T.C.
İSTANBUL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Yüksek Lisans Tezi

MAKİNE ÖĞRENMESİ YÖNTEMLERİ ile ORTAOKUL ÖĞRENCİ BAŞARILARININ TESPİTİ ve BİR UYGULAMA

Suat ŞAHİN

Enformatik Anabilim Dalı

Enformatik Programı

DANIŞMAN
Doç. Dr. Çiğdem EROL

Nisan, 2021

İSTANBUL

Bu çalışma, 26.04.2021 tarihinde aşağıdaki jüri tarafından Enformatik Anabilim Dalı, Enformatik Programında Yüksek Lisans tezi olarak kabul edilmiştir.

Tez Jürisi

Unvan Adı SOYADI(Danışman)
İstanbul Üniversitesi
Fakülte

Unvan Adı SOYADI
Üniversite
Fakülte

Unvan Adı SOYADI
Üniversite
Fakülte

İntihal Programı Beyanı

20.04.2016 tarihli Resmi Gazete'de yayımlanan Lisansüstü Eğitim ve Öğretim Yönetmeliğinin 9/2 ve 22/2 maddeleri gereğince; Bu Lisansüstü teze, İstanbul Üniversitesi'nin abonesi olduğu intihal yazılım programı kullanılarak Fen Bilimleri Enstitüsü'nün belirlemiş olduğu ölçütlere uygun rapor alınmıştır.

Proje Destekleri

Tezden Üretilmiş Yayınların Künye Bilgileri

ÖNSÖZ

Yüksek lisans eğitimim boyunca kendisinden çok şey öğrendiğim, tez sürecinde ilgi ve desteğini eksik etmeyen çok değerli danışmanım Doç. Dr. Çiğdem EROL'a sonsuz teşekkürlerimi sunarım.

Çalışmada kullandığım veri setinin temin edilmesi konusunda yardım ve desteklerini esirgemeyerek bana destek olan Tuzla Dede Korkut Ortaokulu'nun tüm idareci ve öğretmenlerine çok teşekkür ederim.

Tezimi her koşulda bana destek olan sevgili eşim Ceren Akyol ŞAHİN ve kısa süre önce dünyaya gelişleriyle hayatımıza sonsuz güzellikler katan oğullarım Çınar ŞAHİN ve Deniz ŞAHİN'e ithaf ederim.

Nisan 2021 Suat ŞAHİN

İÇİNDEKİLER

Sayfa No

ŞEKİL LİSTESİ

Sayfa No

TABLO LİSTESİ

Sayfa No

SİMGE VE KISALTMA LİSTESİ

Simgeler **Açıklama**

$\boldsymbol{\alpha}$	: Pozitif Lagrange çarpanı
C	: Ceza parametresi
Ç	: Çıktı vektörü
d(a,b)	: Uzaklık fonksiyonu
ξ	: Soft hata değişkeni
f(x)	: Aktivasyon fonksiyonu (Yapay Sinir Ağları)
G	: Girdi vektörü
H	: Ayırma düzlemleri (Destek Vektör Makineleri)
K()	: Kernel (çekirdek) fonksiyonu
L()	: Lagrange fonksiyonu
σ	: Standart sapma
μ	: Ortalama değer
∂	: Kısmi türev
P(x)	: Olasılık fonksiyonu
Φ	: Haritalama fonksiyonu
V	: Veri seti
W	: Ağırlık vektörü
$\mathbf{W_i}$	: i. Ağırlık vektörü
$\mathbf{x_i}$	: Veri setindeki i. örnek
$\mathbf{y_i}$	: Veri setinde hedef niteliğe ait i. örnek

Kısaltmalar	Açıklama
AUC	: ROC eğrisi altında kalan alan (*area under the curve*)
CART	: Sınıflandırma ve Regresyon Ağaçları (*Classification and Regression Trees*)
CRISP-DM	: Veri Madenciliği için Çapraz Endüstri Standardı Süreci (*Cross Industry Standard Process for Data Mining*)
DVM	: Destek Vektör Makineleri
FN	: Yanlış Negatif (*False Negative*)
FP	: Yanlış Pozitif (*False Positive*)
FPR	: Yanlış Pozitif Oranı (*False Positive Rate*)
Knn	: K-en yakın komşu algoritması
MEB	: Milli Eğitim Bakanlığı
HO	: Hold-Out yöntemi
ROC	: Alıcı İşlem Karakteristiği (*Receiver Operating Characteristic*)
SEMMA	: Örnekle, İncele, Düzelt, Modelle, Değerle (*Sample, Explore, Modify, Model, Assess*)
TN	: Doğru Negatif (*True Negative*)
TP	: Doğru Pozitif (*True Positive*)
TPR	: Doğru Pozitif Oranı (*True Positive Rate*)
YSA	: Yapay Sinir Ağları

ÖZET

YÜKSEK LİSANS TEZİ

MAKİNE ÖĞRENMESİ YÖNTEMLERİ İLE ORTAOKUL ÖĞRENCİ BAŞARILARININ TESPİTİ VE BİR UYGULAMA

Suat ŞAHİN

İstanbul Üniversitesi

Fen Bilimleri Enstitüsü

Enformatik Anabilim Dalı

Danışman : Doç. Dr. Çiğdem EROL

Makine Öğrenmesi; verinin analiz edilerek anlamlı sonuçlar çıkarılması, girdi verisinin kullanılarak tahmin modelleri geliştirilmesi amacıyla, eğitim de dahil olmak üzere birçok alanda uygulanmaktadır. Bu tez çalışmasında ortaokul öğrencilerine ait sosyoekonomik, demografik özellikleri ve ders notları verisi kullanılarak sene sonu ağırlıklı not ortalamalarının makine öğrenmesi yöntemleri ile tahmin edilmesi ve başarıya en fazla etki eden niteliklerin belirlenmesi amaçlanmaktadır.

