ulaşmışlardır (Superby ve diğ. 2006).

Başka bir çalışmada üniversite öğrencilerin dönem sonu başarılarını tahmin etmek ve bu doğrultuda öğrencilerin okuldan ayrılmalarını önlemek, öğrencilerin ihtiyaç duyduğu özel ilgiyi sağlamak ve öğrencilere gerekli tavsiyelerde bulunmak için dönem içinde yapılan sınavlar, öğrencilere verilen ödevler, öğrencilerin derslere devam süreleri, öğrencilerin laboratuar çalışmaları ve öğrencilerin önceki dönemlerde almış oldukları notlar kullanılmış. Öğrencilerin başarılı olmalarını sağlayacak gerekli kuralları çıkarmak için karar ağacı algoritmaları olan ID3, C4.5 ve CART algoritmaları kullanılmış ve bu algoritmalar karşılaştırılmış. Bu algoritmalarla öğrencilerin performans tahmininde %45 ile %56 arasında doğru sınıflandırma yüzdesine ulaşmışlardır. %56 ile en yüksek tahmin oranını CART algoritması gerçekleştirmiştir (Yadav ve diğ. 2011).

Lise öğrencileri arasında yavaş öğrenen öğrencilerin tespiti için yapılan bir çalışmada 152 öğrenciye uygulanan anketten elde edilen verilerden ilk önce öğrencilerin performans tahminini en çok etkileyen sekiz nitelik bulunmuş. Daha sonra sınıflandırma tekniklerinden olan; Multilayer Perception, Naïve Bayes, SMO, J48 and REPTree algoritmaları uygulanmış ve sonuçlar karşılaştırılmış.

Karşılaştırılan sonuçlara göre en iyi sonucu %75 doğru tahmin yüzdesiyle Multilayer Perception algortiması yani yapay sinir ağları tekniği vermiş (Kaura ve diğ. 2015).

Kolombiya'nın en büyük üniversitesinde öğrencilerin akademik statü kaybını yani öğrencinin üniversiteden atılması veya uzaklaşmasını tahmin etmek için bir çalışma yapılmış. Bu çalışma da öğrencilerin üniversiteye kabullerinin gerçekleşmesi için yapılan sınav sonuçları, öğrencilerin demografik bilgileri, sosyo-ekonomik durumları ve öğrencilerin akademik bilgileri kullanılarak öğrencilerin akademik statü kayıpları tahmin edilmiş. Bunun için veri madenciliğinin sınıflama metotlarından Naive Bayes ile Karar Ağaçları kullanılmış. Tahmin yapılırken öğrencilerin üniversitede yaptıkları dört kayıt dönemi değerlendirilmiş. Sadece öğrencilerin üniversiteye giriş sınav sonuçları ile değerlendirme yapıldığında iki algoritma da benzer sonuçlar vermiş. Öğrencilerin akademik bilgileri değerlendirmeye dahil edildiğinde bu dört dönemde de en iyi sonuçları Naive Bayes algoritması vermiş. Naive Bayes sonuçları test kümesi üzerinde daha iyi sonuçlar vermiş; Ancak, eğitim ve test verileri arasında farklılıklar varmış. Karar ağaçları sonuçları yeni verileri test ederken daha güvenilir ve daha tutarlı sonuçlar vermiş (Guarin 2015).

Öğrenci performansını tahmin etmede veri madenciliği tekniklerinin karşılaştırılmalı bir değerlendirilmesi yapılmış. Bu çalışma da veri madenciliği algoritmalarında kullanılacak veri olarak öğrencilerin akademik bilgileri ile ailevi özellikleri kullanılmış. Tahmin bilgisi olan mezuniyet puanı dört sınıfa ayrılmış. Öğrenci performansının tahmini için Karar Ağacı (J48) algoritması, Naive-Bayes algoritması, Random Forest algoritması, Classification and Regression Trees (CART) algortiması kullanılmış. Sonuçlar karşılaştırıldığında en iyi sonucu Random Forest algoritması vermiş. Doğru sınıflandırılmış örnek % 61.40'dır (Kumar ve Singh 2017).

Veri madenciliğinin sınıflandırma algoritmalarından olan J48 (Decision Tree),Random Forest ,Naive Bayes, Naive Bayes Multinomial, K-star, IBk algoritmalarını üniversite öğrencilerinin performans tahmininde kullanarak bir karşılaştırma çalışması yapmışlar. Bunun için 480 öğrencinin cinsiyet, uyruk, doğum yeri, ders notları, derse katılımıyla ilgili bilgilerden oluşan 16 parametrelik bir girdi verisi kullanıldı. Öğrencilerin performansı da alt düzey, orta ve üst düzey olarak sınıflandırıldı. Algoritmalar girdi verisine uygulanarak öğrenci performansı doğru

tahmin yüzdeleri karşılaştırıldığında tüm algoritmaların %67 ile %76 arasında bir oranda birbirlerine yakın sonuçlara ulaştığı görüldü. Ancak en iyi sonucu veren algoritmaların J48 ile Random Forest teknikleri olduğu görüldü (Kapur ve diğ. 2017).

Öğrencilerin sosyo-ekonomik ve nüfusa dayalı bilgilerin kullanılmadığı sadece ders performanslarının dikkate alındığı çalışmada öğrencilerin üniversiteye kabul için kullanılan puanlar ile üniversite birinci ve ikinci sınıf sonunda derslerden aldıkları notları kullanarak mezuniyet puanlarını tahmin etmeye çalışmışlar. Mezun puanı 5 aralığa bölünerek hesaplanmış ve tahmin için sınıflandırma algoritmalarından Decision Tree, Random Forest , Naive Bayes, Neural Network, Nearest Neighbour kullanılmış. Çıkan sonuçlar incelendiğinde en iyi performansa sahip sınıflandırıcının %83.65 doğru sınıflandırma yüzdesi ile Naive Bayes algoritması olduğu görülmüş. Bu algoritmaya en yakın sonucu veren algoritmanın %74.04 ile Nearest Neighbour olduğu görülmüş. Ancak bu algoritmalarda hangi niteliklerin tahmini etkilediğini anlamak mümkün olmamaktadır. Öğrencinin performansını etkileyen nitelikleri görmek açısından en iyi algoritmanın Karar Ağaçları olduğu sonucuna ulaşılmıştır (Asif ve diğ. 2017).

Öğrencilerin performanslarını tahmin etmek için kullanılan veri madenciliği teknikleri konusunda genel bir bakış sağlamak için sistematik bir literatür çalışması yapılmış. Yapılan bu çalışmadaki amaç öğrenci performansları analizlerinde kullanılan değişkenleri belirlemek ve öğrenci performans tahmininde kullanılan tahmin yöntemlerini incelemek. Bu amaçla 2002'den 2015 Yılına kadar IEEE Xplore, Springerlink, ScienceDirect, ACM dijital kütüphane veri tabanlarındaki yukarıda bahsedilen amaçlarla yapılmış dergi makaleleri, konferans bildirileri, atölye çalışmaları incelenmiş. İnceleme sonucunda bizimde yaptığımız çalışmaları destekleyecek şu sonuçlara ulaşılmış; genel olarak öğrenci performans tahmininde en iyi metotlar yapay sinir ağları ile karar ağaçlarıdır. Ancak analizlerin içinde öğrencilerin psikometrik faktörler ve ders dışı etkinlikler dahil edildiğinde en iyi sonuçları destek vektör makineleri vermektedir. Öğrencinin genel not ortalaması, öğrencinin demografik özellikleri, lise geçmişi, bir bursa sahip olması, sosyal iletişimi gibi faktörlerin hepsi kullanıldığında Naive Bayes metodu yapay sinir ağları

ve karar ağacı metotlarına göre daha yüksek tahmin yüzdesine sahip olduğu görülmüştür (Shahiria ve diğ. 2015).

Öğrencinin akademik başarım tahmini üzerine yapılan yukarıdaki çalışmalar incelendiğinde öğrencinin akademik performansını etkileyen en önemli etkinin öğrencinin akademik geçmişi olduğu görülmektedir. Öğrencinin akademik performansını etkileyebilecek ailevi durumları da göz önünde bulundurularak çalışmada veri madenciliği yöntemleri aracılığıyla Meslek Yüksek Okulu öğrencilerinin başarıları tahmin edilmeye çalışıldı. Bu amaç doğrultusunda veri madenciliği yöntemlerinden sınıflama algoritmaları kullanılarak öğrencilerin başarılarını tahmin etme de en iyi performansı gösteren sınıflama algoritması seçilmeye çalışılmıştır.

2. Veri Madenciliği

2.1 Veri Madenciliği Nedir?

Teknolojinin gelişmesi ile beraber bilginin önemi artmakta ve bilgiye olan ihtiyaç neticesinde milyonlarca veri üretilmekte ve saklanmaktadır. Bu kadar büyük çapta veriden anlamlı sonuçlar çıkarma ihtiyacı veri madenciliği (data mining) kavramını doğurmuştur. Gelişen teknoloji bu verilerin kolayca saklanabilmesini ve gerektiğinde erişilebilmesini hem kolaylaştırıyor hem de bu işlemlerin her geçen gün daha ucuza mal edilmesi sağlanıyor. Bu veri yığınlarından belirli bir amaç doğrultusunda anlamlı sonuçlar çıkarıp kararlar alabilmek için çeşitli veri madenciliği yöntemleri geliştirilmiştir.

Veri madenciliği; Büyük miktarda veri yığınının içinden değerli ve kullanılabilir bilgilerin açığa çıkarılması ve bu bilgiler üzerinden yönetimsel kararların alınması, gelecekle ilgili tahminler yapılmasını sağlayacak bağıntı ve kuralların bulunması sürecidir (Gökçen 2010). Veri Madenciliği geniş anlamda veri analiz teknikleri bütünüdür. Tek başına bir çözüm değildir. Mevcut problemleri çözmek, kritik kararlar almak ve geleceğe yönelik tahminlerde bulunmak için gerekli olan bilgileri ortaya çıkaran bir araçtır. Ortaya çıkarılan bilgiler çok net olmayan, keşfedilmemiş ama potansiyeli olan kullanışlı ve anlamlı bilgilerdir.

Veri madenciliği, büyük boyutlu veri ambarlarının meydana çıkmasının bir sonucudur. 1960'larda veriler elektronik ortamda toplanmaya ve geçmiş veriler bilgisayarlar ile analiz edilmeye başlanmıştır. 1980'lerde bağıntılı (relational) veritabanları ve SQL ile verilerin dinamik ve anlık analiz edilmesine olanak sağlanmıştır. 1990'lara gelindiğinde toplanmakta olan verinin hacmi çok büyük boyutlara ulaşmış ve verilerin depolanması için veri ambarları kullanılmaya başlanmıştır. Veri madenciliği toplanan bu büyük veri kütlelerinin değerlendirilmesi için istatistik ve yapay zekâ tekniklerinin kullanılması sonucunda ortaya çıkmıştır. Teknolojik gelişmeler, ham verilerin yeni fırsatlar üretmek üzere yönetim ve pazar ihtiyaçlarına yanıt verecek bilgiye dönüştürülmesini kolaylaştırmış ve bir anlamda

kurumları veri madenciliği üzerinde çalışmaya mecbur bırakmıştır (Ergüden ve Erşahin 2008).

Veri Madenciliğini tanımlayan diğer yaklaşımlara bakacak olursak; Veri madenciliği, çok büyük miktardaki gözlenebilir verinin analiz edilmesiyle, beklenmedik veri ilişkilerinin ve sıra dışı sonuçların veri sahibine anlaşılır bir şekilde iletilmesidir (Gülçe 2010).

Başka bir tanım şöyledir; büyük veri tabanlarından güvenilir, geçerli ve kullanılabilir bilgi çıkarma sürecidir. Yani o büyük veri tabanlarından işimize yarayacak kararlarımızda bize yardımcı olacak bilgiyi keşfetme sürecidir (Paul ve diğ. 2002).

2.2 Veri Madenciliğinin Uygulama Alanları

Veri Madenciliği yöntemini günümüzde karar verme sürecine ihtiyaç duyulan birçok alanda uygulamak mümkündür. Bunlar aşağıdaki gibi özetlenmiştir (Ünsal 2011);

2.2.1 Pazarlama Alanında

- Müşterilerin satın alma örüntülerinin belirlenmesi
- Müşterilerin demografik özellikleri arasındaki bağlantıların bulunması
- Posta kampanyalarında cevap verme oranının artırılması
- Mevcut müşterilerin elde tutulması, yeni müşterilerin kazanılması
- Pazar sepeti analizi
- Müşteri ilişkileri yönetimi
- Müşteri değerlendirme
- Satış tahmini

2.2.2 Bankacılık Alanında

- Farklı finansal göstergeler arasında gizli korelâsyonların bulunması
- Kredi kartı dolandırıcılıklarının tespiti
- Kredi kartı harcamalarına göre müşteri gruplarının belirlenmesi
- Kredi taleplerinin değerlendirilmesi
- Risk analizleri

2.2.3 Sigortacılık Alanında

- Yeni poliçe talep edecek müşterilerin tahmin edilmesi
- Sigorta dolandırıcılıklarının tespiti
- Riskli müşteri örüntülerinin belirlenmesi

2.2.4 Savunma Sistemleri Alanında

- Terörist ve düşman eylemlerinin modellenmesi ve kestirimi
- Uçak kazalarında hataların saptanması ve önlemlerin alınması

2.2.5 Borsa Alanında

- Hisse senedi fiyat tahmini
- Genel piyasa analizleri
- Alım-satım stratejilerinin optimizasyonu

2.2.6 Telekomünikasyon Alanında

- Kalite ve iyileştirme analizleri
- Abonelik tespitleri
- Hatların yoğunluk tahminleri

2.2.7 Sağlık ve İlaç Alanında

- Test sonuçlarının tahmini
- Ürün geliştirme
- Tıbbi teşhis
- Tedavi sürecinin belirlenmesi
- Yerleşim yerlerine göre hastalık haritalarının çıkarılması

2.2.8 Endüstri Alanında

- Kalite kontrol analizleri
- Lojistik
- Üretim süreçlerinin optimizasyonu

2.2.9 Eğitim Alanında

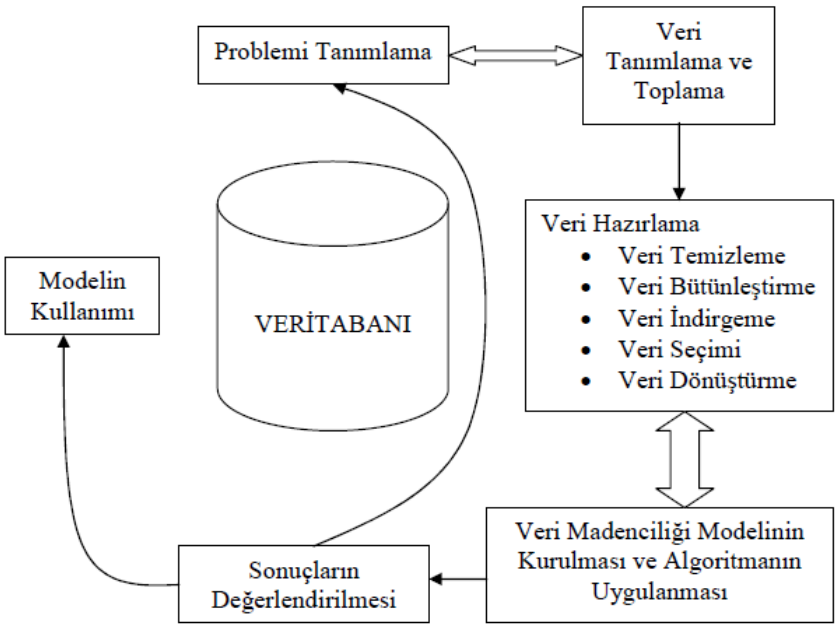
- Ölçme ve değerlendirme çalışmaları
- Mesleki rehberlik faaliyetleri
- Sınav başarısı analizleri

2.3 Veri Madenciliği Süreci

Veri madenciliği pek çok alanda kullanılan bir yöntemdir. Ancak bu yöntem bazı aşamalardan meydan gelmektedir. Kullanılan veri madenciliği yönteminin doğru sonuçlar vermesi önemlidir. Doğru sonuçlara ulaşabilmek içinde veri madenciliği sürecindeki aşamaları doğru olarak yerine getirmek gerekir. Şekil 2.1'de veri madenciliği süreci gösterilmiştir.

2.3.1 Problemin Tanımlanması

Sürecin ilk adımı veri madenciliği çalışmasının hangi amaç için yapılacağının net olarak tanımlanmasıdır. Bu aşamada ihtiyaçlar ve süreç sonunda elde edilecek bilginin hangi amaçla kullanılacağı belirlenmiş olur. Ayrıca bu aşamada çalışmanın süreci de planlanmalıdır.



Şekil 2.1: Veri Madenciliği Süreci (Ünsal 2011)

Çalışma için gerekli olacak veriler neler olduğu, maliyetin ne olacağı, karşılaşılabilecek risklerin neler olacağı değerlendirilmelidir. Değerlendirme uygun bir şekilde yapılmazsa yapılan çalışma sorunu çözmeyeceği gibi baksa sorunların ortaya çıkmasına da neden olabilir. Bu şekilde yapılmış bir veri madenciliği uygulaması hedefine ulaşmaktan çok uzaktır.

2.3.2 Veri Tanımlama ve Toplama

Bu aşamada verilerin ve verilerin hangi kaynaktan alınacağı belirlenir. Ayrıca toplanan verinin amaca uygun olup olmadı belirlenir. Veri yapısı daha iyi anlaşıldıkça problem tanımı değiştirilebilir veya yeniden yapılabilir. Veriler

toplanırken kurumun kendi verileri dışında belirlenen amaca uygun olarak başka kuruluşların veritabanlarından da faydalanılabilir.

2.3.3 Veri Hazırlama

Amaca uygun olarak toplanan verinin uygulanacak olan veri madenciliği modeline uygun hale dönüştürülmesi aşamasıdır. Modelleme aşamasının sağlıklı sonuç vermesi bu aşamada hazırlanan verilere bağlıdır. Veri madenciliği sürecinde verilerin modele bağlı olarak yeniden düzenlenme ihtiyacı doğarsa veri hazırlama aşaması tekrarlanabilmektedir.

Veri hazırlama sürecinde yapılan işlemler aşağıdaki gibi sıralanmıştır (Göker 2012);

- Veri Temizleme
- Veri Birleştirme
- Veri Dönüştürme
- Veri İndirgeme

2.3.3.1 Veri Temizleme

Çeşitli kaynaklardan elde edilen veriler istenilen özelliklere sahip olmayabilir. Bu verilerin içinde eksik veya hatalı verilerle karşılaşabiliriz. Veritabanlarında yer alan bu tür verilere gürültü veriler denir. Bu tür gürültü veriler analizlerden doğru sonuçlar elde etmemizi engellerler. Analizlerden doğru sonuçlar elde edebilmek için bu tür verilerin düzeltilmesi veya silinmesi gerekir. Verilerin düzeltilmesinde kullanılabilecek teknikler aşağıda sıralanmıştır (Taşdemir 2012);

- Eksik değer içeren kayıt veya kayıtlar atılabilir. Bu metot genellikle sınıf etiketi eksik olduğu durumda yapılır. Bu metot satır birden fazla özellik eksik veri içermediği sürece verimli değildir.
- Eksik veri manüel olarak tamamlanabilir. Bu metot zaman alıcı bir yöntemdir ve büyük veri setlerinde uygulanabilir değildir.