Çalışmada kullanılan veri seti İstanbul ili Tuzla ilçesinde bulunan bir ortaokuldan temin edilmiştir. Veri seti üzerinde özellik seçimi yapılarak akademik başarı üzerinde en fazla etkiye sahip değişkenler tespit edilmiş olup modellemede bu nitelikler kullanılmıştır. Hedef niteliğin sınıflandırılması amacı ile; K-En Yakın Komşu, Karar Ağaçları, Rastgele Orman, Destek Vektör Makineleri, Yapay Sinir Ağları, Lojistik Regresyon ve Naive Bayes olmak üzere 7 adet makine öğrenmesi algoritması uygulanmış ve modellerin başarım performansları karşılaştırılmıştır. Model performans değerlendirmesi sonuçlarına göre en başarılı model Rastgele Orman algoritması olarak belirlenmiştir. Öğrencilerin akademik geçmişine ait nitelikler (geçmiş yıllar not ortalaması, Türkçe dersi 1. dönem ortalaması ve Matematik dersi 1. dönem ortalaması) başarıyı en fazla etkileyen öznitelikler olarak tespit edilmiştir. Devamsızlık durumu başarıyı etkileyen bir diğer önemli faktör olarak belirlenmiştir. Demografik ve sosyoekonomik özelliklerin akademik başarıya çok fazla etkisi olmamakla birlikte tahmin başarısına katkısı olduğu görülmüştür. Çalışmanın sonunda Rastgele Orman

modeli kullanılarak, ortaokul öğrencilerinin akademik başarılarının tahminine yönelik örnek bir sistem geliştirilmiştir. Örnek sisteme https://model-tahmin.herokuapp.com adresinden ulaşılabilmektedir. Geliştirilen sistemin eğitim yöneticileri tarafından kullanılabileceği düşünülmektedir.

Nisan 2021, 156 sayfa.

Anahtar kelimeler: Eğitim, makine öğrenmesi, tahmin, sınıflandırma, akademik başarı

SUMMARY

M.Sc. THESIS

DETERMINATION OF SECONDARY SCHOOL STUDENTS ACHIEVEMENTS WITH MACHINE LEARNING METHODS AND AN APPLICATION

Suat ŞAHİN

İstanbul University

Institute of Graduate Studies in Sciences

Department of Informatics

Supervisor : Assoc. Prof. Dr. Çiğdem EROL

Machine Learning; It is applied in many fields, including education, in order to draw meaningful results by analyzing the data and to develop prediction models using input data. In this thesis, it is aimed to predict the year-end weighted grade point averages with machine learning methods by using the socioeconomic, demographic characteristics and course grades data of secondary school students and to determine the qualities that affect the success the most.

The data set used in the study was obtained from a secondary school in Tuzla, Istanbul. By choosing the features on the data set, the variables that have the most impact on academic achievement have been determined and these attributes have been used in modeling. With the aim of classifying the target attribute; 7 machine learning algorithms, K-Nearest Neighbor, Decision Trees, Random Forest, Support Vector Machines, Artificial Neural Networks, Logistic Regression and Naive Bayes, have been applied and the performance performances of the models have been compared. According to the model performance evaluation results, the most successful model has been determined as the Random Forest algorithm. The qualifications of the students' academic background (past years grade point average, Turkish lesson first term average and Mathematic lesson first term average) have been determined as the attributes that affect the success the most. Absenteeism has been identified as another important factor affecting success. Although demographic and socioeconomic characteristics do not have much effect on academic success, it has been observed that they contribute to prediction success. At

the end of the study, a sample system has been developed for predicting the academic success of secondary school students using the Random Forest model. The sample system can be reached at https://model-tahmin.herokuapp.com. It is thought that the developed system can be used by education administrators.

April 2021, 156 pages.

Keywords: Education, machine learning, prediction, classification, academic success

1. GİRİŞ

İçinde bulunduğumuz çağda bilgi ve iletişim teknolojilerinin hızla gelişmesiyle birlikte veri kavramının önemi giderek artmaktadır. Verinin dijital ortamlarda kolay saklanması ve kolay erişilebilir olması nedeniyle veri tabanlarında depolanan veri miktarlarında büyük artışlar kaydedilmektedir. Günümüzde gerek kamu, gerekse özel sektördeki kuruluşların stratejilerinin ve iş akışlarının belirlenmesinde bilginin önemi oldukça büyüktür. Doğru ve güvenilir bilgiye erişilebilmesi için ise verinin doğru analiz edilerek anlamlı bilgiye dönüştürülmesi gerekmektedir. Depolanan verideki artış sonucunda meydana gelen büyük verinin (big data) işlenmesi ve anlamlı sonuçlar çıkarılarak bilgiye dönüşmesi mevcut sistemlerle zorlaşmıştır. Bu gelişmelere paralel olarak makine öğrenmesi ve veri madenciliği gibi disiplinler hayatımıza girmiş olup birçok alanda sıklıkla kullanılmaktadır.

Makine öğrenmesi matematik ve istatistik bilimlerinden faydalanılarak geliştirilmiş birçok disiplini bir araya getiren disiplinler arası bir çalışma alanıdır (Hastie ve diğ., 2009). Makinelerin eğitilerek problemlerin otomatik olarak algılanması ve çözüm üretilmesi prensibine dayanmaktadır (Kodratoff ve Michalski, 1990). Sağlık, finans, astronomi, endüstri, eğitim gibi birçok alanda başarıyla uygulanmaktadır. Makine öğrenmesi ve ilgili diğer disiplinler ile geliştirilen akıllı sistemler, insan yaşamını kolaylaştırmakta, geleceğe dair öngörülerde bulunarak karar alma süreçlerine olumlu katkılar sağlamaktadır (Alpaydın, 2014).

Makine öğrenmesi uygulamalarının eğitim alanındaki kullanımı son yıllarda giderek yaygınlaşmaktadır. Özellikle öğrenci başarısını artırmaya yönelik olarak, makine öğrenmesi ile akademik başarı tahmini sıkça başvurulan bir yöntem haline gelmiştir (Agarwal ve diğ., 2012). Öğrencilerin akademik başarılarının önceden öngörülebilmesi ile eğitimdeki planlama ve rehberlik çalışmalarının daha etkili olabileceği düşünülmektedir. Örneğin risk grubundaki öğrencilerin tespit edilmesi, akademik başarıyı artırmak için takviye kurs, danışmanlık gibi erken tedbirlerin alınması yönünden olumlu katkılar sağlayabilir.

Bu bilgiler ışığında bu tez çalışmasının amacı, ortaokul öğrencilerinin akademik başarılarının makine öğrenmesi yöntemleri ile tahmin edilmesi, başarıya en fazla etki eden faktörlerin belirlenmesi ve akademik başarı tahminine yönelik örnek bir sistem geliştirilmesidir. Bu amaç doğrultusunda, ortaokul öğrencilerine ait demografik, sosyoekonomik ve geçmiş akademik

başarı durumlarına ait özniteliklerden oluşan bir veri seti kullanılmıştır. Çalışmanın sınırlılıkları şu şekilde sıralanmaktadır:

- Çalışma İstanbul Tuzla ilçesinde eğitim-öğretime devam eden bir ortaokulda gerçekleştirilmiştir.
- Çalışmada kullanılan veri seti 2019-2020 eğitim yılında öğrenim gören 328 öğrenciye ait veri ile sınırlıdır.

Veri makine öğrenmesi yöntemleri ile analiz edilmiş olup en yüksek performans gösteren model kullanılarak bir sistem geliştirilmiştir. Geliştirilen sistem ile öğrencilerin sene sonu ağırlıklı not ortalamalarının bir yarıyıl öncesinden tahmin edilmesi amaçlanmaktadır. Literatürde akademik başarı tahminine yönelik çok sayıda çalışma olmasına rağmen özellikle ülkemizde ortaokul öğrencilerine yönelik başarı tahmini çalışmalarının sınırlı sayıda olduğu görülmektedir. Akademik başarı tahminine yönelik bir sistem oluşturulması ve bu sistemin gelecekte daha da geliştirilerek eğitim yöneticilerinin kullanımına sunulabilecek olması açısından önemli bir çalışma olduğu düşünülmektedir.

Bu tez çalışmasında GENEL KISIMLAR bölümünde, çalışmanın temel analiz yöntemi olan makine öğrenmesi disiplini ve işleyiş süreci detaylı olarak açıklanmıştır. Bununla birlikte makine öğrenmesi ile akademik başarı tahminine yönelik literatürde bulunan benzer çalışmalardan örnekler verilmiştir. MALZEME VE YÖNTEM bölümünde çalışmanın amacı doğrultusunda makine öğrenmesi yöntemlerinin eğitim verisi üzerinde uygulanması aşamaları (problemin tanımlanması, verinin anlaşılması, veri ön işleme ve modelleme) anlatılmıştır. BULGULAR bölümünde, uygulanan makine öğrenmesi modellerinin başarım oranları belirlenmiş ve sonuçlar karşılaştırmalı olarak verilmiştir. En yüksek performansı gösteren makine öğrenmesi modeli kullanılarak oluşturulan örnek akademik başarı tahmin sistemi sunulmuştur. TARTIŞMA VE SONUÇ bölümünde ise çalışmaya ait bulgular tartışılmış ve gelecek çalışmalara yönelik öneriler ortaya konulmuştur.

2. GENEL KISIMLAR

2.1. AKADEMİK BAŞARIYI ETKİLEYEN FAKTÖRLER

Eğitim, en basit şekliyle bireyde olumlu davranış geliştirme süreci olarak tanımlanmaktadır (Ertürk, 1994). Öğretim ise eğitimin bir parçası olarak, eğitim kurumlarında bir plan dahilinde belirlenmiş hedeflere ulaşmak için gerçekleştirilen etkinliklerdir (Demirel, 2005). Eğitim-öğretim faaliyetlerinin gerçekleştirilmesinde planlama ve rehberlik kritik bir öneme sahiptir. Eğitimde bakanlık seviyesinden okul seviyesine kadar tüm faaliyetler önceden hazırlanmış planlar çerçevesinde uygulanmaktadır (Aydemir, 2017). Eğitim-öğretim sürecinin sağlıklı bir şekilde yürütülebilmesi için öğrencilere rehberlik edilmesi, doğru yönlendirilmesi ve süreç içerisinde gerekli tedbirlerin alınması gerekmektedir. Bu kapsamda makine öğrenmesi yöntemleri uygulanarak eğitim verilerinden anlamlı sonuçlar çıkarılabilir, elde edilen anlamlı bilgiler eğitimciler ve karar vericiler tarafından kullanılarak eğitim stratejileri güncellenebilir. Akademik başarı tahmini bu amaç doğrultusunda sıkça başvurulan yöntemlerden biridir.

Öğrencilerin akademik başarılarını belirleyen çok fazla etken bulunmaktadır. Başarıyı etkileyen faktörlerin tespit edilebilmesi amacı ile çok fazla çalışma yapılmış olmasına rağmen bu faktörlerin neler olduğu kesin ve net bir şekilde ortaya konulamamaktadır. Literatürde yapılan çalışmalar incelendiğinde, demografik, ailevi ve sosyoekonomik durum ile akademik başarı arasında pozitif yönlü bir ilişki olduğunu gösteren çalışmaların yoğunlukta olduğu görülmektedir.