- Eksik veri genel bir sabit ile doldurulur. Bütün eksik veriler "Bilinmiyor", "∞" gibi aynı sabitle doldurulur. Bu yöntemde Veri Madenciliği yazılımı verilerin hepsinin ortak "Bilinmiyor" verisini içerdiği sonucunu çıkarabilir.
- Değişkenin tüm verileri kullanılarak ortalaması hesaplanır ve eksik değer yerine bu değer kullanılabilir.
- Değişkenin tüm verileri yerine, sadece bir sınıfa ait örneklerin değişken ortalaması hesaplanarak eksik değer yerine kullanılabilir.
- Verilere uygun bir tahmin yapılarak, örneğin regresyon ya da karar ağacı modeli kurularak eksik değer tahmin edilebilir ve eksik değer yerine kullanılabilir.

2.3.3.2 Veri Birleştirme

Veri bütünleştirme işlemi, veri tabanlarında, çeşitli kaynaklardan elde edilen verinin birleştirilmesidir. Tabi farklı veri tabanlarından gelen verilerin tek bir veri tabanında birleştirilmesi esnasında şema birleştirme hataları oluşabilir. Örnek vermek gerekirse, bir veri tabanında cinsiyetle ilgili girişler simgeler şeklinde "E" ve "K" kodlarıyla belirtilmiş olabilir. Burada "E" kodu erkek, "K" kodu ise kadınları simgelemektedir. Başak bir veri tabanında ise cinsiyetle ilgili alan 1 veya 0 (sıfır) değerleriyle ifade edilmiş olabilir. Farklı bir veri tabanında direkt olarak "Erkek" ve "Kadın" ifadeleri kullanılmış olabilir. Bu tip aynı veri alanı için farklı veri tabanlarında farklı simgelemeler kullanılmış olabilir. Farklı veri tabanlarında alınıp birleştirilen bu tür veriler üzerinde analiz yapmak imkânsız hale gelir. Bu nedenle bu tip verilerin analiz aşamasından önce ortak bir türe dönüştürülmesi yani veri bütünleştirmesi yapılması gerekir.

2.3.3.3 Veri İndirgeme

Veri madenciliğinde çözümleme işlemleri bazen çok uzun süre alabilir. Veri kümesinde aynı tipte çok kayıt olduğu biliniyor ve bu kayıtlarının bazılarının çıkarılması sonucu değiştirmeyeceği düşünülüyorsa, kaynak verilerin sayısı

azaltılabilir. Örneğin kayıtları tanımlamada kullanılan kimlik numarası, okul numarası, kayıt tarihi, isim vb. bilgiler model için hazırlanan veri kümesinden çıkartılabilir. Veri indirgeme yapılırken veri küpü oluşturma, boyut indirgeme, veri sıkıştırma, örnekleme ve genelleme teknikleri kullanılabilir (Ünsal 2011).

2.3.3.4 Veri Dönüştürme

Veri Madenciliğinde bazı zamanlar verileri aynen işleme katmak kurulan sistem için uygun olmayabilir. Bazı değişkenlerin ortalaması ve varyansları, diğer değişkenlerden çok büyük veya çok küçük olması durumunda, bu büyük fark yaratan değişkenlerin diğerleri üzerinde analiz aşamasında etkisi daha çok olur ve onların rollerini önemli ölçüde azaltır. Ayrıca değişkenlerin sahip olduğu çok büyük ve çok küçük değerler de çözümlemenin sağlıklı bir şekilde yapılmasını engeller. Bu durumda verinin standartlaşması için Min-Max normalleştirme veya Z-score standartlaştırma yöntemleri kullanılabilir.

Verileri 0 ile 1 arasındaki sayısal değerlere dönüştürmek için min-max normalleştirme yöntemi uygulanır. Bu yöntem, veri içindeki en büyük ve en küçük sayısal değerin belirlenerek diğerleri buna uygun biçimde dönüştürme esasına dayanmaktadır. Söz konusu dönüştürme yapısı denklem (2.1)'de ifade edilmektedir:

$$A' = \frac{A - A_{min}}{A_{max} - A_{min}} \tag{2.1}$$

Bu formülde A gözlenen, Amin en küçük gözlenen, Amax en büyük gözlenen ve A' ise dönüştürme sonucunda elde edilen değeri temsil etmektedir (Ünsal 2011).

Dönüştürme yapılırken kullanılan bir diğer yöntem ise Z-score standartlaştırmadır. Bu yöntem, verilerin ortalaması ve standart hatası göz önüne alınarak yeni değerlere dönüştürülmesi esasına dayanmaktadır. Söz konusu dönüştürme yapısı denklem (2.2)'de ifade edilmektedir:

$$B' = \frac{B - \bar{B}}{\sigma_B} \quad (2.2)$$

Bu formülde B gözlenen, $\overline{B}$ gözlenen değerlerin aritmetik ortalaması ve σ_B ise gözlenen değerlerin standart sapmasını temsil etmektedir (Ünsal 2011).

2.3.4 Veri Madenciliği Modelinin Kurulması ve Algoritmanın Uygulanması

Veri madenciliği yöntemlerini uygulayabilmek için yukarıda sıralanan işlemlerin uygun görünenleri yapılır. Veri hazır hale getirildikten sonra konuyla ilgili veri madenciliği algoritmaları uygulanır. Söz konusu algoritmalar sınıflandırma, kümeleme ve birliktelik kuralları konusunda olacaktır.

2.3.5 Sonuçların Değerlendirilmesi

Veri madenciliği modeli uygulanması ile elde edilen sonuçlar değerlendirilerek kurulan modelin kullanılmaya geçilip geçilmeyeceğine karar verilir. Sonuçların başlangıçta belirlenen hedeflere uygun olmadığı görülürse problem tanımlama aşamasına dönülebilir.

2.4 Veri Madenciliği Modelleri

Veri madenciliği sürecinin iki temel amaca hizmet etmektedir. Bunlardan birincisi mevcut veritabanından verileri analiz ederek tahminler yapmak (tahmin edici model), ikincisi ise veriler arasındaki ilişkilerden davranışlar tanımlamak (tanımlayıcı model). Tahmin edici modellerde, sonuçları bilinen verilerden hareket edilerek bir model oluşturulur ve bu modelden yararlanılarak sonuçları bilinmeyen veri kümelerini için sonuç değerleri tahmin edilmeye çalışılır. Tanımlayıcı modellerde ise karar vermeye yardımcı olarak kullanılabilecek mevcut veriler arasındaki örüntüler tanımlanmaya çalışılır (Akın 2008).

Tahminleyici modeller örneğin, bir bankanın önceki dönemlerde müşterilerine verdiği kredilerden hareketle müşteri özellikleri ile dönen ve dönmeyen krediler arasında bir model oluşturarak daha sonraki dönemlerde müşteri özelliklerine göre verilecek olan kredinin dönüp dönmeyeceğini tahmin edebilir. Tanımlayıcı bir model ise daha çok veriler arasında gizli kalmış ilişkiyi ortaya çıkarırlar ve şöyle bir sonuç elde edebilirler: geliri X-Y aralığında ve iki veya daha fazla arabası olan çocuklu aileler ile geliri X-Y aralığından daha düşük ve çocuğu olmayan ailelerin satına alma güçlerinin birbirine benzerlik gösterdiğini söyleyebilir (Üçgün 2009).

Veri madenciliği modelleri işlevlerine göre 3 temel grupta toplanır (Taşdemir 2012).

- Sınıflandırma(Classification)
- Kümeleme (Clustering)
- Birliktelik kuralları ve sıralı örüntüler (Association rules and sequential patterns)

2.4.1 Sınıflandırma(Classification)

Verinin içerdiği ortak özelliklere göre ayrıştırılması işlemi sınıflandırma olarak adlandırılır. Sınıf olmak için her verinin sınıf içinde yer alan diğer verilerle belirlenmiş bir ortak özelliği olması gerekir (Bırtıl 2011).

Sınıflama en çok bilinen veri madenciliği yöntemlerinden biridir. Örüntü tanıma, hastalık tanıları, dolandırıcılık tespiti, kalite kontrol çalışmaları, pazarlama konuları, bankacılık sektörü sınıflandırma tekniklerinin kullanıldığı alanlardır. Verilerin sınıflandırılması için belirli bir süreç izlenir. Öncelikle var olan veri tabanın bir kısmı eğitim amacıyla kullanılarak sınıflandırma kurallarının oluşturulması sağlanır. Böylelikle geçmiş verinin hangi sınıflara ait olduğu belirlenir. Daha sonra oluşturulan kurallar yardımıyla yeni bir durumla karşılaşıldığında gelen yeni verinin hangi sınıfa dâhil olduğu bulunur (Göker 2012).

Sınıflandırma yöntemiyle ilgili örnek bir model şu şekildedir: satışlarını arttırmak için kampanya düzenlemek isteyen bir firma önceden satış yapmış olduğu müşterilerinin verilerini kullanarak kampanyasına katılma ihtimali olan potansiyel alıcıları belirleyebilir ve kampanyasını bu doğrultuda oluşturur (Bırtıl 2011).

Sınıflama modelinde kullanılan başlıca yöntemler şunlardır:

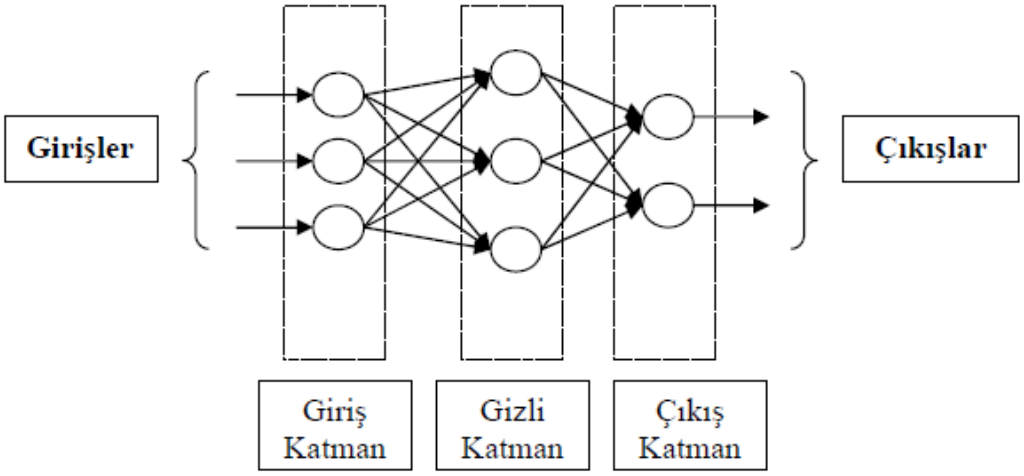
- Karar Ağaçları (Decision Trees)
- Bayes Sınıflandırması
- K-En Yakın Komşu (K-Nearest Neighbor)
- Yapay Sinir Ağları (Artificial Neural Networks)
- Genetik Algoritmalar (Genetic Algorithms)
- Destek Vektör Makineleri

2.4.1.1 Yapay Sinir Ağları

Yapay sinir ağları (YSA) teknolojisi insanlığın doğayı araştırma ve taklit etme çabalarının bir sonucu olarak ortaya çıkmıştır. YSA, basit biyolojik sinir sisteminin çalışma şekli simüle edilerek tasarlanan bir programlama yaklaşımıdır. Biyolojik sistemlerde öğrenme, nöronlar arasındaki sinaptik bağlantıların oluşması ile olur. İnsanlar doğumlarından itibaren yaşayarak öğrenme sürecine içerisine girerler. Bu süreç içerisinde beyin sürekli bir gelişim göstermektedir. İnsanlar yaşayıp tecrübe ettikçe sinaptik bağlantılar ayarlanır ve hatta yeni sinaptik bağlantılar oluşur. Bu sayede öğrenme gerçekleşir. Bu durum YSA için de geçerlidir. YSA'lar simüle edilen sinir hücreleri (nöronlar) içerirler ve bu nöronlar çeşitli şekillerde birbirlerine bağlanarak ağı oluştururlar. Bu ağlar öğrenme, hafızaya alma ve veriler arasındaki ilişkileri ortaya çıkarmaya kapasitesine sahiptirler. Yani YSA'lar normalde bir insanın düşünme ve gözlemlemeye yönelik doğal yeteneklerini gerektiren problemlere çözüm üretmektedir (Şanlı 2008).

YSA'larda öğrenme örnekler kullanılarak eğitme yoluyla olur. Yani nöronlara giren ve çıkan verilerin eğitme algoritması tarafından kullanılarak nöronlar arasındaki bağlantı ağırlıklarını bir yakınsama sağlanana kadar, tekrar tekrar ayarlamasıyla oluşur.

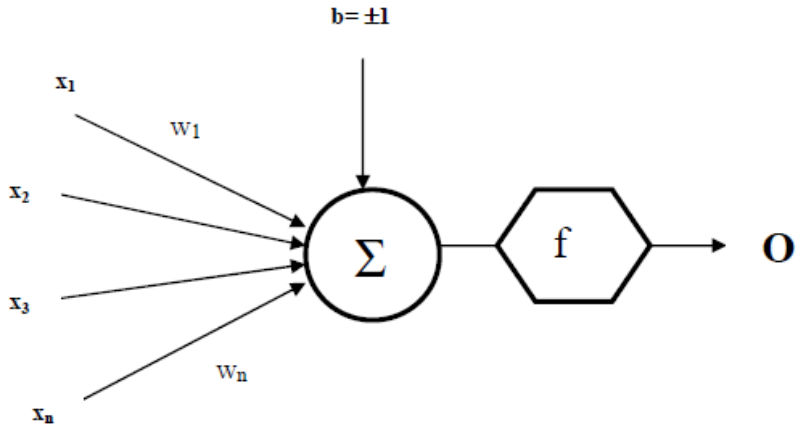
YSA'lar, ağırlıklandırılmış şekilde birbirine bağlanmış birçok işlem biriminden yani nöronlardan oluşan matematiksel sistemlerdir. Bu işlem birim aslında transfer fonksiyonu olarak anılan bir matematiksel denklemdir. Bu işlem birimi diğer hücrelerden verileri alır; bunları birleştirir, dönüştürür ve sayısal bir sonuç elde eder. Bu birimler bir grup halinde işlem gördüklerinden ağ olarak adlandırılır. Yapay hücreler yani birimler birbirleriyle bağlantılar aracılığıyla bir araya gelmeleri yapay sinir ağlarını oluşturur. Hücrelerin aynı doğrultu üzerinde bir araya gelmeleriyle katmanlar oluşmaktadır. YSA'lar üç katmandan oluşur. Bu katmanlar sırasıyla; Girdi katmanı, ara katman, çıktı katmanıdır. Araştırmacını isteğine göre ara katman sayısı arttırılabilir. Giriş katmanı giriş verisini içerir, çıkış katmanı ise ara katmanlarda işlem yapıldıktan sonra oluşturulan sonucu içerir. Şekil 2.2'de yapay sinir ağının katmanları görülmektedir (Özdemir 2010).



Şekil 2.2: YSA Katmanları(Özdemir 2010)

Temel bir yapay sinir ağı hücresi biyolojik sinir hücresine göre çok daha basit bir yapıya sahiptir. En temel hücre modeli Şekil 2.3'de görülmektedir. Bir yapay sinir ağı hücresinin girdiler, ağırlıklar, toplama fonksiyonu, aktivasyon fonksiyonu ve çıktılar olmak üzere 5 ana öğesi vardır. Diğer hücrelerden alınan veriler yani girişler ağırlıklar aracılığıyla hücreye bağlanır. Bu ağırlıkların gelen verinin etkisini göstermektedir. Bu gelen verilerden elde edilen net girdiyi hesaplamak için kullanılan fonksiyona toplam fonksiyonu denir. Genellikle ağırlıklarla gelen bilgi çarpılarak toplanır. Hücrenin girişi toplam fonksiyonla belirlendikten sonra hücrenin bu gelen bilgiyi işleyip bir çıktı üretmesi gerekir. Bu üretilen çıktının hesaplanması

için kullanılan fonksiyona da aktivasyon fonksiyonu denilmektedir. Genelde aktivasyon fonksiyonu doğrusal olmayan bir fonksiyondur (Bilen 2014).



Şekil 2.3: Temel Yapay Sinir Ağı Hücresi(Özdemir 2010)

2.4.1.2 Genetik Algoritmalar

Genetik algoritmalar doğal evrim süreçlerini modelleyerek olası çözümler arasından optimum çözümü arayan ve etkin çözümler sunan bir araştırma tekniğidir. Genetik algoritmalar geleneksel yöntemlerle çözümü zor veya imkânsız olan problemlerin çözümünde kullanılmaktadır. Herhangi bir problemin genetik algoritma ile çözümü, problemi sanal olarak evrimden geçirmek suretiyle yapılmaktadır (Parlak 2007).

Algoritma popülasyon olarak adlandırılan ve kromozomlar tarafından temsil edilen bir dizi sonuçla işlemlere başlamaktadır. Veri madenciliği açısından bakıldığında kromozom, veri tabanındaki her bir kaydı ifade etmekte kullanılmaktadır. Bu kromozomlar üretilecek yeni sonuçlar hakkında bilgiler içermektedir. Eldeki kromozomlar kullanılarak yeni bir sonuç elde edilmektedir. Elde edilen her yeni sonucun bir öncekinden daha iyi olması beklenmektedir. Durma kriterine ulaşılıncaya kadar yeni sonuçların üretimine devam edilir (Şekeroğlu 2010).

Genetik algoritma sürecinin başlaması için öncelikle başlangıç popülasyonundaki bireylerin her birinin uygunluk değerleri hesaplanması

gerekmektedir. Daha sonra seçim yöntemleri kullanılarak bu bireyler içinde yeni popülasyona aktarılacak olanlar seçilecektir. Seçilen popülasyon arasında evrimsel işlemler uygulanmaktadır. Önce çaprazlamaya (crossing-over) maruz kalan kromozomlar daha sonra mutasyon (mutation) geçirmektedirler. Oluşan yeni kromozomların uygunluk fonksiyonları yeniden hesaplanmaktadır. Kalacak olan bireyler seçilir ve elenecek olan bireyler çözümler kümesinden silinirler. Silinen bireyler yerine uygunluk değeri nispeten daha iyi olan çözümlerin kopyaları eklenir. Burada elde edilen çözümlerin her birine birey veya kromozom adı verilir. Uygunluk değerine dayanarak bir sonraki nesilde hangi kromozomların var olacağına ve hangilerinin eleneceğine karar veren yönteme seçme (seleksiyon) işlemi denir. Bu işlem süreci problemin niteliğine ve beklentilerine göre en uygun sonuç elde edilinceye kadar sürer (Kaya 2012).

Genetik algoritmalar açıklanabilir sonuçlar üretirler. Değişik tiplerdeki verileri işleme özelliğine sahiptirler. Ayrıca genetik algoritmalar yapay sinir ağları ile çalışarak başarılı sonuçlar üretmektedirler. Ancak genetik algoritmalarda elde edilen sonucun optimal olduğuna dair bir kanıt bulunmamaktadır (Şekeroğlu 2010).