Pettigrew (2009), 4. Sınıf öğrencilerinin sosyoekonomik durumları ile akademik başarıları arasındaki ilişkiyi incelediği çalışmasında, sosyoekonomik olarak avantajlı öğrencilerin dezavantajlı öğrencilere göre Matematik, Dil Sanatları, Sosyal Bilgiler ve Fen Bilgisi alanlarında daha başarılı olduklarını gözlemlemiştir. Benzer şekilde Pakistan'da 1580 öğrenci ile gerçekleştirilen bir araştırmada (Akhtar ve Niazi, 2011), sosyoekonomik seviye ile akademik başarı arasında doğrusal bir ilişki olduğu sonucuna varılmıştır. Gelbal (2008), çalışmasında 8. Sınıf öğrencilerinin sosyoekonomik durumları ile Türkçe dersi başarıları arasındaki ilişkiyi analiz etmiştir. 30.714 öğrenci ile gerçekleştirilen çalışmada sonuç olarak, sosyoekonomik seviye arttıkça başarının arttığı gözlemlenmiştir. Özellikle anne eğitim seviyesinin başarıyı belirlemede önemli bir etken olduğu, bunun yanı sıra öğrencinin kendine ait odasının olmasının

başarıyı pozitif yönde etkilediği, kardeş sayısı ile akademik başarı arasında negatif yönlü bir ilişki olduğu belirtilmiştir.

Tomul ve Savasci (2012), Türkiye'de 7. Sınıf öğrencilerinin sosyoekonomik durumları ile SBS (Seviye Belirleme Sınavı) başarıları arasındaki ilişkiyi inceledikleri çalışmada, sosyoekonomik seviye ile sınav başarıları arasında yüksek korelasyon olduğunu belirtmişlerdir. Şirin (2005), 1990-2000 yılları arasında sosyoekonomik durum ile akademik başarı arasındaki ilişkinin incelendiği çalışmaları derlemiş, sosyoekonomik seviye ile akademik başarı arasında pozitif ilişki olduğunu gösteren çalışmaların yoğunlukta olduğunu belirtmiştir. Buradan hareketle öğrencilerin sosyoekonomik durumlarına ait özniteliklerin akademik başarı tahmini amacı ile kullanılabileceği düşünülmüştür.

Literatür incelendiğinde, akademik başarı tahmininde en belirleyici değişkenlerin, öğrencinin akademik geçmişine ait öznitelikler olduğu görülmektedir (Aydemir, 2017). Lisans öğrencilerinin genel Matematik ders başarıları ile üniversiteye giriş sınavı puanları arasındaki ilişkinin incelendiği bir çalışmada (Çetin ve Mahir, 2006), iki değişken arasında pozitif yönlü bir ilişki olduğu ortaya konulmuştur. Benzer bir çalışmada (Çırak, 2012), lisans öğrencilerinin akademik başarı tahmininde, üniversiteye giriş puanının ve ortaöğretim mezuniyet notunun en belirleyici öznitelikler olduğu belirtilmiştir. Türkiye'de ortaöğretime geçiş sınavı sonuçlarının tahmin edildiği bir çalışmada (Şen ve diğ., 2012), tahmine yönelik en belirleyici iki özniteliğin 7. ve 8. sınıf seviye belirleme sınavı sonuçları olduğu görülmektedir.

2.2. MAKİNE ÖĞRENMESİ

Öğrenme kavramı, bireylerde çeşitli yöntemlerle meydana gelen ölçülebilir ve kalıcı davranış değişikliği olarak tanımlanmaktadır. 1940'lı yıllardan itibaren bilim insanları tarafından "makineler düşünebilir mi?" sorusuna cevap bulunmaya çalışılmıştır (Turing, 1948). Yapay Zeka (*Artificial Intelligence*) kavramı ilk kez 1956 yılında kullanılmış ve kabul görmüştür (Akpınar, 2017). Yapay zeka alanında, bilgisayarların kapasite sorunları ve veri işlemedeki yetersizliklerinden dolayı uzun yıllar boyunca istenilen seviyede ilerleme kaydedilememiştir. 1990'lı yılların sonlarından itibaren teknolojinin gelişmesine paralel olarak bilgisayarların kapasitelerinin artması ile birlikte yapay zeka alanındaki çalışmalar yeniden hız kazanmıştır. Örneğin Deep Blue adlı bilgisayar programının dünya satranç şampiyonuna karşı yapılan oyunu kazanması, konuşma tanıma yazılımlarının hayata geçirilmesi gibi gelişmelerden sonra yapay

zekaya olan ilgi ve bu alana yapılan yatırımlar artmıştır (Nilsson, 1998; Anyoha, 2017). 2000'li yıllardan itibaren makine öğrenmesi ve derin öğrenme modelleri sürekli olarak gelişerek günümüze kadar gelmiş ve yapılan çalışmalar hız kesmeden devam etmektedir.