2.4.1.3 Bayes Sınıflandırması

Bayes teoremi, istatistiksel yöntemler kullanılarak yapılan bir sınıflandırma işlemidir. Genellikle sonrasal olasılıkları hesaplamakta kullanılan ve iki rastgele olayın koşullu olasılıklarını ilişkilendiren bir teoremdir. Örneğin, kilosu ve boyu verilen kişilerin hangi beden sınıfına girdiğini tahmin edebilir (Bahadır 2008).

Bayes teoremi şu şekilde formüle edilir (Bahadır 2008).

$$p(A\backslash B) = \frac{p(A) \times p(B\backslash A)}{p(B)} \tag{2.3}$$

p(A): A'nın olma olasılığı p(B): B'nin olma olasılığı

p(A\B): B olduğu zaman A'nın olma olasılığı

p(B\A): A olduğu zaman B'nin olma olasılığı

2.4.1.4 K-En Yakın Komşu Algoritması

K-En Yakın komşu Algoritması (k-nn) sınıflandırma ve kümeleme alanlarında etkin ve yaygın bir şekilde kullanılan, algoritmik olarak basit bir metottur. Bu yöntem, sınıfları belli olan bir örnek kümesindeki gözlem değerlerinden yaralanarak, örneğe katılacak yeni bir gözlemin hangi sınıfa ait olduğunu belirlemek amacıyla kullanılmaktadır.

Bu yöntemde öncelikle k değeri seçilir. K değerini seçmek için herhangi bir yöntem yoktur. Ama genellikle 3 veya 5 seçilir. Sonra bir gözlem değeri seçilir. Bu gözlem değerinin örnek kümedeki gözlem değerleriyle arasındaki uzaklıklar hesaplanır ve en küçük uzaklığa sahip k sayıda gözlem seçilir. Seçilen gözlemler arasında sayısal olarak karşılaştırma yapılarak en yüksek sayıya ulaşan sınıf seçilir.

Uzaklıkların hesaplanmasında Öklid uzaklık formülü kullanılabilir. Aralarındaki uzaklık hesaplanacak iki vektör x ve y vektörleri olsun bu iki vektör arasındaki uzaklık için aşağıdaki Öklid uzaklık formülü kullanılabilir (Kolyiğit 2013).

$$d(x,y) = \sqrt{\sum_{k=1}^{p}(x_{ik} - x_{ij})^2} \tag{2.4}$$

2.4.1.5 Karar Ağaçları

Karar ağaçları sınıflandırma problemlerinde en çok kullanılan algoritmalardan birisidir. Diğer yöntemlere göre uygulanması ve anlaşılması daha kolay bir yöntemdir. Sınıflandırma yapılabilmesi için öncelikle bir ağaç oluşturulmalıdır. Daha sonra veri tabanındaki her bir kayıt bu ağaca uygulanır ve çıkan sonuca göre de kayıtlar sınıflandırılır (Silahtaroğlu 2013).

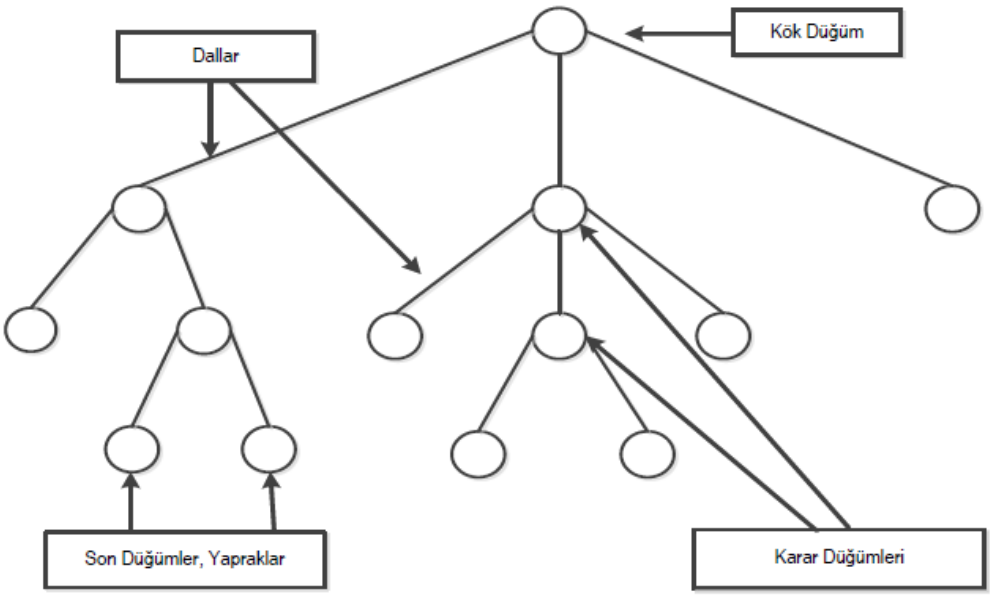
Karar ağaçları temel prensip olarak veri setini eğitim ve test verisi olarak ikiye ayırır. Eğitim verisi karar ağacının oluşturulmasında ve sınıf kurallarının belirlenmesinde kullanılır. Test verileri ise oluşturulan sınıflandırma kurallarının

denenmesi ve karar ağacının başarısının belirlenmesinde kullanılır. Karar ağacı algoritması başarılı bir sınıflandırma gerçekleştirdiyse yeni verilerin oluşan bu kurallar doğrultusunda sınıflandırılması sağlanır (Pala 2013).

Bir karar ağacı bir kök düğümü, karar düğümleri, dallar ve yapraklardan oluşan ağaç yapısına benzer bir akış şemasıdır. Ağaç yapısındaki en dıştaki düğüm kök düğümdür. Bu akış şemasında her düğüm veriye uygulanacak testi tanımlar, her dal testin sonucunu gösterir ve ağacın yaprakları da sınıfları tanımlar. Karar Ağacı oluşturulurken tüm veri kök düğümde toplanır bütün veriler aynı sınıftansa, bu düğüm yaprak haline gelir ve bu sınıfın adını alır. Aksi takdirde veri daha küçük alt kümelere bölünür. Bu bölme işlemi uygun bir bölme kalmayıncaya kadar tekrarlı bir biçimde yapılır. Bu şekilde de sınıfları belirten yaprak düğümler oluşur. Karar düğümleri gerçekleştirilecek testi belirtir. Test niteliğinin her bilinen değeri için bir dal oluşturulur ve tüm veriler buna göre dallara ayrılır. Dalın sonunda veri daha fazla bölünemiyorsa ve dalın sonunda belirli bir sınıf oluşuyorsa, odalın sonunda yaprak vardır. Her bir sınıf ağaçta tek yaprak olarak gösterilir. Bu yüzden bir sınıfa giden sadece bir yol olmalıdır. Yapraklar arasında her hangi kısa bir yol veya bağ yoktur. Dallanma işlemi yaprak düğüme ulaşılıncaya kadar devam eder. Karar ağacı işlemi kök düğümünden başlar ve yukarıdan aşağı doğru yaprağa ulaşana kadar ardışık düğümleri takip ederek gerçekleşir. Böylelikle verinin hangi sınıfta yer aldığı belirlenmiş olur. Şekil 2.4'de bir karar ağacının örnek yapısı gösterilmiştir (Aksoy 2014).

Karar ağacı oluşturulurken yapılan başka bir işlem de budama işlemidir. Öğrenme verisinden oluşan ağaç çok büyük olabilir. Böyle bir ağaçta öğrenme kümesi verisinden başka bir veriyle test edildiğinde doğruluğu çok yüksek sonuçlar elde edilebilir. Ancak böyle bir ağaç test verisiyle test edildiğinde doğru sonuçlar üretmeyebilir. Ağacın böyle çok büyümesine şişme (overfitting) denir. Bunun iki nedeni olabilir. Birincisi veri içinde gürültü vardır. Gürültü ağaçta gereksiz dallanmalara ve gereksiz kurallara neden olur. İkinci neden ise veri kümesinin o olayı temsil yeteneğinin olmamamsıdır. Ağacın dengeli olabilmesi için belli bir büyüklüğün üzerinde olması gerekir. Bu büyüklük arttıkça da test verisinde hata oranı artar. Böyle durumlarda yapılması gereken işlem budama işlemidir. Budama işlemi, bazı dalların ya da alt dalların kaldırılarak o dala ait nesnelerin baskın sınıfı

yaprak olarak yaratılır. İki türlü ağaç budama tekniği bulunmaktadır. Bunlar Ön budama(Pre-Pruning) ve sonradan budama(Post-Pruning)'dır. Ön budama işlemi ağaç yaratılırken yapılırken, sonradan budama ağaç oluşturulduktan sonra yapılır. Ön budama ağaç oluşurken yapıldığından ağacın yeterli olgunluğa ulaşmasını engellemektedir. Bu da hatalı sonuçlar üretebilir (Koçtürk 2010).



Şekil 2.4: Karar Ağacı Yapısı(Pala 2013)

Karar ağacı kullanımının kullanıldığı duruma göre avantaj ve dezavantajları vardır. Avantajları arasında aşağıdaki durumlar sayılabilir (Sezer 2008).

- Karar ağacı oluşturmak zahmetsizdir, yorumlamak kolaydır.
- Anlaşılabilir kurallar oluşturulabilir.
- Sürekli ve ayrık nitelik değerler kullanılabilir.

Dezavantajları ise;

- Sürekli nitelik değerlerini tahmin etmekte çok başarılı değil.
- Sınıf sayısı fazla ve öğrenme kümesi örnekleri sayısı az olduğunda model oluşturma çok başarılı değil.

- Zaman ve yer karmaşıklığı öğrenme kümesi örnekleri sayısına, nitelik sayısına ve oluşan ağacın yapısına bağlıdır.
- Ağaç oluşturma karmaşıklığı ve ağaç budama karmaşıklığı fazladır.

Veri Madenciliği kullanılan birçok karar ağacı algoritması bulunmaktadır. Bunlardan bazıları arasında ID3, C4.5, C5.0, CART, QUEST, SPRINT, SLIQ algoritmaları yer almaktadır. Bu çalışmada karar ağacı algoritması olarak C4.5 algoritması kullanılmıştır.

2.4.1.5.1 C4.5 Algoritması

ID3 ve C4.5 algoritmaları dallanmanın hangi niteliğe göre olacağını belirlemek için entropi kavramından yararlanır. Entropi, eldeki bilgilerin sayısallaştırılmasıdır. Yani entropi, eldeki verinin belirsizliğinin ölçülmesi anlamına gelir. entropi 0-1 arasında değişen bir değer alır. Verilerin hepsi tek bir sınıfa aitse entropi sıfır(0) olacaktır. Bütün olasılıklar eşit olduğunda ise entropi 1 değerini alır. entropi hesabı için kullanılan matematiksel formül aşağıdaki gibi verilebilir.

$$H(p_1, p_2, \dots, p_n) = \sum (p_i \log(1/p_i)) \qquad (2.5)$$

Burada ($p_1, p_2, \dots, p_n$)olasılıkları ifade etmektedir ve tüm olasılıkların toplamı 1'e eşittir.

ID3 ve C4.5 algoritmaları veritabanı bölünmeden önce doğru sınıflandırma yapmak için nitelikler arasında bir ilişki kurar. Bu ilişki, veritabanı bölünmeden önce gelen bilgi ile bölündükten sonra gelen bilgi arasındaki farktır. Bu aradaki fark kazanım olarak adlandırılır. Kazanım bize öncelikli düğüme ve dallanmalara karar vermemize yardımcı olur. Kazanım şu şekilde hesaplanır: Verilerin ham halinin entropisi ile her bir alt bölümün entropilerinin ağırlıklı toplamı arasındaki fark alınır. ID3 algoritmasında bu fark hangi alt bölüm için büyükse o alt bölüme doğru dallanma yapılır (Silahtaroğlu 2013).

$$Kazanım(D;S) = H(D) - \sum_{i=1}^{n} P(D_i)H(D_i) \quad (2.6)$$

ID3 algoritmasını geliştiren Quinlan, bu algoritmada bulunan bazı eksikleri ve sorunları gidererek C4.5 algoritmasını oluşturdu. ID3 algoritmasında bazı veritabanlarında niteliklerin özelliklerinin çok çeşitli olmasından kaynaklanan kazanım bilgisinin yüksek çıkması gereksiz kural oluşmasına neden olabiliyor. Bu sorunu gidermek için Quinlan C4.5 algoritmasında bölünme bilgisi kavramıyla algoritmasını yeniledi.

Bu algoritma değer çeşitliliği fazla olan özelliklerin bilgi kazancını azaltarak algoritmanın gereksiz bazı çıkarımlar yapmasını engellemektedir. Bu noktada bölünme bilgisi denilen yeni bir kavram ekleniyor bu algoritmaya. A bir özellik, Ai bu özelliğin değerleri, Ti Ai özelliğinin bu veride kaç kez tekrarlandığı ve T ise ele alınan olay sayısını temsil etsin. Bu durumda bölünme bilgisi;

$$-\sum_{i=1}^{n} \frac{T_i}{T} \times \log_2(\frac{|T_i|}{T}) \quad (2.7)$$

Olarak ifade edilir. Bu bölünme bilgisi tüm özelliklerin bilgi kazanç formülüne bölen olarak eklenir ve bu kazanç oranı olarak ifade edilir. Bu durumda A özelliğinin kazanç oranı;

$$Kazanım\ Oranı(A) = \frac{Kazanım(A)}{BölünmeBilgisi(A)} \quad (2.8)$$

Şeklinde hesaplanır (Han ve Kamber 2006).

C4.5 algoritmasını ID3 algoritmasından ayıran diğer özellikler; özelliklerin kayıp değerleriyle baş edebilmesi ve sayısal özellik değerlerini de hesaba katabilmesidir.

2.4.1.6 Destek Vektör Makineleri

Destek vektör makinelerinin(DVM) zemini 1960'lara dayansa da ilk olarak Vladimir Vapnik ve arkadaşları Bernhard Boser ve Isabelle Guyon tarafından 1992 yılında yayınlanmıştır. DVM'nin eğitim süresi son derece yavaş olmasına rağmen, karmaşık ve doğrusal olmayan karar sınırlarını belirlemede oldukça doğru kararlar vermektedir. DVM'ler diğer metotlara göre aşırı uyuma daha az eğilimlidir. DVM'ler sınıflandırmanın yanı sıra tahminde de kullanılabilmektedir. DVM'ler el yazısı tanıma, nesne tanıma, konuşmacı tanıma gibi bir çok alanda kullanılmaktadır (Han ve Kamber 2006).

Veri madenciliğinde sınıflama problemlerinde kullanılan bir diğer yöntem Destek Vektör Makinesi yöntemidir. Bu yöntem, sınıflandırmayı doğrusal ya da doğrusal olmayan bir fonksiyon yardımıyla yerine getirir. Destek vektör makinesi yöntemi, veriyi birbirinden ayırmak için en uygun fonksiyonu tahmin etmeye çalışır (Yalçın 2013).

DVM temelde iki sınıflı problemlerin çözümünde doğrusal bir sınıflayıcı kullanırken, doğrusal olarak ayrılamayan veya çok sınıflı sınıflama problemlerinin çözümünde de kullanılmaktadır.

Doğrusal olarak ayrılabilen problemlerde, verileri ayırabilecek sonsuz sayıdaki doğru içinden en uygun doğru seçilmeye çalışılır. Bunu için iki sınıfın sınırlarında birbirine en yakın iki örneğin arasındaki mesafenin (marj) en fazla olması amaçlanır ve bu en iyi sağlayan ayırıcı doğru seçilmeye çalışılır. Doğrusal olarak sınıflandırılamayanlar ise kernel(çekirdek) fonksiyonları kullanılarak çok boyutlu bir uzaya aktarılır. Bu uzayda verileri sınıflara ayıran düzlemler arasından en iyi ayıran üstün düzlem bulunmaya çalışarak yüksek boyutlu uzayda verilerin sınıflandırılması gerçekleşir (Kartal 2012).

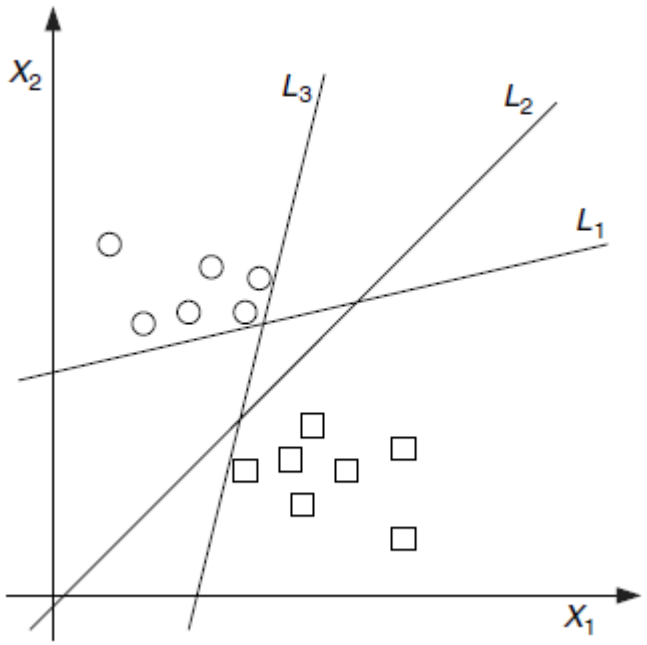
DVM'yi etkin bir şekilde kullanabilmek için DVM'nin nasıl çalıştığını iyi bilmek gerekir. Nerde hangi çekirdek fonksiyonu kullanılmalı, DVM'de hangi parametreler kullanılmalı bunlarla ilgili kararlar doğru verilmelidir. Aksi takdirde istenilen performans elde edilemez.

DVM'lerin sınıflandırma mekanizması, üç ayrı veri durumu için detaylandırılabilir.

1. Doğrusal olarak ayrılabilir veriler
2. Doğrusal olarak ayrılamaz veriler
3. Ayırımı doğrusal olmayan veriler

2.4.1.6.1 Doğrusal Olarak Ayrılabilir Veriler

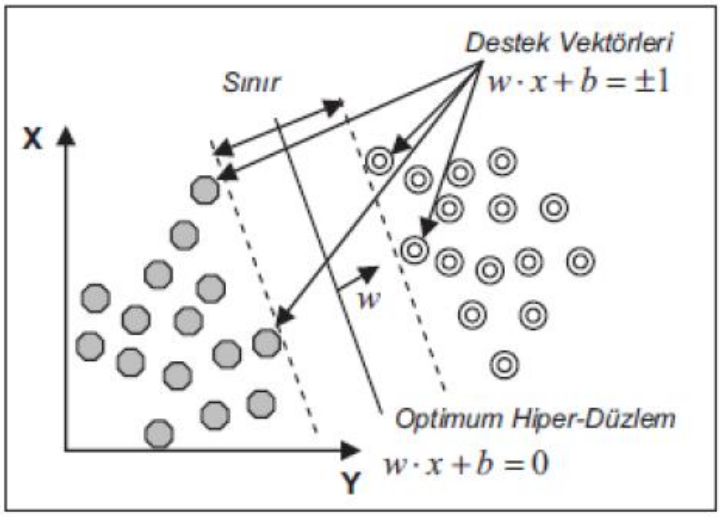
Destek vektör makinesinin en basit ve ilk olarak tanıtılan modeli doğrusal sınıflandırıcıdır. Eğitim verileri X=(x1,x2,x3,…,xn) ve $y_i \in \{-1,1\}$, i=1,2,3,…,n, $x_i \in R^n$ olarak tanımlanabilir. Aşağıdaki şekilde görüldüğü gibi veriyi iki boyutlu alanda göz önüne alalım.



Şekil 2.5: Doğrusal olarak ayrılabilen veri(Olsen ve Delen 2008)

Verinin birbirinden farklı biçimlerde doğrusal olarak ayrılacağı görülmektedir. Şekil üzerinde görüldüğü gibi veri farklı ve çok sayıda doğru ile ayrılabilmektedir. Çok boyutlu uzayda bu doğruların yerini hiper düzlemler almaktadır. Veriyi birbirinden ayıran bu hiper düzlemlerden bir tanesi maksimum ayırma başarısına sahiptir. Maksimum ayırma başarısı veri setindeki iki sınıfın

birbirine en yakın noktalarını en iyi şekilde sınıflandıracak en geniş aralığın seçilmesi anlamına gelir.



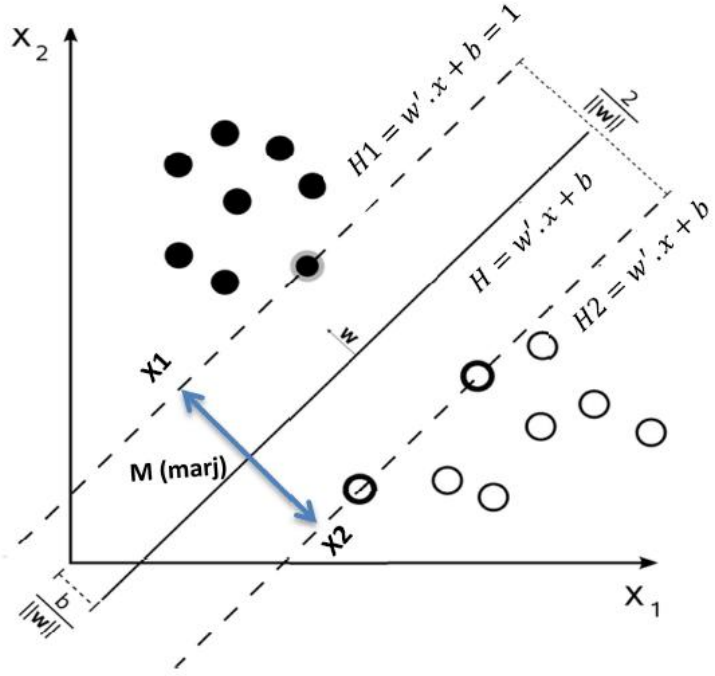
Şekil 2.6: Destek Vektörler(Bahadır 2008)

Şekilde düz çizgi ile belirtilen doğru optimum hiper düzlem sınıflandırıcısıdır. Destek vektörleri arasında kalan kısım ise sınır(marj) olarak adlandırılmaktadır. Bu marjın neden maksimum olması gerektiğinin birçok açıklaması bulunmaktadır. Nedenlerden bir tanesi kümeler birbirinden ne kadar uzak olursa yanlış sınıflandırma o kadar düşük olacaktır. Diğer bir neden ise yerel minimumdan kaçınmayı sağlamasıdır.

Veriyi iki sınıfa ayıran hiper düzlemin bir tarafında kalan veriler ve y_i=+1, diğer tarafında kalanlar ise y_i =-1 olarak etiketlenirler. Bir hiper düzlemin genel formu;

$$(w'.x) + b = 0 \qquad (2.9)$$

Şeklinde ifade edilir. Bu eşitlikte x bir vektör noktası, w ağırlık vektörü ve b yan (bias) plmak üzere bir sabit sayıdır. w ağırlık vektörü ayırıcı hiper düzleme dik normal vektördür, b sabiti ise hiper düzlemin orijinden ne kadar sapacağını belirler (w_0). Şekil 2.7'de bu eşitlik ve vektörlerin gösterimine ilişkin bir örnek verilmiştir (Bahadır 2008).



Şekil 2.7: Marj hesaplaması(Karakaynak 2014)

Şekilde yuvarlak içine alınan gözlemler destek vektörlerdir. Destek vektörlerden geçen hiper düzlemler şu formda gösterilirler.

$$H1: (w'.x) + b = +1 \tag{2.10}$$

$$H2: (w'.x) + b = -1 \tag{2.11}$$

H1 ve H2 hiper düzlemleri arasındaki uzaklığı bulmak için H1 ve H2 üzerinde birer x noktası alınırsa, H1 üzerindeki x1, H2 üzerindeki x2 olur. Bu durumda bu uzaklık geometri yardımıyla denklemler (2.12) İle (2.13) Arasındaki işlemler ile bulunabilir (Karakaynak 2014).

$$d = H1 - H2 \tag{2.12}$$

$$(w'.(x1 - x2)) = 2 \tag{2.13}$$

$$d = \left(\frac{w}{\|w\|}.(x1 - x2)\right) = \frac{2}{\|w\|} \tag{2.14}$$

Amaç d'yi yani marjı maksimum yapmaktır. Bu durumda yapılması gereken $\frac{2}{\|w\|}$ ifadesinin maksimize edilmesidir. Bunun için ise $\frac{2}{||w||}$ ifadesinin minimum

yapılması gerekmektedir. ||w|| minimum yaparken aynı zaman da veri noktalarının marjın içine düşmesini engellemek için (2.15) ve (2.16)'de gösterilen kısıtlar da bu minimizasyon problemine eklenmelidir (Karakaynak 2014).

$$y = -1 \; için \; (w'.x) + b \leq -1 \tag{2.15}$$

$$y = +1 \; için \; (w'.x) + b \geq +1 \tag{2.16}$$

Şekil 2.7'de ayırıcı hiper düzlemin üst tarafında (pozitif tarafında) kalan ifadeler için (2.16) eşitsizliği, alt tarafında kalan ifadeler için ise (2.15) eşitsizliği kullanılabilir. Bu iki eşitsizlik (2.17)'deki şekilde birleştirilebilir (Karakaynak 2014).

$$y_i(w'.x + b) - 1 \geq 0 \tag{2.17}$$

Maksimum sınırın bulunması işlemi;

$$min \frac{1}{2}\|w\|^2 \tag{2.18}$$

$$y_i(w'.x_i + b) - 1 \geq 0 \quad \forall i \tag{2.19}$$

İle ifade edilir. Burada (2.18) çözülecek problem ve (2.19) problemin çözümü sırasında kullanılan koşuldur ve bu ifade ikinci dereceden optimizasyon problemidir. Problemin çözümü için problemin Langrange formülasyonu yapılır. Pozitif lagrangian çarpanları a_i'ler kullanılarak dönüştürülen yeni optimizasyon problemi aşağıdaki gibidir;

$$L_p = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} a_i y_i \; (w'.x_i + b) + \sum_{i=1}^{n} a_i \tag{2.20}$$

Bu Lagrangian formülü birincil (primal) değişkenler w ve b bakımından minimize edilmeli, ikincil (dual) değişkenler bakımından maksimize edilmelidir. Ancak bu problemin çözümü oldukça karmaşıktır. Dolayısıyla Karush- Kuhn-Tucker koşulları olarak bilinen yöntem ile çözüm sağlamak için öncelikle formül (2.20)'nin w ve b'ye göre türevleri alınır (Karakaynak 2014).

$$\frac{\partial L_p}{\partial w} = 0 \; \rightarrow w = \sum_{i=1}^{n} a_i y_i x_i \tag{2.21}$$

$$\frac{\partial L_p}{\partial b} = 0 \rightarrow w = \sum_{i=1}^{n} a_i y_i \tag{2.22}$$

Bu koşullar (2.20)'de yerine yazılacak olursa;

$$L_p = \frac{1}{2}(w'w) - w' \sum_{i=1}^{n} a_i y_i x_i - b \sum_{i=1}^{n} a_i y_i + \sum_{i=1}^{n} a_i \tag{2.23}$$

$$L_p = -\frac{1}{2}(w'w) + \sum_{i=1}^{n} a_i \tag{2.24}$$

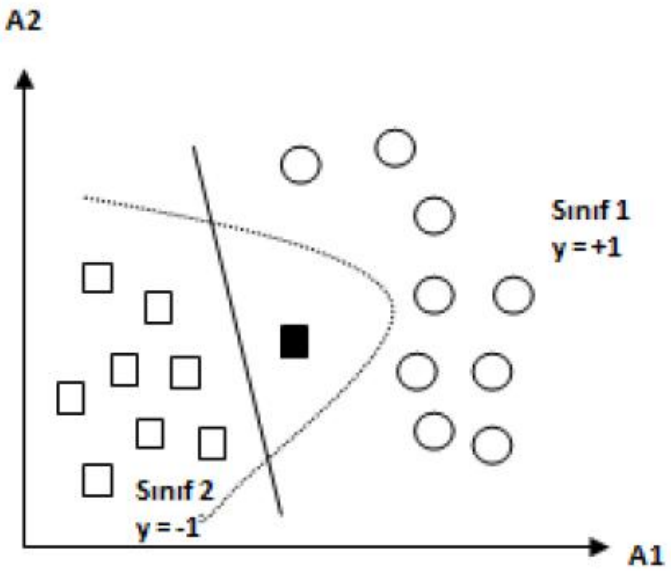
$$L_p = \sum_{i=1}^{n} a_i - \frac{1}{2} \sum_{i,j}^{n} a_i a_j y_i y_j x_i' x_j \quad a_i \geq 0 \; \forall i \tag{2.25}$$

İfadesi elde edilmiş olur. Burada dikkat edilirse her eğitim örneği için bir tane Lagrange çarpanının olduğu görülür. Çözümde elde edilen Lagrange çarpanlarının büyük çoğunluğunun değeri sıfır olacaktır. Geriye kalan $a_i > 0$ değerli x_i örnekleri Destek vektörlerdir ve H1 ve H2 hiper düzlemlerinin arka taraflarında kalan örneklerdir (Karakaynak 2014).

2.4.1.6.2 Doğrusal Olarak Ayrılamaz Veriler

Doğrusal olarak ayrılabilinen verileri iki sınıfa ayırırken doğrusal bir düzlem kullanılabiliyordu. Gerçek hayattaki durumlarda bu her zaman geçerli olmayabilir. Yani veriler doğrusal bir düzlem ile birbirinden ayrılmayabilir. Veride gürültüler veya yanlış veri girişleri olabilir. Bu tür durumlarda ayırıcının hata yapmasına izin verilir (Kartal 2012).

Eğer ki, örnekler doğrusal olarak tamamen ayrılabilir durumda değilse problemin çözümü için pozitif zayıflık değişkenleri, ξ_i, i = 1,2,…..,N kullanılır. Aşağıda ki denklemler zayıflık değişkenleri ile yeniden tanımlanarak oluşturulmuş ifadelerdir:



Şekil 2.8: Doğrusal olarak ayrılamayan veri(Yakut 2012)

$$y_i = +1 \; için \;\; w^T.x_i + b \geq +1 - \xi_i \tag{2.26}$$

$$y_i = -1 \; için \;\; w^T.x_i + b \geq -1 + \xi_i \tag{2.27}$$

$$\xi_i \geq 0, \qquad \forall i \tag{2.28}$$

$\xi_i = 0$ olması durumunda x_i örneği doğru sınıflandırılmış, $0 < \xi_i < 1$ olması durumunda x_i örneği doğru sınıflandırılmış ancak marj bölgesi içerisinde yer alıyor, $\xi_i \geq 1$ ise yanlış sınıflandırılmış demektir.

Doğrusal olarak ayrılamama durumunda eğitim hatası için bir C üst sınırı eklenir. Bu üst sınır Lagrange çarpanlarının alabilecekleri maksimum değeri göstermektedir. Bu şekilde Lagrange çarpanlarının $0 \leq a_i \leq C$ aralığında kalması sağlamaktadır. Bu bilgilere göre Lagrange formülasyonu yeniden şu şekilde ifade edilecektir (Yakut 2012);

$$L_p = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N} a_i\{y_i(w^T.x_i + b) - 1 + \xi_i\} - \sum_{i=1}^{N}\mu_i\xi_i \tag{2.29}$$

Yukarıdaki formülasyonda μ_i, ξ_i'nin pozitif olmasını sağlamak için kullanılan Lagrange formülasyonunda çözülmesi zor olduğundan dolayı dual problemine dönüştürülmektedir. Bu problemde Karush-Kuhn-Tucher şartları uygulanırsa;

$$\frac{\partial L_p}{\partial w} = w - \sum_{i=1}^{N} a_i y_i x_i = 0 \tag{2.30}$$

$$\frac{\partial L_p}{\partial b} = -\sum_{i=1}^{N} a_i y_i = 0 \tag{2.31}$$

$$\frac{\partial L_p}{\partial \xi} = C - a_i - \mu_i = 0 \tag{2.32}$$

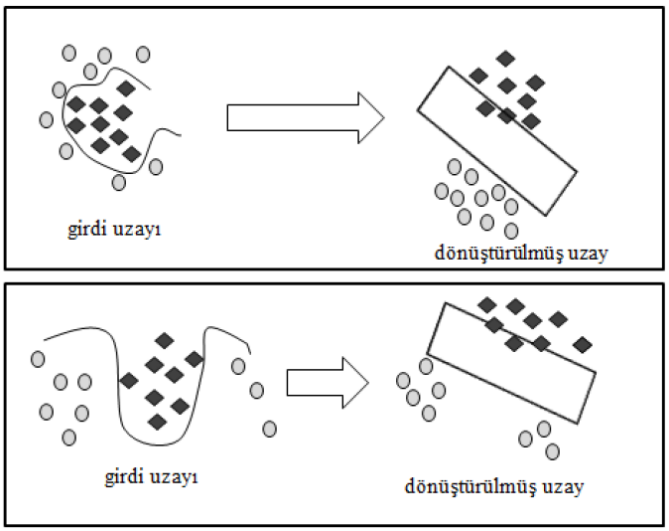
İfadeleri elde edilir. Bu ifadeler tekrar düzenlenirse;

$$L_p = \sum_{i=1}^{N} a_i - \frac{1}{2}\sum_{i,j}^{N} a_i a_j y_i y_j x_i^T x_j \quad 0 \le a_i \le C, \forall i \tag{2.33}$$

Elde edilir. Bu problemin çözümünde $0 \le a_i \le C$ aralığında yer alan Lagrange çarpanlarına karşılık gelen x_i değerleri destek vektörlerdir (Yakut 2012).

2.4.1.6.3 Ayrımı Doğrusal Olmayan Veriler

Şimdiye kadar veri kümelerinin doğrusal bir hiper düzlem ile ayrılabildiği durumlar irdelendi. Ancak uygulamada her zaman yukarıda anlatıldığı gibi verilerin doğrusal olarak ayrılabildiği durumlarla karşılaşılmamaktadır. Şekil 2.9'da olduğu gibi olduğu gibi iki sınıf iç içe geçmiş gibi ya da veri grupları arasında kalmış gibi bir yapı gösterebilir (Uçar 2013).



Şekil 2.9: Ayrımı doğrusal olmayan veri(Uçar 2013)

DVM'ler böyle durumlarla karşılaştığında verileri doğrusal olarak ayırabileceği n boyutlu girdi uzayından daha yüksek boyuta sahip olay (feature) uzayına taşır.

$$x \in R^n \rightarrow \Phi(x) \in R^f \tag{2.34}$$

Doğrusal olmayan DVM, verilerin taşındığı bu yeni boyutta doğrusal DVM gibi çalışarak verileri ayıracak optimum çoklu düzlem arar.

Dönüştürme işlemi için kullanılacak fonksiyon $\Phi(x)$, x'lerin $\Phi(x)$ 'e dönüşümünü sağlayan sabit bir fonksiyondur. Girdi uzayını oluşturan x vektörü x_i gözlemlerinden oluşurken, özellik uzayını oluşturan $\Phi(x)$ vektörü $\Phi_i(x)$ 'lerden oluşmaktadır. Böylelikle Φ uzayında x görüntülerinin doğrusal olarak sınıflandırılabildiği bir ortam bulunması amaçlanmaktadır. Buradan hareketle dönüştürülmüş uzayda kullanılacak karar fonksiyonu (Karakaynak 2014):

$$< w, \Phi(x) > + b = 0 \tag{2.35}$$

Şeklinde olacaktır. Destek vektörlerinin üzerinde yer aldığı ve ayırıcı çoklu düzleme paralel doğruların ayırdığı veriler aşağıdaki şekilde sınıflanır:

$$< w, \Phi(x) > + b \geq +1 \tag{2.36}$$

$$< w, \Phi(x) > + b \leq -1 \tag{2.37}$$

Aynı şekilde nesne fonksiyonu ve buna ilişkin formül (3.25) ile (3.26)'nın birleşiminden oluşan kısıt aşağıdaki gibidir:

$$min_{w,b}\tau(w) = \frac{1}{2}\|w\|^2 \tag{2.38}$$

$$y_i(< \Phi(x).w > + b) - 1 \geq 0\,, \quad \forall i \; için \tag{2.39}$$

Burada iki sorun ortaya çıkmaktadır. İlk olarak dönüştürülmüş uzayda oluşturulacak doğrusal karar sınırı ile ilgili nasıl bir haritalama fonksiyonu kullanılacağı açık değildir. İkinci sorun ise uygulanan haritalama fonksiyonu biliniyorsa, kurulan optimizasyon probleminin yüksek boyutlu olay uzayında çözümü zor ve karmaşık hesaplamalar gerektirecektir. w ve b parametrelerini hesaplamak için:

$$w = \sum a_i y_i \Phi(x_i) \tag{2.40}$$

$$f(x) = \left(\sum a_i y_i \Phi(x_i) . \Phi(x) \right) + b = 0 \tag{2.41}$$

Yukarıdaki denklemler dönüştürülmüş uzaydaki iki vektörün iç çarpımını içermektedir. Boyut sorunundan dolayı bu iç çarpımların hesaplanması zordur. Bu sorunu önlemek amacıyla çekirdek düzenlemesi olarak adlandırdığımız "Kernel Trick" yöntemi önerilmiştir (Karakaynak 2014).

Çekirdek Düzenlemesi ve Çekirdek Fonksiyonları

Çekirdek fonksiyonu kullanmanın temel avantajı, bütün değerlerin tekrar tekrar iç çarpım değerlerinin hesaplanarak bulunması yerine doğrudan çekirdek fonksiyonunda değerin yerine koyularak nitelik uzayındaki değerinin bulunmasıdır. Bu sayede, son derece yüksek boyutlu bir nitelik uzayı ile uğraşma olasılığı kalmaz. Diğer avantajı ise, direk girdi uzayındaki veriler kullanılacağı için Φ haritalama fonksiyonun kesin olarak ne olduğunu bilmeye gerek yoktur.