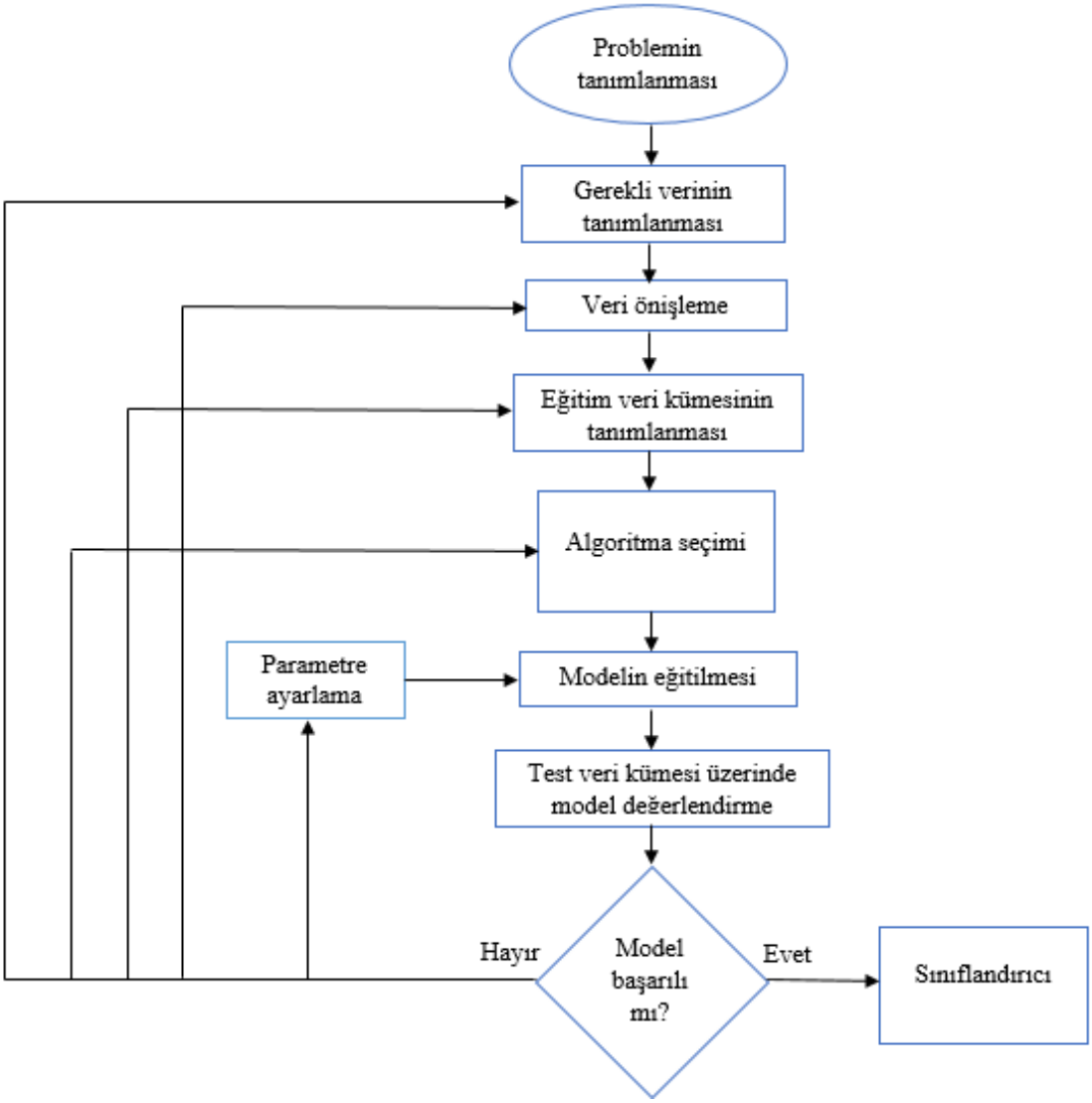
Makine öğrenmesi, yapay zeka alanının alt disiplinlerinden biridir (Patterson, 1990; Alpaydın, 2014). Eğitim, sağlık, mühendislik vb. birçok alanda faydalanılmaktadır. Genel olarak makineye verilen verinin işlenerek veriden anlamlı sonuçlar çıkarılması ve bu sonuçlardan yola çıkılarak bilinmeyene ya da geleceğe yönelik tahminlerde bulunulması prensibine dayanmaktadır. Matematik, istatistik ve bilgisayar bilimleri makine öğrenmesi disiplininin temellerini oluşturmaktadır (Harrington, 2012; Paper, 2018;).

Makine öğrenmesi yöntemleri temel olarak gözetimli öğrenme, gözetimsiz öğrenme ve pekiştirmeli öğrenme olmak üzere 3 başlık altında değerlendirilmektedir (Dangeti, 2017).

2.2.1. Gözetimli Öğrenme (Supervised Learning)

Gözetimli öğrenme, makine öğrenmesi yöntemleri arasında en yaygın kullanılan yöntemlerden biridir (Langley, 1996; Müller ve Guido, 2016). Bağımsız nitelikler (X) kullanılarak hedef niteliğin (Y) tahmin edilmesi prensibine göre çalışmaktadır. Burada hem bağımsız nitelikler hem de hedef nitelik etiketlenmiş veriden oluşmaktadır (Kotsiantis, 2007). Verilerin etiketlenmesi geçmiş deneyimlere dayanmaktadır. Daha geniş bir ifadeyle, bilinen etiketlenmiş girdi verisi ile çıktı verisi arasında bir ilişki kurularak bilinmeyen çıktıların tahmin edilmesi amaçlanmaktadır (Mohri ve diğ., 2012). Örneğin hedef niteliğin "pozitif" ve "negatif" şeklinde kategorik olarak etiketlenmiş olduğu bir veri seti olsun. Sonucun pozitif olup olmaması durumuna etki eden bağımsız nitelikler ile hedef nitelik arasında bir model kurularak yüksek tahmin başarısı elde edildiği takdirde gelecekteki duruma/kişilere ait bağımsız nitelikler kullanılarak sonucun pozitif veya negatif olduğu tahmin edilebilmektedir.

Gözetimli öğrenme literatürde danışmanlı öğrenme, denetimli öğrenme ve eğitmenli öğrenme isimleri ile de tanımlanmaktadır (Kartal, 2015). Gözetimli öğrenme problemleri için Şekil 2.1'deki aşamalar takip edilebilir.



Şekil 2.1: Gözetimli öğrenme süreci (Kotsiantis, 2007).

Gözetimli öğrenme süreci, sınıflandırma ve regresyon olmak üzere iki başlık altında incelenebilir;

2.2.1.1. Sınıflandırma

Hedef niteliğin kategorik yapıda olduğu durumlarda kullanılan yöntemdir. Hedef niteliğe ait sınıflar önceden tanımlanmıştır. Sınıflandırma problemlerinde amaç hedef niteliğin sınıf etiketini tahmin etmektir (Han ve Kamber, 2006; Harrington, 2012). Bu amaç doğrultusunda veri eğitim kümesi ve test kümesi olmak üzere ikiye ayrılır. Sınıflandırma algoritmaları kullanılarak eğitim verisi ile eğitilen bir model geliştirilir. Geliştirilen model bağımsız değişkenler ile bağımlı değişken arasında bir takım kurallar oluşturur. Daha sonra test kümesi

üzerinde modelin başarımı değerlendirilir. Modellin performans başarım oranı olumlu ise gelecekteki verilerin hangi sınıfa ait olabileceği hakkında tahmin yapılabilir.

Sınıflandırmaya ait temel kavramlar aşağıda belirtildiği gibi özetlenmiştir (Gorunescu, 2011):

- Sınıf (class) – Sınıflandırmadaki çıktı nesnesi, hedef sınıf, hedef nitelik, kategorik yapıda bulunan bağımlı değişken
- Nitelik (predictor/attribute/feature) – Bağımlı değişkene etki eden bağımsız değişkenler, bağımsız nitelikler, veri setinde bağımlı değişken haricindeki tüm sütunlar
- Örnek/kayıt (records/tuples/vectors/instances/objects/samples) – Veri setindeki her bir satır, veri seti elemanı
- Sınıflandırıcı (classifier) – Sınıflandırma işlevini gerçekleştiren fonksiyon
- Eğitim veri seti (training dataset) – Modelin/sınıflandırıcının eğitilmesinde kullanılan veri seti, sınıflandırmadaki girdi nesnesi
- Test veri seti (testing dataset) – Modelin/sınıflandırıcının başarım oranının belirlenmesi için kullanılan veri seti

Sınıflandırma görevleri hedef niteliğin yapısına bağlı olarak; ikili sınıflandırma (*binary classification*), çoklu-sınıflı sınıflandırma (*multi-class classification*), çoklu-etiketli sınıflandırma (*multi-label classification*) ve hiyerarşik sınıflandırma (*hierarchical classification*) olmak üzere 4 farklı şekilde gerçekleştirilebilmektedir (Sokolova ve Lapalme, 2009). İkili sınıflandırmada hedef niteliğe ait yalnızca 2 sınıf vardır (“evet-hayır”, “başarılı-başarısız” vb.). Çoklu-sınıflı, çoklu-etiketli ve hiyerarşik sınıflandırmada hedef nitelik ikiden fazla sınıfa sahiptir. Çoklu-sınıflı sınıflandırmada, girdi nesnesi hedef niteliğe ait sınıflardan yalnızca birine ait iken çoklu-etiketli sınıflandırma, girdinin hedef niteliğe ait sınıflardan birkaç tanesine ait olabileceği durumları ifade eder. Hiyerarşik sınıflandırma ise çoklu-sınıflı modele benzer olup sınıf değerlerinin hiyerarşik yapıya sahip olması sebebiyle bu modelden ayrılmaktadır. Tüm bu sınıflandırma modelleri için süreç aynı şekilde işlemektedir. Bu tez çalışmasında hiyerarşik sınıflandırma yönteminden faydalanılmıştır.

2.2.1.2. Regresyon

Regresyon metodunda uygulanan süreç sınıflandırma yöntemiyle çok benzerdir. Sınıflandırma modellerinde olduğu gibi eğitim kümesi ile model eğitilerek test kümesi üzerinde performans değerlendirmesi yapılmaktadır. İki yöntem arasındaki en önemli fark regresyon yönteminin

hedef niteliğin kategorik değil sürekli yapıda olduğu durumlarda kullanılmasıdır (Narsky ve Porter, 2013; Alpaydın, 2014). Dolayısıyla çıktı olarak "hasta-hasta değil" şeklinde kategorik değerler değil 89.6 gibi sayısal bir değer üretilir.

2.2.2. Gözetimsiz Öğrenme (Unsupervised Learning)

Gözetimsiz öğrenme sürecinde gözetimli öğrenme sürecinin aksine, makine herhangi bir denetim ve hedef nitelik olmadan kendi kendine öğrenir (Langley, 1996; Alpaydın, 2014; Dangeti, 2017). Veri seti yalnız X kümesinden oluşmaktadır. Gözetimsiz öğrenmede temel mantık bir değişkenin hangi gruba ait olduğunu bulmak ya da veri setinde benzer özelliklere sahip nitelikleri gruplamaktır. Gözetimli öğrenme yönteminden bir diğer farkı da verilerin etiketlenmemiş olmasıdır (Harrington, 2012; Hackeling, 2014). Müşteri segmentasyonu, gen araştırmalarında benzerlik tespiti, pazar-sepet analizi gibi uygulamalarda sıklıkla gözetimsiz öğrenme yöntemlerinden faydalanılmaktadır (Han ve Kamber, 2006; Shalev-Shwartz ve Ben-David, 2014).

2.2.2.1. Kümeleme

Kümeleme yönteminin temel amacı verilerin analiz edilerek gruplanmasıdır. Veri setindeki kayıtların benzer özelliklerine göre kümelere/gruplara ayrılması sağlanır (MacQueen, 1967; Olson, 2017). Verilerin benzerlikleri, bulundukları uzayda birbirlerine olan uzaklıkların hesaplanması ile tespit edilmektedir (Akın, 2008). Kümeleme işlemi aşağıda belirtilen kurallar gözetilerek gerçekleştirilmektedir (Chen ve diğ., 1996).