İç çarpımlar iki girdi vektörü arasındaki benzerliğin bir ölçüsüdür. Çekirdek fonksiyonu da veriyi kullanarak dönüştürülmüş uzayda bir benzerlik hesaplaması yapar. Dönüştürülmüş uzaydaki iki girdi vektörü u ve v için iç çarpımlar:

$$\Phi(u)\Phi(v) = \left(u_1^2, u_2^2, \sqrt{2}u_1, \sqrt{2}u_2, 1\right)\left(v_1^2, v_2^2, \sqrt{2}v_1, \sqrt{2}v_2, 1\right) = (uv+1)^2 \tag{2.42}$$

Dönüştürülmüş uzaydaki iç çarpımlar orijinal nitelik verisinden hesaplandığı için bu benzerlik fonksiyonu K ile gösterilen "çekirdek fonksiyon" olarak adlandırılır.

$$K(u,v) = \Phi(u)\Phi(v) = (uv+1)^2 \tag{2.43}$$

DVM yaygın olarak kullanılan dört çekirdek fonksiyonu vardır. Bu fonksiyonlar:

1) Doğrusal Fonksiyon
2) Polinomiyal Fonksiyon
3) Sigmoid Fonksiyon
4) Radyal Tabanlı Fonksiyon

Doğrusal Fonksiyon

Girdi uzayında veriler doğrusal olarak ayrılabiliyor ise veriyi yüksek boyuta taşımaksızın doğrusal çekirdek fonksiyonu yardımıyla sınıflama işlemi yapılır. Bu fonksiyon herhangi bir boyut değeri ya da katsayı içermemektedir (Uçar 2013).

$$K(x_i, x_j) = x_i^T x_j \tag{2.44}$$

Polinomiyal Fonksiyon

Polinomiyal çekirdek fonksiyon, d gibi belirli bir derecede girdi vektörlerinin iç çarpımından oluşmaktadır. Fonksiyonun matematiksel gösterimi aşağıdaki şekildedir:

$$K(x_i, x_j) = (x_i x_j)^d \tag{2.45}$$

d=1 olduğu durumlarda polinomiyal fonksiyon doğrusal fonksiyona dönüşmektedir (Uçar 2013).

Sigmoid Fonksiyon

Sigmoid fonksiyon k ve δ gibi iki parametre içermektedir. Kaynaklar belirli parametreler için sigmoid radyal tabanlı fonksiyon çalıştığını göstermektedir (Uçar 2013).

$$K(x_i, x_j) = \tanh(kx_i, x_j - \delta) \tag{2.46}$$

Radyal Tabanlı Fonksiyon

Radyal tabanlı fonksiyon çekirdek fonksiyonlar arasında kullanımı en yaygın çekirdek fonksiyondur. R programında sistem standart çıktılarını radyal tabanlı fonksiyona göre vermektedir. γ yarıçap kontrolünü sağlayan parametredir (Uçar 2013).

$$K(x_i, x_j) = \exp\left(-\gamma \left\| x_i - x_j \right\|^2\right), \qquad \gamma > 0 \tag{2.47}$$

2.4.2 Kümeleme

Kümele, veri tabanındaki verilerin benzer özelliklerine göre gruplara ayrılmasıdır. Buradaki amaç benzer özelliklere sahip nesneleri bir araya toplayarak farklı özelliklere sahip gruplar oluşturmaktır. Yani aynı küme içindeki nesnelerin benzerliği fazla olmalı, diğer kümelerle benzerlik az olmalıdır. Sınıflandırma tekniği de verileri gruplara ayırmak için kullanılan bir araçtır ancak sınıflandırma tekniğinde sınıflar önceden bellidir. Kümele yönteminde ise verilerin hangi gruplara ayrılacağı belli değildir. Verilerin birbirine olan benzerliğine göre gruplar belirlenir (Han ve Kamber 2006).

Kümeleme yönteminde veriler sadece kendi değerlerine göre değil, diğer verilere olan yakınlığına veya uzaklığına göre ve diğer verilerin durumuna göre de kümelere ayrılıyor. Bu da kümeleme yöntemini dinamik bir yöntem haline getiriyor. Bu özellik kümele yöntemini sınıflandırma yönteminden ayıran başka bir özelliktir (Saygılı 2013).

Kümeleme yöntemi birçok alanda kullanılmaktadır. Biyoloji alanında bitki ve hayvanların sınıflandırılmasında ve bu canlıların genlerinin işlevlerine göre gruplara ayrılmasında kullanılmaktadır. Pazarlama alanında farklı müşteri gruplarının alışveriş davranışlarını belirlemede kullanılmaktadır. Kredi kartı dolandırıcılığının tespitinde kullanılmaktadır. Ayrıca istatistik, makine öğrenmesi, veri madenciliği alanlarında da kullanılmaktadır (Han ve Kamber 2006).

İyi bir kümeleme analizi yöntemi şu özelliklere sahip olmalıdır (Han ve Kamber 2006).

- Ölçeklenebilir olmalıdır. Oluşturulan kümele algoritması küçük veri setlerinde başaralı olduğu gibi büyük veri setlerinde de başarılı olmalıdır.
- Farklı veri türleri ile kullanılabilmelidir. Sadece sayısal veri türleri ile değil, diğer veri türleri ile de kullanılabilmelidir.
- Farklı şekle sahip kümeleri de bulabilmelidir.
- Az sayıda giriş değişkeni gerektirmelidir. Giriş değişkenleri arttıkça kümeleme işlemi zorlaşır.

- Gürültü içeren veriler ile de kullanılabilmelidir.
- Veri kümesindeki verilerin sıralamasından etkilenmemelidir. Kümeye yeni veri giriş olduğunda küme bundan etkilenmemelidir.
- Çok boyutlu veri tabanlarına uygulanmalıdır.
- Kümele işlemi yaparken belirtilen kısıtlamaları dikkate alabilmelidir.
- Yorumlanabilir ve kullanabilir sonuçlar üretmelidir.

3. UYGULAMA

3.1 Amaç

Bu çalışmanın amacı Meslek Yüksek Okulu öğrencilerinin başarılarını tahmin etmektir. Bu amaç doğrultusunda veri madenciliği yönteminin sınıflama algoritmaları kullanılarak öğrencilerin başarılarını tahmin etme de en iyi performansı gösteren sınıflama algoritması seçilecektir.

Bu amaçla, çalışmanın uygulama kısmında Pamukkale Üniversitesi'nin bilgi işlem merkezinden Pamukkale Üniversitesi Meslek Yüksek Okullarına 2009 ile 2012 yılları arasında kayıt yaptıran öğrencilerin bilgileri kullanılmıştır. Pamukkale Üniversitesi her yıl üniversiteye yeni kayıt yaptıran öğrencilere bir anket uygulamaktadır ve uyguladıkları bu anket verilerini veri tabanlarında saklamaktadır. Bu çalışma doğrultusunda Pamukkale Üniversitesi'nden 2009-2012 yılları Meslek Yüksek Okullarına kayıt yaptıran öğrencilerin kayıt bilgileri ile ailevi durum bilgilerini içeren anket sorularına verdikleri cevaplar üniversitenin bilgi işlem merkezinden talep edilmiştir. Elde edilen veriler önişlemden geçirilmiş, temizlenmiş ve çalışmaya uygun hale dönüştürülmüştür.

Öğrencilerin başarılarını tahmin etmek için iki hedef nitelik seçilmiştir. Bu nitelikler öğrencilerin akademik ortalamaları ile mezun oldukları yıllar olarak seçilmiştir. Her bir hedef nitelik için Weka programında sınıflama algoritmaları uygulanmıştır. Yine her bir hedef nitelik için uygulanan sınıflama algoritmalarının sonuçları karşılaştırılarak en iyi performansı sergilen algoritmalar bulunmaya çalışılmıştır.

3.2 Veri Toplama Süreci

Bu çalışmada araştırmanın evreni olarak 2009 yılından 2012 yılına kadar Pamukkale Üniversitesi Meslek Yüksek Okuluna giriş yapan tüm öğrencilerin

verileri kullanılmıştır. Belirtilen yıllar arasında Pamukkale Üniversitesi Meslek Yüksek Okuluna kayıt yaptıran tüm öğrencilere üniversite tarafından bir anket uygulanmıştır ve bu anket verileri üniversitenin öğrenci bilgi işlem merkezi tarafından veritabanına kaydedilmiştir. Bu araştırma için üniversitenin bilgi işlem merkezine başvuru yapılarak öğrencilerin üniversiteye giriş yılı, programı, giriş puanı, akademik ortalaması, durumu, mezuniyet yılı ile birlikte ailevi durum bilgilerini içeren toplamda 13 sorudan oluşan ankete verilen cevapları istenmiştir. Üniversite bilgi işlem tarafından istenilen tüm bilgiler bir excel dosyası formatında verilmiştir.

3.3 Veri Temizleme

Üniversite bilgi işlemi tarafındın excel formatında sunulan dosya da öğrencilere ait olan bazı niteliklerin sütün bazı niteliklerin satır olarak sunulduğunu görüldü. Veriler üzerinde yapılacak işlemlerde niteliklerin hepsinin aynı düzende olması gerektiği düşünülerek tüm nitelikleri sütunlar haline dönüştürüldü. Bunu yapmak için de Visual Basic Script'leri kullanılmıştır. Excel dosyası içine yazılan visual basic scriptleri ile satırlar halinde olan nitelikler sütunlara dönüştürülmüştür. Yapılacak çalışmaya katkı sağlayacak nitelikler ve alabileceği değerler Tablo 3.1'de belirlenmiştir.

Bu veri setinde "giriş yılı", "giriş puanı", "Akademik ortalaması", "Lise Diploma notu" nitelikleri içerisine girilen sıfır değerli veya null değerli veriler ile bu alanlara girilen yanlış değerli veriler excel programında bu alanlara filtre uygulanarak tespit edilmiş ve silinmiştir. Sadece "mezuniyet yılına" null girilen veriler ile giriş puanı sıfır olan veriler silinmemiştir. Çünkü bu veri setindeki öğrencilerden hala okuldan mezun olmayan öğrenciler bulunmaktadır. Ayrıca meslek liselerinden mezun olan öğrencilerin Meslek Yüksek Okullarına sınavsız geçiş hakkı olduğu için "giriş puanı" niteliğine sıfır girilen öğrencilerin bilgileri silinmemiştir. Diğer alanlarında alabilecekleri değerler göz önünde bulundurularak yine filtre yöntemi uygulanarak girilen değerlerin uygunluğu kontrol edilmiştir.

Tablo 3.1: Çalışmada kullanılan nitelikler ve alabileceği değerler

Giriş Yılı	2009-2012
Programı	Bilgisayar Programcılığı, Bilgisayar Programcılığı (i.ö.), Elektrik, Elektrik (i.ö.), Elektrik Teknolojisi, Elektrik Teknolojisi (i.ö.), Geleneksel El Sanatları(i.ö.), Kimya Teknolojisi, Kimya Teknolojisi(i.ö.), Makine, Makine(i.ö.), Mobilya ve Dekorasyon, Mobilya ve Dekorasyon(i.ö.), Otomotiv Teknolojisi(i.ö.), Tekstil Teknolojisi, Tekstil Teknolojisi(i.ö.), Turizm ve Otel İşletmeciliği, Turizm ve Otel İşletmeciliği(i.ö.), Turizm ve Seyahat Hizmetleri
Giriş Puanı	0-400
Akademik Ortalaması	0-4
Durumu	Aktif(AÖSA), Aktif(NÖSA), İzinli, Eksik evrak nedeniyle kaydı silindi, Kayıt yenilememe nedeniyle kaydı silindi, Kendi isteğiyle ayrıldı, Mezun, Vefat nedeniyle kaydı silindi, Yatay geçiş sebebiyle kaydı silindi
Mezuniyet yılı	Null, 2011-2014
Anne Sağ mı?	Evet, Hayır
Annenin eğitim durumu	İlköğretim, Ortaöğretim, Yükseköğretim
Baba Sağ mı?	Evet, Hayır
Babanın eğitim durumu	İlköğretim, Ortaöğretim, Yükseköğretim
Dershaneye gittiniz mi?	Evet, Hayır
Lise diploma notu	0-100
Okuduğunuz Lisenin Türü	Fen Lisesi, Anadolu Lisesi, Normal Lise, Meslek Lisesi, İmam Hatip Lisesi, Diğer

3.4 Veri Dönüştürme

Veriler yapılacak olan çalışmaya uygun hale getirmek için bazı veriler üzerinde verilerin niteliğini değiştiren dönüşümler yapılmıştır. Meslek liselerinden Meslek Yüksek Okuluna sınavsız geçiş yapan öğrencilerin "Giriş Puanı" niteliğine sıfır girilmiş, sınavla gelen öğrencilerin bu niteliğe sınavda aldıkları puanlar yazılmıştı. Çalışmada sınıflandırma tekniklerini kullanılacağı ve bu iki öğrenci grubunu ayırt etmek için "Giriş Puanı" niteliği sıfırdan farklı olan yani belli bir sınav puanı ile giriş yapan öğrencilerin giriş puanları 1 rakamına dönüştürüldü. Bunun içinde excel dosyasında visual basic kodları kullanıldı. Kullanılan kodlar aşağıda gösterilmiştir.

```
Sub gelen()
sonHucre = Worksheets("Sayfa2").Cells(Rows.Count, "D").End(xlUp).Row
For i = 2 To sonHucre
    If Worksheets("Sayfa2").Range("D" & i).Value <> 0 Then
        Worksheets("Sayfa2").Range("D" & i).Value = 1
```

```
    End If
Next i
End Sub
```

Sınıflama tekniklerinin birçoğu numeric değerler yerine nominal değerler kullanmaktadırlar. Bu çalışmada sınıflandırma tekniklerini kullanılacağından bazı sayısal nitelikler belli aralıklara dönüştürüldü. Bu amaçla doğrultusunda veri setindeki "Akademik Ortalama" ve "Lise Diploma Notu" nitelikleri numeric değerlerden nominal değerlere dönüştürüldü. "Lise Diploma Notu" niteliği nominal değerlere dönüştürülürken lise not sistemi göz önünde bulundurulmuştur. Lise diploma notu 50'nin altında kalanlar mezun olamadığı için 50'nin altında kalanlar "Başarısız" olarak değerlendirilmiş ve silinmiştir. Diploma notu 50 ile 70 arasında olanlar "Orta", 70 ile 85 arasında olanlar "iyi", 85 ve üzeri olanlar "Çok iyi" olarak veri setinde dönüştürülmüştür. Bu dönüşüm için Excel de visual basic scriptleri kullanılmıştır. Kullanılan script kodları aşağıdaki gösterilmiştir.

```
Sub LiseNotu()
sonHucre = Worksheets("Sayfa2").Cells(Rows.Count, "P").End(xlUp).Row
For i = 2 To sonHucre
    If Worksheets("Sayfa2").Range("P" & i).Value < 50 Then
        Worksheets("Sayfa2").Range("P" & i).Value = "Basarisiz"
    ElseIf Worksheets("Sayfa2").Range("P" & i).Value >= 50 And
Worksheets("Sayfa2").Range("P" & i).Value < 70 Then
        Worksheets("Sayfa2").Range("P" & i).Value = "Orta"
    ElseIf Worksheets("Sayfa2").Range("P" & i).Value >= 70 And
Worksheets("Sayfa2").Range("P" & i).Value < 85 Then
        Worksheets("Sayfa2").Range("P" & i).Value = "iyi"
    Else
        Worksheets("Sayfa2").Range("P" & i).Value = "Cok_iyi"
    End If
Next i
End Sub
```

"Akademik Ortalama" niteliğinde yapılan veri dönüşümleri Pamukkale Üniversitesi'nin Akademik Not Sistemine göre yapılmıştır. Akademik ortalaması

1’in altında kalanlar “Başarısız”, 1 ile 1.8 arasında olanlar “Koşullu geçer”, 1.8 ile 2.7 arasında olanlar “Orta”, 2.7 ile 3.7 arasında olanlar “iyi”, 3.7 ve üzerinde olanlar “Çok iyi ” olarak veri setinde dönüştürülmüştür. Bu dönüşüm için Excel de visual basic scriptleri kullanılmıştır. Kullanılan script kodları aşağıdaki gösterilmiştir.

```
Sub AkademikOrtalama()

sonHucre = Worksheets("Sayfa2").Cells(Rows.Count, "E").End(xlUp).Row

For i = 2 To sonHucre

    If Worksheets("Sayfa2").Range("E" & i).Value < 1 Then

        Worksheets("Sayfa2").Range("E" & i).Value = "Basarisiz"

    ElseIf Worksheets("Sayfa2").Range("E" & i).Value >= 1 And
Worksheets("Sayfa2").Range("E" & i).Value < 1.8 Then

        Worksheets("Sayfa2").Range("E" & i).Value = "KosulluGecer"

    ElseIf Worksheets("Sayfa2").Range("E" & i).Value >= 1.8 And
Worksheets("Sayfa2").Range("E" & i).Value < 2.7 Then

        Worksheets("Sayfa2").Range("E" & i).Value = "Orta"

    ElseIf Worksheets("Sayfa2").Range("E" & i).Value >= 2.7 And
Worksheets("Sayfa2").Range("E" & i).Value < 3.7 Then

        Worksheets("Sayfa2").Range("E" & i).Value = "iyi"

    Else

        Worksheets("Sayfa2").Range("E" & i).Value = "CokIyi"

    End If

Next i End Sub
```

3.5 Modelin Oluşturulması

Bu araştırmanın amacı meslek yüksek okulu öğrencilerinin başarılarını tahmin etmek ve en iyi tahmini yapacak modeli oluşturmaktır. Bu amaç doğrultusunda bu çalışmada sınıflandırma teknikleri kullanılmıştır. Ön işlemeler sonucunda 1387 kayıt üzerinde sınıflama modellerinden karar ağaçları, bayes, K-en yakın komşu, yapay sinir ağları, destek vektör makinesi modellerine ait birer algoritma seçilerek bunların başarıları karşılaştırılmıştır.

Algoritmaları çalıştırırken test yöntemi olarak "10-kat çarpraz doğrulama" metodu kullanılmıştır. Bu yöntemle veri kaynağı 10 bölüme ayrılır ve her bölüm bir kez test kümesi, kalan diğer 9 bölüm öğrenme kümesi olarak kullanılır.

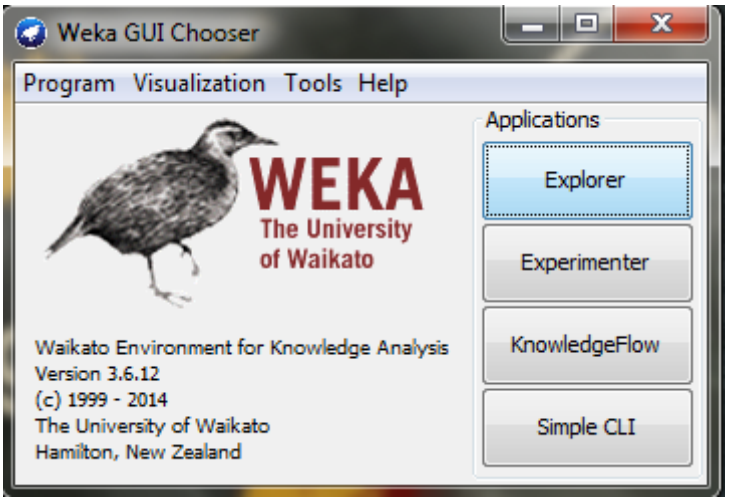
Bu çalışma da başarı tahmini yaparken hedef nitelik olarak öğrencilerin akademik ortalamaları ile mezuniyet yılları seçilmiştir. Her iki ayrı nitelik içinde seçilen tüm sınıflama algoritmaları kullanılmıştır. Öğrencilerin başarılarını daha çok etkileyeceğini düşündüğümüz 13 nitelik üzerinde bu algoritmalar uygulanmıştır. Bu on üç nitelik şunlardır; "Giriş Yılı", "Programı", "Giriş Puanı", "Akademik Ortalaması", "Durumu", "Mezuniyet yılı", "Anne Sağ mı?", "Annenin eğitim durumu", "Baba Sağ mı?", "Babanın eğitim durumu", "Dershaneye gittiniz mi?", "Lise diploma notu", "Okuduğunuz Lisenin Türü"

3.5.1 Uygulamada Kullanılan Veri Madenciliği Aracı

WEKA (Waikato Environment for Knowledge Analysis), Yeni Zelanda Waikato Üniversitesi'nde geliştirilen bir veri madenciliği ve makine öğrenmesi yazılımıdır. WEKA yazılımı nesneye yönelik programlama dillerinden olan Java ile geliştirilmiştir. Java birçok WEKA'nın en güçlü özelliği birçok sınıflandırma tekniğini içermesidir. Weka programın ara yüzü Şekil 3.1'deki gibidir.

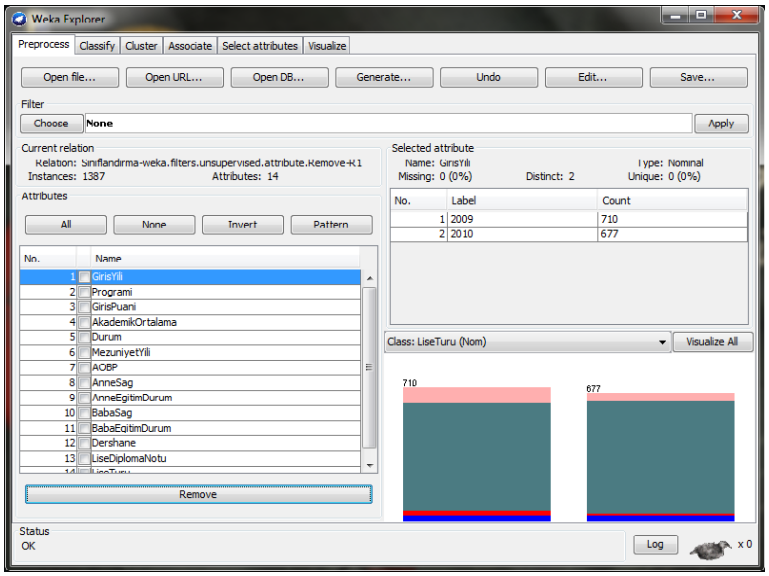
Weka programı 4 temel uygulamayı barındırır, bunlar;

- Explorer
- Experimenter
- KnowledgeFlow
- Simple CLI



Şekil 3.1: Weka programının arayüzü

Explorer, çeşitli veri madenciliği algoritmalarının uygulanabileceği, veriler üzerinde önişlemlerin yapılabileceği, kullanımı kolay bir arayüzdür. Experimenter arayüzü bir probleme birden fazla algoritmanın uygulanarak hangisinin daha iyi sonuç verdiğini analiz etmeye yarayan arayüzdür. Knowledge Flow, grafiksel ikonların taşı-bırak metoduyla taşınarak veri akış diyagramı oluşturarak veri madenciliği algoritmalarının kullanılmasını sağlayan bir arayüzdür. Simple CLI arayüzü ise bir konsol uygulamasıdır. Bu konsoldan Java kodları girilerek veri madenciliği uygulamaları yapılabilmektedir. Bu çalışma da Explorer arayüzü kullanılmıştır. Explorer ara yüzü Şekil 3.2'deki gibidir;



Şekil 3.2: Weka programı Explorer arayüzü

WEKA'da; Preprocess (ön işleme), Classify (sınıflama), Cluster (kümeleme), Associate (birliktelik kuralları), Select Attribute (nitelik seçme) ve Visualize (görselleştirme) panelleri bulunmaktadır. Veri dosyaları ön işleme panelinden yüklenir. Desteklediği veri kaynakları; metin tabanlı arff, csv, c45, libsvm, svmlight, Xarff formatlarıdır. jdbc sürücüsü bulunan veritabanlarına direk bağlantı yapabilir.

3.5.2 Veri Kaynağının Ön İşleme Süreci

Excel dosyası halinde olan verilerin Weka programı tarafından kullanılabilmesi için verilerin Weka programının desteklediği dosya türlerine dönüştürülmesi gerekir. Bunun içinde Excel dosyası halinde bulunan veriler Excel dosyası virgülle ayrılmış .csv uzantılı dosyaya dönüştürülmüştür. Daha sonra bu dosya notepad ile açılarak nitelikler ve bu niteliklerin alabileceği nominal değerler aşağıdaki gibi girilmiştir.

```
@relation Siniflandirma

@Attribute Ogrenci-id string

@Attribute GirisYili  { 2009,2010 }

@Attribute Programi  {
ELEKTRIK,ELEKTRONIK_TEKNOLOJISI_(I.O.),ELEKTRONIK_TEKNOLOJISI,OTOMOTIV_TEKNOLOJISI_
(I.O.),ELEKTRIK_(I.O.),MAKINE,TEKSTIL_TEKNOLOJISI,KIMYA_TEKNOLOJISI_(I.O),BILGISAYA
R_PROGRAMCILIGI,TEKSTIL_TEKNOLOJISI_(I.O.),MAKINE_(I.O.),TURIZM_VE_OTEL_ISLETMECILI
GI,TURIZM_VE_OTEL_ISLETMECILIGI_(I.O.),BILGISAYAR_PROGRAMCILIGI_(I.O.),MOBILYA_VE_D
EKORASYON_(I.O),KIMYA_TEKNOLOJISI,GELENEKSEL_EL_SANATLARI_(I.O.),TURIZM_VE_SEYAHAT_
HIZMETLERI,MOBILYA_VE_DEKORASYON }

@Attribute GirisPuani  { 0,1 }

@Attribute AkademikOrtalama  { KosulluGecer,Basarisiz,Orta,iyi,CokIyi }

@Attribute Durum  {
Aktif_(NOSA),Kendi_istegiyle_Ayrildi,Mezun,Kayit_Yenilememe_Nedeniyle_Kaydi_Silindi
,Aktif_(AOSA),izinli,Yatay_Gecis_Sebebiyle_Kaydi_Silindi,Vefat_Sebebiyle_Kaydi_Sili
ndi,Eksik_Evrak_Sebebiyle_Kaydi_Silindi }

@Attribute MezuniyetYili  { 0,2011,2012,2014,2013 }

@Attribute AOBP real

@Attribute AnneSag {Evet,Hayir}

@Attribute AnneEgitimDurum {ilkOgretim,OrtaOgretim,YuksekOgretim}

@Attribute BabaSag {Evet,Hayir}
```

```
@Attribute BabaEgitimDurum {ilkOgretim,OrtaOgretim,YuksekOgretim}
@Attribute Dershane {Evet,Hayir}
@Attribute LiseDiplomaNotu { Orta,iyi,Cok_iyi }
@Attribute LiseTuru {Anadolu_Lisesi,Diger,Fen_Lisesi,Meslek_Lisesi,Normal_Lise}
@Data
```

Bu değerler girildikten sonra dosya weka programına aktarılabilecek .arff uzantılı dosya olarak kaydedilmiştir. Weka programının Explorer arayüzün de Preprocess panelinden Open file ile programa aktarılacak olan dosya seçilir. Dosya yüklendikten sonra nitelikler üzerinde değişiklik yapılmak istenirse yine Preprocess panelinde bu işlemler yapılabilir. Öğrencilerin id değerleri yapacağımız işlemler için gereksiz bir nitelik olduğu için bu nitelik bu ara yüz üzerinde seçilerek remove komutuyla kaldırılmıştır.

3.5.3 Model Başarımını Denetleme

Model başarımını değerlendirirken kullanılan temel kavramlar doğruluk oranı, kesinlik, duyarlılık, F-ölçütü ve kapa istatistiğidir.

Test sonucunda sınıflandırma modelinin başarısı hata matrisindeki doğru sınıflandırılan değişkenler ile yanlış sınıflandırılan değişkenlerin sayısıyla hesaplanır. Hata matrisinde satırlar test kümesindeki örneklere ait gerçek sayıları, kolonlar ise modelin tahminlemesini ifade eder. Tablo 3.2'de iki sınıflı bir veri kümesinde oluşturulmuş bir modelin hata matrisi verilmiştir. n x n boyutlarındaki bir hata matrisinde ana köşegen doğru tahminlenmiş örnek sayılarını; ana köşegen dışında kalan matris elemanları ise hatalı sonuçları verir. TP (True Pozitif) ve TN(True Negatif) değerleri doğru sınıflandırılmış örnek sayısıdır. False Pozitif (FP), aslında 0 (negatif) sınıfındayken 1 (pozitif) olarak tahminlenmiş örneklerin sayısıdır. False Negative (FN) ise 1 (pozitif) sınıfındayken 0 (negatif) olarak tahminlenmiş örneklerin sayısını ifade eder (Şık 2014).

Tablo 3.2: Hata matrisi

		Tahmin Edilen Sınıf	
		Sınıf=1	Sınıf=0
Gerçek Sınıf	Sınıf=1	TP	FP
	Sınıf=0	FN	TN

Doğruluk Oranı: Doğru sınıflandırılmış örnek sayısının (TP +TN), toplam örnek sayısına (TP+TN+FP+FN) oranıdır (Şık 2014).

$$Doğruluk = \frac{TN + TP}{FN + TN + TP + FP} \tag{3.1}$$

Kesinlik(Precision): Doğru sınıflandırılmış pozitif örneklem sayısının sınıfı pozitif olarak öngörülmüş toplam örneklem sayısına oranıdır. [0,1] Aralığında bir değer alır (Şık 2014).

$$Kesinlik = \frac{TP}{TP + FN} \tag{3.2}$$

Duyarlılık (Recall): Doğru sınıflandırılmış pozitif örneklem sayısının gerçek sınıfı pozitif olan tüm örneklemlerin sayısına oranıdır. Duyarlılık, gerçek pozitiflik oranı (True Positive Rate) olarak da adlandırılır. [0,1] Aralığında bir değer alır (Şık 2014).

$$Duyarlılık = \frac{TP}{TP + FP} \tag{3.3}$$

F-Ölçütü: Çoğunlukla kesinlik ve duyarlılık arasında zıt bir ilişki vardır ki birinin değerini arttırmak diğerinin değerini düşürebilir. Bu nedenle daha kesin ve duyarlı sonuçlar elde etmek için her iki ölçütün harmonik ortalaması olan F-ölçütü kullanılır (Şık 2014).

$$F - \text{Ölçütü} = \frac{2 \times Duyarlılık \times Kesinlik}{Duyarlılık + Kesinlik} \tag{3.4}$$

Kappa İstatistiği: Kappa istatistiği, yapılan tahminin doğruluk ölçüsüdür. Kappa sonuçları [0,1] aralığında değişir. Kappa değeri 0,4 ile 0,6 arasında ise orta

seviyede bir uyum vardır. 0,6 ile 0,8 arasında ise iyi seviyede bir uyum vardır. 0,8 ile 1 arasında ise çok iyi seviyede bir uyum vardır (Landis ve diğ. 1977).

3.5.4 Hedef Nitelik Olarak "Akademik Ortalama"

3.5.4.1 Karar Ağacı Modelinin Başarım Ölçütü

Weka programında model oluşturulurken kullanılabilecek pek çok karar ağacı algoritması mevcuttur. Bu çalışmada C4.5 karar ağacının Weka tarafından Java'da kodlanan J4.8 algoritması kullanılmıştır. J4.8 algoritmasından alınan sonuç Tablo 3.3'de verilmiştir.

Tablo 3.3: Akademik ortalamaya göre Karar Ağacı sınıflandırma modelinin başarım ölçütü

```
Doğru sınıflandırılmış örnekler         826               59.553  %
Hatalı Sınıflandırılan Örnekler         561               40.447  %
Kappa istatistiği                         0.440
Ortalama mutlak hata                      0.195
Ortalama karesel hatanın karekökü         0.335
Göreceli mutlak hata                     67.609 %
Göreceli karesel hatanın karekökü        88.282 %
Toplam örnek sayısı                    1387

=== Sınıflara Göre Ayrıntılı Doğruluk ===

  TP Oran    FP Oran   Duyarlılık  Kesinlik  F-Ölçütü   ROC Alanı  Sınıf
     0.214     0.096      0.285     0.214     0.245      0.749 KosulluGecer
     0.649     0.170      0.526     0.649     0.581      0.838    Basarisiz
     0.633     0.190      0.683     0.633     0.657      0.756    Orta
     0.752     0.097      0.689     0.752     0.719      0.881    iyi
     0         0          0         0         0          0.866    CokIyi
Ort. 0.596     0.149      0.584     0.596     0.587      0.802

=== Hata Matrisi===

   a   b   c   d   e   <-- Sınıflandırılmış
  45 120  45   0   0 |   a = KosulluGecer
  65 203  45   0   0 |   b = Basarisiz
  45  59 345  96   0 |   c = Orta
   3   4  70 233   0 |   d = iyi
   0   0   0   9   0 |   e = CokIyi
```

J4.8 algoritması verileri %59.553 doğruluk oranı ile sınıflandırmıştır. 1387 veriden 826'sını doğru, 561'ini yanlış sınıflandırmıştır.

3.5.4.2 Bayes Sınıflandırma Modelinin Başarım Ölçütü

Bayes sınıflandırma için Weka'da var olan NaiveBayes algoritması seçilmiştir. NaiveBayes algoritması, Bayes teoremine dayanan standart olasılıklı sınıflandırma yöntemidir. NaiveBayes algoritmasından alınan sonuç Tablo 3.4'de gösterilmiştir.

Tablo 3.4: Akademik ortalamaya göre NaiveBayes sınıflandırma modelinin başarım ölçütü

```
Doğru sınıflandırılmış örnekler         787               56.741 %
Hatalı Sınıflandırılan Örnekler         600               43.258 %
Kappa istatistiği                         0.419
Ortalama mutlak hata                      0.194
Ortalama karesel hatanın karekökü         0.332
Göreceli mutlak hata                     67.491 %
Göreceli karesel hatanın karekökü        87.574 %
Toplam örnek sayısı                    1387

=== Sınıflara Göre Ayrıntılı Doğruluk ===

       TP Oran   FP Oran  Duyarlılık  Kesinlik  F-Ölçütü   ROC Alanı  Sınıf
        0.295     0.103      0.339     0.295     0.316      0.809    KosulluGecer
        0.824     0.232      0.509     0.824     0.629      0.876    Basarisiz
        0.450     0.102      0.740     0.450     0.559      0.770    Orta
        0.716     0.133      0.608     0.716     0.658      0.898    iyi
        0         0.001      0         0         0          0.838    CokIyi
Ort.    0.567     0.138      0.593     0.567     0.557      0.829

=== Hata Matrisi===

   a   b   c   d   e   <-- Sınıflandırılmış
  62 143   5   0   0 |   a = KosulluGecer
  53 258   1   0   1 |   b = Basarisiz
  63 103 245 134   0 |   c = Orta
   5   3  80 222   0 |   d = iyi
   0   0   0   9   0 |   e = CokIyi
```

NaiveBayes algoritması verileri %56.741 doğruluk oranı ile sınıflandırmıştır. 1387 veriden 787'sini doğru, 600'ünü yanlış sınıflandırmıştır.

3.5.4.3 K-En Yakın Komşu Modelinin Başarım Ölçütü

K-en yakın komşu sınıflandırması için Weka'da var olan IBk algoritması seçilmiştir. Bu algoritma sınıflandırma yaparken veritabanındaki her bir kaydın diğer kayıtlarla olan uzaklığını hesaplar. Ancak, bir kayıt için diğer kayıtlardan sadece k adedi göz önüne alır. Algoritmada k değeri önceden seçilir. Bu çalışmada da k değeri 3 olarak seçilmiştir. IBk algoritmasından alınan sonuç Tablo 3.5'de gösterilmiştir.

Tablo 3.5: Akademik ortalamaya göre K-En Yakın Komşu modelinin başarım ölçütü

```
Doğru sınıflandırılmış örnekler         777               56.020 %
Hatalı Sınıflandırılan Örnekler         610               43.979 %
Kappa istatistiği                         0.393
Ortalama mutlak hata                      0.200
Ortalama karesel hatanın karekökü         0.333
Göreceli mutlak hata                     69.323 %
Göreceli karesel hatanın karekökü        87.664 %
Toplam örnek sayısı                    1387

=== Sınıflara Göre Ayrıntılı Doğruluk ===

       TP Oran   FP Oran  Duyarlılık  Kesinlik  F-Ölçütü   ROC Alanı  Sınıf
        0.300     0.123      0.303     0.300     0.301      0.781    KosulluGecer
        0.661     0.170      0.531     0.661     0.589      0.867    Basarisiz
        0.589     0.213      0.642     0.589     0.614      0.752    Orta
        0.600     0.096      0.644     0.600     0.621      0.898    iyi
        0         0          0         0         0          0.635    CokIyi
Ort.    0.56      0.162      0.562     0.56      0.559      0.814

=== Hata Matrisi===

   a   b   c   d   e   <-- Sınıflandırılmış
  63 113  33   1   0 |   a = KosulluGecer
  80 207  24   2   0 |   b = Basarisiz
  62  68 321  94   0 |   c = Orta
   3   2 119 186   0 |   d = iyi
   0   0   3   6   0 |   e = CokIyi
```

IBk algoritması verileri %56.020 doğruluk oranı ile sınıflandırmıştır. 1387 veriden 777'sini doğru, 610'unu yanlış sınıflandırmıştır.

3.5.4.4 Yapay Sinir Ağları Sınıflandırma Modelinin Başarım Ölçütü

Yapay sinir ağları sınıflandırma için Weka'da yer alan MultiLayerPerceptron algoritması kullanılmıştır. Bu yapay sinir ağı algoritmasında istenildiği kadar ara katman oluşturulabiliyor. Bu çalışmada giriş katmanı, bir ara katman birde çıkış katmanı kullanılmıştır. MultiLayerPerceptron algoritmasından alınan sonuçlar Tablo 3.6'da gösterilmiştir.

MultiLayerPerceptron algoritması verileri %55.299 doğruluk oranı ile sınıflandırmıştır. 1387 veriden 767'sini doğru, 620'sini yanlış sınıflandırmıştır.

3.5.4.5 Destek Vektör Makinesi Sınıflandırma Modelinin Başarım Ölçütü

Bu yöntem veriyi birbirinden ayırmak için en uygun fonksiyonun tahmin edilmesi esasına dayanır. Destek vektör makinesi sınıflandırması için Weka'da var

olan John Platt'in geliştirdiği sıralı minimal optimizasyon olan SMO algoritması kullanılmıştır. SMO algoritmasından alınan sonuç Tablo 3.7'de gösterilmiştir.