- Veri setinde aynı kümede yer alan örneklerin benzerliği maksimize edilir,
- Aynı kümede yer alan örneklerin diğer kümelerdeki örneklerle benzerliği minimize edilir,
- Veri setindeki örnekler arasındaki benzerlikler tespit edilerek benzerliğe dayalı kümeler elde edilir.

Aynı zamanda sınıflandırma uygulanmak istenen verinin sınıf etiketlerine sahip olmaması durumunda kümeleme yöntemi kullanılarak sınıf etiketleri belirlenebilmektedir (Koçoğlu, 2017). k-means ve hiyerarşik kümeleme sık kullanılan kümeleme algoritmalarıdır.

2.2.3. Pekiştirmeli Öğrenme (Reinforcement Learning)

Pekiştirmeli öğrenme davranışçılık kuramından esinlenerek geliştirilmiş bir makine öğrenmesi modelidir. Makinenin öğrenme işlemi, yapılan bir eylem sonucunda çevreden aldığı geri bildirimlerle gerçekleşmektedir (Kaelbling ve diğ., 1996; Sutton ve Barto, 1998; Alpaydın, 2014). Pekiştirmeli öğrenme modellerinde öğrenme süreci aşağıda belirtildiği şekilde işleyebilmektedir (Nandy ve Biswas, 2017).

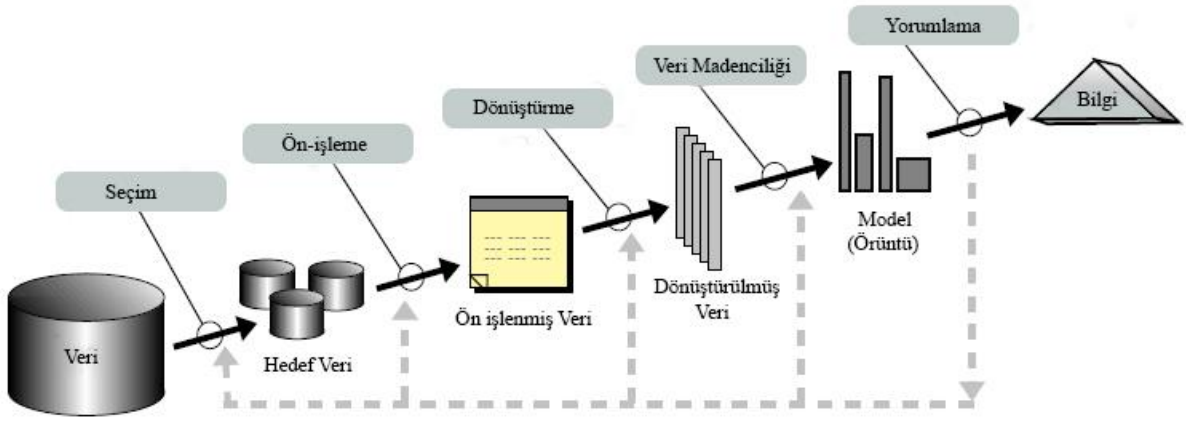
- Model tarafından hedefe ulaşmak için olası durumlar denenir,
- Hedefe ulaşılırsa ödül, ulaşılamazsa ceza puanı alınır,
- Ceza puanı alınan hamleler tekrarlanmaz,
- Ödül puanı alınan hamleden yararlanılarak öğrenmeye devam edilir,
- Sistem maksimum ödülü amaçlayarak sürekli olarak öğrenmeye devam eder ve öğrenme süreci durdurulmaz.

Literatürde takviyeli öğrenme olarak da tanımlanmaktadır.

2.3. MAKİNE ÖĞRENMESİ AŞAMALARI

Makine öğrenmesi problemlerinin çözülmesi için çeşitli yaklaşımlar bulunmaktadır. Literatürdeki mevcut yaklaşımlar birbirleri ile benzer süreçlere sahip veri madenciliği yöntemleridir. Veri madenciliği sürecinde makine öğrenmesi algoritmaları kullanılmaktadır (Özkan ve Selçukcan Erol, 2017). Bu nedenle makine öğrenmesi aşamaları, genellikle veri madenciliğinin literatürde tanımlanan modelleri ile yürütülmektedir. Literatür incelendiğinde CRISP-DM (*Cross Industry Standard Process for Data Mining*), SEMMA (*Sample, Explore, Modify, Model, Assess*) ve KDD (*Knowledge Discovery in Databases*) olmak üzere üç farklı süreç modelinden bahsedilmektedir (Olson ve Delen, 2008).

Veri tabanlarında bilgi keşfi süreci (KDD), Fayyad ve diğ. (1996) tarafından yayınlanmış bir veri madenciliği süreç modelidir. Veri tabanlarında bilgi keşfi modelinde ham verinin bilgiye dönüştürülmesi süreci beş adımda gerçekleştirilmektedir (Şekil 2.2).