Tablo 3.6: Akademik ortalamaya göre Yapay Sinir Ağları sınıflandırma modelinin başarım ölçütü

```
Doğru sınıflandırılmış örnekler         767               55.299 %
Hatalı Sınıflandırılan Örnekler         620               44.700 %
Kappa istatistiği                         0.382
Ortalama mutlak hata                      0.186
Ortalama karesel hatanın karekökü         0.381
Göreceli mutlak hata                     64.463 %
Göreceli karesel hatanın karekökü       100.385 %
Toplam örnek sayısı                    1387

=== Sınıflara Göre Ayrıntılı Doğruluk ===

       TP Oran   FP Oran  Duyarlılık  Kesinlik  F-Ölçütü   ROC Alanı  Sınıf
        0.333     0.127      0.318     0.333     0.326      0.773    KosulluGecer
        0.572     0.133      0.556     0.572     0.564      0.841    Basarisiz
        0.593     0.241      0.614     0.593     0.603      0.723    Orta
        0.629     0.110      0.621     0.629     0.625      0.866    iyi
        0         0.004      0         0         0          0.769    CokIyi
Ort.    0.553     0.169      0.554     0.553     0.553      0.790

=== Hata Matrisi===

   a   b   c   d   e   <-- Sınıflandırılmış
  70  88  50   2   0 |   a = KosulluGecer
  88 179  44   2   0 |   b = Basarisiz
  58  53 323 110   1 |   c = Orta
   4   2 105 195   4 |   d = iyi
   0   0   4   5   0 |   e = CokIyi
```

Tablo 3.7: Akademik ortalamaya göre Destek Vektör Makinesi sınıflandırma modelinin başarım ölçütü

```
Doğru sınıflandırılmış örnekler         832               59.985 %
Hatalı Sınıflandırılan Örnekler         555               40.014 %
Kappa istatistiği                         0.450
Ortalama mutlak hata                      0.261
Ortalama karesel hatanın karekökü         0.348
Göreceli mutlak hata                     90.522 %
Göreceli karesel hatanın karekökü        91.715  %
Toplam örnek sayısı                    1387

=== Sınıflara Göre Ayrıntılı Doğruluk ===

       TP Oran   FP Oran  Duyarlılık  Kesinlik  F-Ölçütü   ROC Alanı  Sınıf
        0.310     0.093      0.371     0.310     0.338      0.802    KosulluGecer
        0.748     0.183      0.543     0.748     0.629      0.861    Basarisiz
        0.589     0.165      0.698     0.589     0.639      0.739    Orta
        0.684     0.101      0.660     0.684     0.672      0.882    iyi
        0         0          0         0         0          0.752    CokIyi
Ort.    0.600     0.143      0.601     0.600     0.594      0.808

=== Hata Matrisi===

   a   b   c   d   e   <-- Sınıflandırılmış
  65 118  27   0   0 |   a = KosulluGecer
  57 234  22   0   0 |   b = Basarisiz
  48  76 321 100   0 |   c = Orta
   5   3  90 212   0 |   d = iyi
   0   0   0   9   0 |   e = CokIyi
```

SMO algoritması verileri %59.985 doğruluk oranı ile sınıflandırmıştır. 1387 veriden 832'sini doğru, 555'ini yanlış sınıflandırmıştır.

3.5.4.6 Oluşturulan Modellerin Karşılaştırılması

Öğrencilerin akademik ortalamalarını etkileyeceğini düşündüğümüz "Giriş Yılı", "Programı", "Giriş Puanı", "Akademik Ortalaması", "Durumu", "Mezuniyet yılı", "Anne Sağ mı?", "Annenin eğitim durumu", "Baba Sağ mı?", "Babanın eğitim durumu", "Dershaneye gittiniz mi?", "Lise diploma notu", "Okuduğunuz Lisenin Türü"'den oluşan onüç özellik seçildi ve bu özellikler üzerinde J4.8, NaiveBayes, IBk, SMO ve MultiLayerPerceptron algoritmaları uygulandı. Her algoritma için oluşmuş olan modele ait test istatistiği bir önce ki bölümde verilmiştir. Algoritmalar arasında karşılaştırma yapabilmek için her modele ait doğru sınıflandırma yüzdesi, Kappa istatistiği, duyarlılık(precision), kesinlik(recall) ve F-Ölçütü ölçüt değerleri alınmıştır. Bu değerler tablo 3.8'de gösterilmiştir.

Tablo 3.8: Akademik ortalamaya göre oluşturulan modellerin karşılaştırılması

Algoritmalar	Doğru Sınıflandırma Yüzdesi	Kappa İstatistiği	Duyarlılık (Precision)	Kesinlik (Recall)	F-Ölçütü
J4.8	%59.55	0.440	0.584	0.596	0.587
NaiveBayes	%56.74	0.419	0.593	0.567	0.557
IBk	%56.02	0.393	0.562	0.560	0.559
SMO	%59.98	0.450	0.601	0.600	0.594
MultiLayerPerceptron	%55.29	0.382	0.554	0.553	0.553

Algoritmaların doğru sınıflandırma yüzdelerine baktığımızda en iyi sınıflandırma modelinin J4.8 algoritması ile SMO algoritması olduğunu görüyoruz. Bu iki algoritmanın kappa istatistiklerine baktığımızda çok az bir farkla SMO algoritmasının daha iyi sonuç verdiğini görüyoruz. SMO algoritmasının duyarlılık, kesinlik ve f-ölçütü değerlerinin de J4.8 algoritmasından az bir farkla yüksek olduğu görülmektedir. Tüm ölçüt değerlerinde en düşük değerlere sahip olan algoritmanın MultiLayerPerceptron algoritması olduğu açık bir şekilde görülüyor.

3.5.5 Hedef Nitelik Olarak Mezuniyet Yılı

Meslek Yüksek Okullarında eğitim süresi 2 yıldır. Öğrencilerin bazıları, tüm derslerde başarılı olamayabilirler ve böyle durumlarda öğrencilerin okullarından mezun olma süreleri 2 yılın üzerine çıkabilir. Aslında bu da öğrencilerin başarı

durumlarını gösteren diğer bir niteliktir. Bu sebeple de bu çalışma da ikinci bir hedef nitelik olarak öğrencilerin mezun oldukları yıllar seçilmiştir.

3.5.5.1 Karar Ağacı Modelinin Başarım Ölçütü

Weka programında model oluşturulurken kullanılabilecek pek çok karar ağacı algoritması mevcuttur. Bu çalışmada C4.5 karar ağacının Weka tarafından Java'da kodlanan J4.8 algoritması kullanılmıştır. J4.8 algoritmasından alınan sonuç Tablo 3.9' da verilmiştir.

Tablo 3.9: Mezuniyet yılına göre Karar Ağacı sınıflandırma modelinin başarım ölçütü

```
Doğru sınıflandırılmış örnekler        1128               81.326 %
Hatalı Sınıflandırılan Örnekler         259               18.673 %
Kappa istatistiği                         0.712
Ortalama mutlak hata                      0.098
Ortalama karesel hatanın karekökü         0.231
Göreceli mutlak hata                     37.069 %
Göreceli karesel hatanın karekökü        63.497 %
Toplam örnek sayısı                    1387

=== Sınıflara Göre Ayrıntılı Doğruluk ===

       TP Oran   FP Oran  Duyarlılık  Kesinlik F-Ölçütü  ROC Alanı  Class
        1         0          1         1         1          1        0
        0.834     0.048      0.744     0.834     0.787      0.952    2011
        0.832     0.166      0.596     0.832     0.695      0.878    2012
        0.010     0.015      0.050     0.010     0.016      0.815    2013
        0         0.004      0         0         0          0.773    2014
Ort.    0.813     0.046      0.750     0.813     0.776      0.940

=== Hata Matrisi===

   a   b   c   d   e   <-- Sınıflandırılmış
 698   0   0   0   0 |   a = 0
   0 166  27   5   1 |   b = 2011
   0  42 263   9   2 |   c = 2012
   0   7  95   1   2 |   d = 2013
   0   8  56   5   0 |   e = 2014
```

J4.8 algoritması verileri %81.326 doğruluk oranı ile sınıflandırmıştır. 1387 veriden 1128'ini doğru, 259'unu yanlış sınıflandırmıştır. Hata matrisine baktığımızda mezun olamayanların tamamını doğru sınıflandırmıştır. Ancak 2013 ve 2014 yılında mezun olan öğrencileri doğru sınıflandıramamıştır.

3.5.5.2 Bayes Sınıflandırma Modelinin Başarım Ölçütü

Bayes sınıflandırma için Weka'da var olan NaiveBayes algoritması seçilmiştir. NaiveBayes algoritması, Bayes teoremine dayanan standart olasılıklı

sınıflandırma yöntemidir. NaiveBayes algoritmasından alınan sonuç Tablo 3.10'da gösterilmiştir.

Tablo 3.10: Mezuniyet yılına göre NaiveBayes sınıflandırma modelinin başarım ölçütü

```
Doğru sınıflandırılmış örnekler        1125               81.110 %
Hatalı Sınıflandırılan Örnekler         262               18.890 %
Kappa istatistiği                         0.712
Ortalama mutlak hata                      0.100
Ortalama karesel hatanın karekökü         0.224
Göreceli mutlak hata                     37.486 %
Göreceli karesel hatanın karekökü        61.458 %
Toplam örnek sayısı                    1387

=== Sınıflara Göre Ayrıntılı Doğruluk ===

       TP Oran   FP Oran  Duyarlılık  Kesinlik  F-Ölçütü  ROC Alanı  Class
         0.993     0          1         0.993     0.996      1         0
         0.769     0.041      0.757     0.769     0.763      0.972     2011
         0.810     0.144      0.624     0.810     0.705      0.919     2012
         0.162     0.025      0.347     0.162     0.221      0.878     2013
         0.087     0.020      0.182     0.087     0.118      0.889     2014
 Ort.    0.811     0.042      0.789     0.811     0.794      0.963

=== Hata Matrisi===

   a   b   c   d   e   <-- Sınıflandırılmış
 693   3   0   2   0 |   a = 0
   0 153  43   2   1 |   b = 2011
   0  34 256  13  13 |   c = 2012
   0   5  70  17  13 |   d = 2013
   0   7  41  15   6 |   e = 2014
```

NaiveBayes algoritması verileri %81.110 doğruluk oranı ile sınıflandırmıştır. 1387 veriden 1125'ini doğru, 262'sini yanlış sınıflandırmıştır. Hata matrisine baktığımızda mezun olamayanların tamamını doğru sınıflandırmıştır. 2013 ve 2014 yılında mezun olan öğrencileri sınıflandırırken yanlış sınıflandırma oranı doğru sınıflandırma oranından yüksek çıkmıştır.

3.5.5.3 K-En Yakın Komşu Algoritması Modelinin Başarım Ölçütü

K-en yakın komşu sınıflandırması için Weka'da var olan IBk algoritması seçilmiştir. Bu çalışmada da k değeri 3 olarak seçilmiştir. IBk algoritmasından alınan sonuç Tablo 3.11'de gösterilmiştir.

IBk algoritması verileri %79.019 doğruluk oranı ile sınıflandırmıştır. 1387 veriden 1096'sını doğru, 291'in yanlış sınıflandırmıştır. Hata matrisine baktığımızda 2013 ve 2014 yılında mezun olan öğrencileri sınıflandırırken yanlış sınıflandırma oranı doğru sınıflandırma oranından yüksek çıkmıştır.

Tablo 3.11: Mezuniyet yılına göre K-En Yakın Komşu sınıflandırma modelinin başarım ölçütü

```
Doğru sınıflandırılmış örnekler        1096               79.019 %
Hatalı Sınıflandırılan Örnekler         291               20.980 %
Kappa istatistiği                         0.680
Ortalama mutlak hata                      0.115
Ortalama karesel hatanın karekökü         0.243
Göreceli mutlak hata                     43.179 %
Göreceli karesel hatanın karekökü        66.667 %
Toplam örnek sayısı                    1387

=== Sınıflara Göre Ayrıntılı Doğruluk ===

        TP Oran   FP Oran  Duyarlılık  Kesinlik  F-Ölçütü  ROC Alanı  Class
         0.979     0.003      0.997     0.979     0.988      0.999    0
         0.809     0.070      0.660     0.809     0.727      0.961    2011
         0.769     0.159      0.588     0.769     0.667      0.895    2012
         0.048     0.020      0.167     0.048     0.074      0.781    2013
         0.058     0.008      0.267     0.058     0.095      0.773    2014
 Ort.    0.790     0.050      0.756     0.790     0.764      0.942

=== Hata Matrisi===

   a   b   c   d   e   <-- Sınıflandırılmış
 683   7   6   2   0 |   a = 0
   1 161  36   1   0 |   b = 2011
   0  56 243  13   4 |   c = 2012
   1  13  79   5   7 |   d = 2013
   0   7  49   9   4 |   e = 2014
```

3.5.5.4 Yapay Sinir Ağları Sınıflandırma Modelinin Başarım Ölçütü

Yapay sinir ağları sınıflandırma için Weka'da yer alan MultiLayerPerceptron algoritması kullanılmıştır. Bu çalışmada giriş katmanı, bir ara katman birde çıkış katmanı kullanılmıştır. MultiLayerPerceptron algoritmasından alınan sonuçlar Tablo 3.12'de gösterilmiştir.

MultiLayerPerceptron algoritması verileri %78.226 doğruluk oranı ile sınıflandırmıştır. 1387 veriden 1085'ini doğru, 302'sini yanlış sınıflandırmıştır. Hata matrisine baktığımızda 2013 ve 2014 yılında mezun olan öğrencileri sınıflandırırken yanlış sınıflandırma oranı doğru sınıflandırma oranından yüksek çıkmıştır.

3.5.5.5 Destek Vektör Makinesi Sınıflandırma Modelinin Başarım Ölçütü

Destek vektör makinesi sınıflandırması için Weka'da var olan John Platt'in geliştirdiği sıralı minimal optimizasyon olan SMO algoritması kullanılmıştır. SMO algoritmasından alınan sonuç Tablo 3.13'de gösterilmiştir.

SMO algoritması verileri %80.605 doğruluk oranı ile sınıflandırmıştır. 1387 veriden 1118'ini doğru, 269'unu yanlış sınıflandırmıştır. Hata matrisine baktığımızda mezun olamayanların tamamını doğru sınıflandırmıştır. 2013 ve 2014 yılında mezun olan öğrencileri sınıflandırırken yanlış sınıflandırma oranı doğru sınıflandırma oranından yüksek çıkmıştır.

Tablo 3.12: Mezuniyet yılına göre Yapay Sinir Ağları sınıflandırma modelinin başarım ölçütü

```
Doğru sınıflandırılmış örnekler        1085               78.226 %
Hatalı Sınıflandırılan Örnekler         302               21.773 %
Kappa istatistiği                         0.670
Ortalama mutlak hata                      0.092
Ortalama karesel hatanın karekökü         0.269
Göreceli mutlak hata                     34.806 %
Göreceli karesel hatanın karekökü        73.837 %
Toplam örnek sayısı                    1387

=== Sınıflara Göre Ayrıntılı Doğruluk ===

        TP Oran   FP Oran   Duyarlılık  Kesinlik  F-Ölçütü  ROC Alanı  Class
         0.999     0.003      0.997     0.999     0.998      1         0
         0.709     0.046      0.719     0.709     0.714      0.958     2011
         0.680     0.129      0.609     0.680     0.643      0.887     2012
         0.190     0.055      0.220     0.190     0.204      0.802     2013
         0.174     0.027      0.250     0.174     0.205      0.795     2014
 Ort.    0.782     0.043      0.773     0.782     0.777      0.943

=== Hata Matrisi===

   a   b   c   d   e   <-- Sınıflandırılmış
 697   0   0   1   0 |   a = 0
   0 141  45   8   5 |   b = 2011
   2  42 215  41  16 |   c = 2012
   0   8  62  20  15 |   d = 2013
   0   5  31  21  12 |   e = 2014
```

Tablo 3.13: Mezuniyet yılına göre Destek Vektör Makinesi sınıflandırma modelinin başarım ölçütü

```
Doğru sınıflandırılmış örnekler        1118               80.605 %
Hatalı Sınıflandırılan Örnekler         269               19.394 %
Kappa istatistiği                         0.701
Ortalama mutlak hata                      0.251
Ortalama karesel hatanın karekökü         0.333
Göreceli mutlak hata                     94.293 %
Göreceli karesel hatanın karekökü        91.397 %
Toplam örnek sayısı                    1387

=== Sınıflara Göre Ayrıntılı Doğruluk ===

        TP Oran  FP Oran  Duyarlılık  Kesinlik  F-Ölçütü   ROC Alanı  Class
         1         0         1         1         1          1         0
         0.754     0.030     0.806     0.754     0.779      0.932     2011
         0.829     0.185     0.570     0.829     0.675      0.843     2012
         0.048     0.016     0.200     0.048     0.077      0.771     2013
         0.043     0.011     0.167     0.043     0.069      0.770     2014
 Ort.    0.806     0.048     0.772     0.806     0.778      0.926

=== Hata Matrisi===

   a   b   c   d   e   <-- Sınıflandırılmış
 698   0   0   0   0 |   a = 0
   0 150  48   1   0 |   b = 2011
   0  33 262  12   9 |   c = 2012
   0   1  93   5   6 |   d = 2013
   0   2  57   7   3 |   e = 2014
```

3.5.5.6 Mezuniyet Yılına Göre Oluşturulan Modellerin Karşılaştırılması

Öğrencilerin başarılarının bir göstergesi olan mezun olma sürelerini etkilediğini düşündüğümüz "Giriş Yılı", "Programı", "Giriş Puanı", "Akademik Ortalaması", "Durumu", "Mezuniyet yılı", "Anne Sağ mı?", "Annenin eğitim durumu", "Baba Sağ mı?", "Babanın eğitim durumu", "Dershaneye gittiniz mi?", "Lise diploma notu", "Okuduğunuz Lisenin Türü"'den oluşan onüç özellik seçildi ve bu özellikler üzerinde J4.8, NaiveBayes, IBk, SMO ve MultiLayerPerceptron algoritmaları ile analiz edilerek her algoritma için oluşmuş olan modele ait test istatistiği bir önce ki bölümde verilmiştir. Algoritmalar arasında karşılaştırma yapabilmek için her modele ait doğru sınıflandırma yüzdesi, Kappa istatistiği, duyarlılık(precision), kesinlik(recall) ve F-Ölçütü ölçüt değerleri alınmıştır. Bu değerler aşağıdaki tabloda gösterilmiştir.

Tablo 3.14: Mezuniyet yılına göre oluşturulan modellerin karşılaştırılması

Algoritmalar	Doğru Sınıflandırma Yüzdesi	Kappa İstatistiği	Duyarlılık (Percision)	Kesinlik (Recall)	F-Ölçütü
J4.8	%81.326	0.712	0.750	0.813	0.776
NaiveBayes	%81.110	0.712	0.789	0.811	0.794
IBk	%79.019	0.680	0.756	0.790	0.764
SMO	%80.605	0.701	0.772	0.806	0.778
MultiLayerPerceptron	%78.226	0.670	0.773	0.782	0.777

Algoritmaların doğru sınıflandırma yüzdelerine baktığımızda J4.8 ve NaiveBayes algoritmaları birbirine çok yakın değerler bulmuşlar. Kappa istatistiği tahmin doğruluğunu ölçen bir birimdir. Bu birime baktığımızda en iyi sonucu J4.8 ve NaiveBayes algoritmaları vermektedir ve bu iki algoritmanın kapa istatistiği değerleri birbirine çok yakındır. Bu iki algoritmanın duyarlılıklarına baktığımızda NaiveBayes algoritmasının çok az bir farkla J4.8 algoritmasından daha iyi olduğunu görüyoruz. F-ölçütü duyarlılık ile kesinlik değerlerinin harmonik bir ortalamasından oluşmaktadır. Bu değere baktığımızda NaiveBayes algoritmasını çok az bir farkla J4.8 algoritmasından önde görmekteyiz. Doğru sınıflandırma yüzdesi ve Kappa istatistiğine baktığımızda en kötü sonucu veren algoritmanın MultiLayerPerceptron algoritması olduğu görülmektedir.

4. SONUÇ VE ÖNERİLER

Öğrencilerin akademik başarılarını tahmin etmek oldukça zor bir durumdur. Başarı çok soyut bir kavramdır ve başarıyı veya başarısızlığı etkileyen çok fazla etken vardır. Akademik başarıyı etkileyen nitelikleri bulmak ve bu niteliklerden öğrencinin başarısını tahmin etmeye çalışmak çok fazla bilginin birlikte kullanılmasını gerektirmektedir. Çok fazla veriden bir örüntü çıkarmak için kullanılan en uygun yöntem de veri madenciliğidir. Bu çalışmada Meslek Yüksek Okulu öğrencilerinin başarılarını tahmin etmek için veri madenciliği yöntemleri kullanıldı. Bunun için Pamukkale Üniversitesi Meslek Yüksek Okulu'na 2009-2012 yılları arasında kayıt yaptıran tüm öğrencilere uygulanan 13 sorulu anketten oluşan öğrencilerin akademik bilgileri ve ailevi durum bilgilerinden yararlanıldı. Bu bilgiler doğrultusunda öğrencilerin başarıları tahmin edilmeye çalışıldı. Bu amaçla veri madenciliği yönteminin sınıflandırma teknikleri kullanıldı ve bu tekniklerden öğrencilerin akademik başarılarını en iyi tahmin eden model bulunmaya çalışıldı. Öğrencilerin akademik başarılarını tahmin etmeye çalışılırken iki tane bağımlı değişken kullanıldı. Bunlar öğrencilerin akademik ortalamaları ile mezun oldukları yıllardır.

Öğrencilerin akademik ortalamaları Pamukkale Üniversitesinin not sistemine bağlı kalınarak "Başarısız, Koşullu geçer, orta, iyi ve çok iyi" olmak üzere 5 kısma bölündü. Öğrencileri bu başarı kategorilerden hangisini gerçekleştireceği tahmin edilmeye çalışıldı. Bunun için de 5 tane algoritma kullanıldı. Akademik ortalamaya göre algoritmaların karşılaştırma tablosuna baktığımızda doğru sınıflandırma yüzdelerinin %55 ile %59 arasında olduğu görülmektedir. Bu 5 algoritmanın birbirine yakın sonuçlar verdiği görülmektedir ancak en iyi sonucu veren algoritma SMO algoritması ile destek vektör makinesi tekniğidir.

Portekiz'de yapılan bir çalışmada, ortaöğretim öğrencilerinin matematik ve Portekizce derslerindeki başarıları tahmin edilmeye çalışılmış ve bu çalışmada öğrencilerin daha önce almış olduğu notların başarı tahminini ne kadar etkilediği de araştırılmış. Öğrencilere anket uygulanarak öğrencilerin nüfusa dayalı, sosyal ve okulla ilgili bilgileri toplanmış ve bu bilgilerle beraber öğrencilerin matematik ve

Portekizce derslerine ait notları beş ayrı sınıflandırmaya tabi tutularak bu veriler üzerinde veri madenciliği teknikleri olan karar ağaçları, yapay sinir ağları ve destek vektör makineleri yöntemleri uygulanmış. Bu teknikler başarı tahmininde %43 ile %77 oranları arasında doğru tahminlerde bulunmuş (Cortez ve Silva 2008). Bu çalışmada ve bizim yapmış olduğumuz çalışmada da öğrencilerin notları beş farklı düzeye tabi tutulmaktadır. Öğrencilerin notlarının sınıflandırılma düzeyi arttıkça modellerin tahmin yüzdeleri düşmektedir.

Öğrencilerin akademik başarılarını gösteren diğer bir değişken mezuniyet yılıdır. Çünkü öğrencilerin gösterdikleri başarı veya başarısızlık okulu zamanında bitirip bitirememelerine ya da okuldan ayrılmalarına sebep olmaktadır. Bu doğrultuda bağımlı değişken olarak öğrencilerin mezun oldukları yıllar tahmin edilmeye çalışıldı. Bunun için de aynı 5 algoritma uygulanıp sonuçları incelendiğinde, doğru sınıflandırma yüzdelerinin %78 ile %81 arasında birbirine yakın ve yüksek değerler aldığı görüldü. Öğrencilerin mezuniyet yıllarını en iyi tahmin eden algoritmalar; bir karar ağacı algoritması olan J4.8 ile bayes sınıflandırma algoritması olan NaiveBayes algoritmaları olduğu görülmektedir.

5. KAYNAKLAR

Akın Y.K., "Veri Madenciliğinde Kümeleme Algoritmaları ve Kümeleme Analizi", Doktora Tezi, *Marmara Üniversitesi Sosyal Bilimler Enstitüsü*, Ekonometri Anabilim Dalı, İstanbul, (2008).

Aksoy E., "Matematik Alanında Üstün Yetenekli ve Zekâlı Öğrencilerin Bazı Değişkenler Açısından Veri Madenciliği ile Belirlenmesi", Yüksek Lisans Tezi, *Dokuz Eylül Üniversitesi Eğitim Bilimleri Enstitüsü*, İlköğretim Anabilim Dalı, İzmir, (2014).

Asif R., Merceron A., Ali S.A., Haider N.G., "Analyzing undergraduate students' performance using educational data mining.", *Computer & Education* Vol. 113 p177-194, http://dx.doi.org/10.1016/j.compedu.2017.05.007, (2017).

Aybek H.S.Y., Okur M.R. "Predicting Achievement with Neural Networks: The Case of Anadolu University.", In *Proceedings of Global Learn-Global Conference on Learning and Technology* Association for the Advancement of Computing in Education (AACE), (pp. 561-568). Limerick, Ireland, (2016).

Ayık Y.Z., Özdemir A. ve Yavuz U., "Lise Türü ve Lise Mezuniyet Başarısının, Kazanılan Fakülte ile İlişkisinin Veri Madenciliği Tekniği ile Analizi", *Atatürk Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, 10 (2), 449-452, (2007).

Bahadır İ., "Bayes Teoremi ve Yapay Sinir Ağları Modelleriyle Borsa Gelecek Değer Tahmini Uygulaması", Yüksek Lisans Tezi, *TOBB Ekonomi ve Teknoloji Üniversitesi Fen Bilimleri Enstitüsü*, Bilgisayar Mühendisliği Anabilim Dalı, Ankara, (2008).

Bırtıl F.S., "Kız Meslek Lisesi Öğrencilerinin Akademik Başarısızlık Nedenlerinin Veri Madenciliği Tekniği İle Analizi", Yüksek Lisans Tezi, *Afyon Kocatepe Üniversitesi Fen Bilimler Enstitüsü*, Bilgisayar Anabilim Dalı, Afyon, (2011).

Bilen M., "Yapay Sinir Ağları için Web Tabanlı bir Eğitim Yazılımı Geliştirilmesi", Yüksek Lisans Tezi, *Süleyman Demirel Fen Bilimleri Enstitüsü*, Bilgisayar Mühendisliği Anabilim Dalı, syf:16-18, Isparta, (2014).

Cortez P., Silva A., "Using Data Mining to Predict Secondary School Student Performance". *In A. Brito and J. Teixeira Eds., Proceedings of 5th FUture BUsiness TEChnology Conference (FUBUTEC 2008)* pp. 5-12, Porto (2008).

Çetin N. ve Mahir N., "Genel Matematik Dersindeki Öğrenci Başarısı İle Öss Başarısı Arasındaki İlişki", *İnönü Üniversitesi Eğitim Fakültesi Dergisi*, 7 (11), 45-46, (2006).

Çırak G., "Yükseköğretimde Öğrenci Başarılarının Sınıflandırılmasında Yapay Sinir Ağları ve Lojistik Regresyon Yöntemlerinin Kullanılması", Yüksek Lisans Tezi, *Ankara Üniversitesi Eğitim Bilimleri Enstitüsü*, Ölçme Ve Değerlendirme Anabilim Dalı, Ankara, (2012).

Ergüden Ş. ve Erşahin B., "*Veri Madenciliği: Veriden Bilgiye, Masraftan Değere",* İstanbul: ARGE Danışmanlık, 15-17, (2008).

Gökçen H., "Kamuda Karar Destek Sistemlerinin Kullanımı ve Bir Model Önerisi", *Kamu Bilgi İşlem Merkezleri Yöneticileri Birliği Kamu Bilişim Platformu XII. Nihai Rapor*, (2010).

Göker H., "Üniversite Giriş Sınavında Öğrencilerin Başarılarının Veri Madenciliği Yöntemleri ile Tahmin Edilmesi", Yüksek Lisans Tezi, *Gazi Üniversitesi Bilişim Enstitüsü*, Ankara, (2012).

Guarín C.E.L., Guzmán E.L., González F.A., "A Model to Predict Low Academic Performance at a Specific Enrollment Using Data Mining", *IEEE Revıstа Iberoamerıcana De Tecnologıas Del Aprendızaje*, Vol. 10, No. 3, (2015).

Gülçe G. "Veri Ambarı ve Veri Madenciliği Teknikleri Kullanılarak Öğrenci Karar Destek Sistemi Oluşturma", Yüksek Lisans Tezi, *Pamukkale Üniversitesi Fen Bilimleri Enstitüsü*, Bilgisayar Mühendisliği Anabilim Dalı, Denizli, (2010).

Gündüz Ş. ve Odabaşı F., "Bilgi Çağında Öğretmen Adaylarının Eğitiminde Öğretim Teknolojileri ve Materyal Geliştirme Dersinin Önemi", *The Turkish Online Journal of Educational Technology*, 3 (1), 43-48, (2004).

Han J., Kamber S.F., "*Data Mining: Concepts and Techniques*", Morgan Kaufmann Publishers, (2006).

Hotaman D., "Demokratik Eğitim: Demokratik Bir Eğitim Programı1", *Kuramsal Eğitimbilim*, 3 (1), 29-42, (2010).

Kapur B., Ahluwalia N., Sathyaraj R., "Comparative Study on Marks Prediction using Data Mining and Classification Algorithms*", International Journal of Advanced Research in Computer Science*, Vol. 8 Issue 3, p632-636. 5p., (2017).

Karakaynak Z., "Destek Vektör Makineleri İle Sınıflandırma Ve Görüntü Tanıma Üzerine Bir Uygulama", Yüksek Lisans Tezi, *Mimar Sinan Güzel Sanatlar Üniversitesi Fen Bilimleri Enstitüsü*, İstatistik Anabilim Dalı, İstanbul, (2014).

Kartal H.B., "Envanter Sınıflandırmada Yapay Öğrenme Yöntemlerinin Kullanımı ve Destek Vektör Makineleri İle Bir Uygulama", Yüksek Lisans Tezi, *İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü*, İşletme Mühendisliği Anabilim Dalı, İstanbul, (2012).

Kaura P, Singhb M, Josanc G.S., "Classification and Prediction Based Data Mining Algorithms to Predict Slow Learners in Education Sector", *In 3rd International Conference on Recent Trends in Computing* (ICRTC-2015), Procedia Computer Science 57:500-508v, (2015).

Kaya İ., "Genetik Algoritmaların Optimal Güzergâh Belirlenmesine Uygulanması", Yüksek Lisans Tezi, *Haliç Üniversitesi Fen Bilimleri Enstitüsü*, Bilgisayar Mühendisliği Anabilim Dalı, syf:16-18, İstanbul, (2012).

Koçtürk Y., "Veri Madenciliğinde Bağlılık", Yüksek Lisans Tezi, *İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü*, Bilgisayar Mühendisliği Anabilim Dalı, İstanbul, (2010).

Kolyiğit Ö., "Türkçe Dokümanlar İçin Yazar Tanıma", Yüksek Lisans Tezi, *Adnan Menderes Üniversitesi Fen Bilimleri Enstitüsü*, Matematik Anabilim Dalı, Aydın, (2013).

Kumar M., Singh A.J., "Evaluation of Data Mining Techniques for Predicting Student's Performance", *International Journal of Modern Education and Computer Science(IJMECS)*, Vol.9, No.8, pp.25-31, (2017).

Landis, Richard J., Koch G.G., "The Measurement Of Observer Agreement For Categorical Data", Biometrics, C. 33, S. 1, s.59–174., (1977).

Olsen D. L., Delen D., "*Advanced Data Mining Techniques*", Springer, 110-113, (2008).

Özdemir A., "Genetik Algoritma ile Yapay Sinir Ağlarında Yapı ve Parametre Optimizasyonu", Yüksek Lisans Tezi, *Fırat Üniversitesi Fen*

Bilimler Enstitüsü, Elektronik-Bilgisayar Eğitimi Anabilim Dalı, syf:7-10, Elazığ, (2010).

Pala T., "Tıbbi Karar Destek Sisteminin Veri Madenciliği Yöntemleriyle Gerçekleştirilmesi", Yüksek Lisans Tezi, *Marmara Üniversitesi Fen Bilimleri Enstitüsü*, Elektronik Bilgisayar Eğitimi Anabilim Dalı, İstanbul, (2013).

Parlak M., "Genetik Algoritmaların Hesapsal ve Yapısal Olarak İncelenmesi", Yüksek Lisans Tezi, *Ondokuz Mayıs Üniversitesi Fen Bilimleri Enstitüsü*, Elektrik-Elektronik Mühendisliği Anabilim Dalı, Samsun, (2007).

Paul S., Guatam N., Balint R., "*Preparing and Mining Data with Microsoft® SQL Server 2000 and Analysis Services*", Microsoft, (2002).

Ramaswami M., Bhaskaran R., "A Chaid Based Performance Prediction Model in Educational Data Mining", *IJCSI International Journal of Computer Science Issues*, 7(1), 10-18, (2010).

Saygılı A., "Veri Madenciliği ile Mühendislik Fakültesi Öğrencilerinin Okul Başarılarının Analizi", Yüksek Lisans Tezi, *Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü*, Bilgisayar Mühendisliği Anabilim Dalı, İstanbul, (2013).

Sezer Ü., "Karar Ağaçlarının Birliktelik Kuralları ile İyileştirilmesi", Yüksek Lisans Tezi, *Kocaeli Üniversitesi Fen Bilimleri Enstitüsü*, Bilgisayar Mühendisliği Anabilim Dalı, Kocaeli, (2008).

Shahiria A.M., Husaina W., Rashida N.A., "A Review on Predicting Student's Performance using Data Mining Techniques", *The Third Information Systems International Conference, Procedia Computer Science* 72 ,414 – 422, (2015).

Silahtaroğlu G., "*Veri Madenciliği Kavram ve Algoritmaları*", İstanbul:Papatya Yayıncılık, 68-74, (2013).

Superby J.F., Vandamme J.P., Meskens N., "Determination Of Factors İnfluencing The Achievement Of The First-Year University Students Using Data Mining Methods", *8th international conference on intelligent tutoring systems, Educational Data Mining Workshop*, Jhongli, Taiwan, (2006).

Şanlı O., "Yapay Sinir Ağları ile Kredibilite Tespiti", Yüksek Lisans Tezi, *Haliç Üniversitesi Fen Bilimler Enstitüsü*, Bilgisayar Mühendisliği Anabilim Dalı, syf:11-17, İstanbul (2008).

Şekeroğlu S., "Hizmet Sektöründe Bir Veri Madenciliği Uygulaması", Yüksek Lisans Tezi, *İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü*, Endüstri Mühendisliği Anabilim Dalı, syf:56-59, İstanbul, (2010).

Şık M.Ş., "Veri Madenciliği ve Kanser Erken Teşhisinde Kullanımı", Yüksek Lisans Tezi, *İnönü Üniversitesi Sosyal Bilimler Enstitüsü*, Ekonometri Ana Bilim Dalı, Malatya, (2014).

Şimşek A.S., "Bilişsel ve Duyuşsal Özelliklerin Yükseköğretimdeki Akademik Başarıyı Yordama Gücü", Yüksek Lisans Tezi, *Ankara Üniversitesi Eğitim Bilimleri Enstitüsü*, Ölçme Ve Değerlendirme Anabilim Dalı, Ankara, (2012).

Taşdemir M., "Veri Madenciliği (Öğrenci Başarısına Etki Eden Faktörlerin Regresyon Analizi ile Tespiti)", Yüksek Lisans Tezi, *Dicle Üniversitesi Sosyal Bilimler Enstitüsü*, İşletme Anabilim Dalı, Diyarbakır, (2012).

Tokat S., Karagül K., Aydemir E., "Early Strategıc Guıdance For Hıgher Vocatıonal School Students Usıng Support Vector Machınes", *International Journal on New Trends in Education and Their Implications*, Volume: 5 Issue: 3 Article: 17 ISSN 1309-6249, (2014).

Uçar E., "Ortaöğretime Geçiş Sistemi (OGES) Yerleştirme Puanlarının Uzman Sistemler İle Tahmini", Doktora Tezi, *Karabük Üniversitesi Fen Bilimleri Enstitüsü*, Bilgisayar Mühendisliği Anabilim Dalı, Karabük, (2013).

Üçgün K., "Öğretim Okulları için Öğrenci Otomasyonu Tasarımı ve Öğrenci Verileri Üzerine Veri Madenciliği Uygulamaları", Yüksek Lisans Tezi, *Marmara Üniversitesi Fen Bilimler Enstitüsü*, Elektronik-Bilgisayar Eğitimi Anabilim Dalı, İstanbul, (2009).

Ünsal Ö., "Mesleki Alan Seçimlerinin Makine Öğrenmesi Algoritması Kullanılarak Belirlenmesi", Yüksek Lisans Tezi, G*azi Üniversitesi Bilişim Enstitüsü*, Bilgisayar Eğitimi, Ankara, (2011).

Yadav S.K., Bharadwaj B., Pal S., "Data Mining Applications: A Comparative Study for Prediction Student's Performance", *International Journal of Innovative Technology & Creative Engineering*, 1(12), 13-19, (2011).

Yakut E., "Veri Madenciliği Tekniklerinden C5.0 Algoritması Ve Destek Vektör Makineleri İle Yapay Sinir Ağlarının Sınıflandırma Başarılarının Karşılaştırılması: İmalat Sektöründe Bir Uygulama", Doktora Tezi, *Atatürk*

Üniversitesi Sosyal Bilimler Enstitüsü, İşletme Anabilim Dalı, Erzurum, (2012).

Yalçın Ö., "*Veri Madenciliği Yöntemleri*", İstanbul:Papatya Yayıncılık, 185-187, (2013).

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

6. ÖZGEÇMİŞ

Adı Soyadı	: Burak AYDEMİR
Doğum Yeri ve Tarihi	: Ödemiş 25.02.1986
Lisans Üniversite	: Dokuz Eylül Üniversitesi (Bilgisayar ve Öğretim Teknolojileri Öğretmenliği)
Y. Lisans Üniversite	: Pamukkale Üniversitesi Bilgisayar Mühendisliği
Elektronik posta	: burak_aydemir86@hotmail.com
İletişim Adresi	: Değirmenönü mah. Mimar Sinan cad. No:63 K:3 D:4 DENİZLİ
Yayın Listesi	**: -**