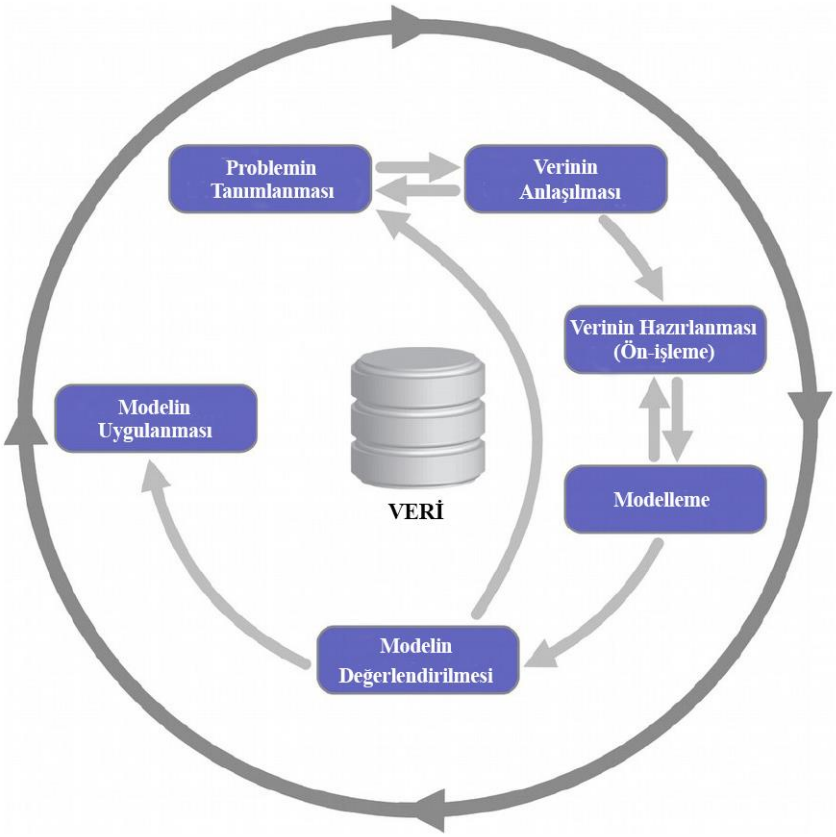


Şekil 2.2: KDD süreci akış şeması (Fayyad ve diğ., 1996; Akpınar, 2017).

SEMMA modeli SAS Institute tarafından SAS Enterprise Miner aracı için geliştirilmiş veri madenciliği süreci olup beş adımda gerçekleştirilmektedir (SAS, 2017). Bu adımlar aşağıda belirtildiği gibidir:

- Örnekleme: Verinin örneklendiği aşamadır.
- İnceleme: Veri setindeki nitelikler arasındaki ilişkilerin keşfedilmesi, verinin anlaşılması ve görselleştirilmesi aşamasıdır.
- Dönüştürme: Verinin model kurma işlemi için hazır hale getirildiği adımdır.
- Modelleme: Veri üzerinde veri madenciliği teknikleri kullanılarak model geliştirildiği aşamadır.
- Değerleme: Kurulan modelin başarısının ve geçerliliğinin değerlendirildiği aşamadır.

Veri Madenciliği için Çapraz Endüstri Standart Süreç Modeli (CRISP-DM) SPSS tarafından geliştirilmiş bir yöntemdir (Cheapman ve diğ., 2000; Akpınar, 2017). Problemin Tanımlanması, Verinin Anlaşılması, Verinin Hazırlanması, Modelleme, Model Değerlendirme ve Modelin Uygulanması olmak üzere altı aşamada gerçekleştirilmektedir (Shearer, 2000). CRISP-DM uygulama aşamaları Şekil 2.3'de gösterilmektedir. Bu tez çalışmasında makine öğrenmesi modelleri CRISP-DM süreci aşamaları takip edilerek geliştirilmiştir.



Şekil 2.3: CRISP-DM modeli aşamaları (Shearer, 2000).

2.3.1. Problemin Tanımlanması

Problemin tanımlanması aşaması tüm bilimsel süreçlerde olduğu gibi makine öğrenmesi çalışmalarında da uygulanması gereken ilk aşamadır. Öncelikle problemin ne olduğu ve kurulacak olan makine öğrenmesi modelinin amacı net bir şekilde tanımlanmalıdır. Net bir şekilde tanımlanan makine öğrenmesi problemi, sürecin başarılı olması için gerekli ilk adımdır (Mitchell, 1997; Kartal, 2015).

2.3.2. Verinin Anlaşılması

Verinin sağlıklı bir şekilde analiz edilebilmesi ve model kurulabilmesi için çalışmada kullanılan veri seti yapısının iyi anlaşılması gerekmektedir. Bu amaç doğrultusunda veri setinin genel görünümü, verinin istatistiksel dağılımı, nitelikler arası korelasyon, eksik, aykırı ve gürültülü verilerin tespiti gibi işlemler yapılarak mevcut durum görselleştirilebilmektedir. Görselleştirme için sütun, pasta grafikleri, kutu grafikleri (*box-plot*), histogram ve ısı haritası gibi nesnelerden faydalanılmaktadır (Roiger ve Geatz, 2003; Kartal, 2015).

Bu aşamadaki analizlerden elde edilen gözlemler ışığında veri ön-işleme aşamasında uygulanacak teknikler belirlenebilmektedir.

2.3.3. Veri Hazırlama – Veri Ön-işleme (Data Preprocessing)

Veri ön-işleme, veri seti üzerinde algoritmalar uygulanmadan önce verinin modellemeye hazır hale getirildiği adımdır. Çalışmalarda kullanılan veri setleri genellikle doğrudan modellemeye elverişli değildir (Zhang ve diğ., 2003; Emre, 2017). Eksik (*missing*), aykırı (*outlier*), gürültülü (*noisy*) veya tutarsız (*inconsistent*) değerler içerebilmektedir (Roiger, 2016). Eksik veri olması durumunda makine öğrenmesi modelleri çalışmazken aykırı ve tutarsız veriler oluşturulan modelin performansını ve doğruluğunu olumsuz şekilde etkilemektedirler. Bu nedenle verinin modelleme öncesinde bazı işlemlerden geçmesi gerekmektedir. Veri hazırlama adımı makine öğrenmesi süreci açısından önemli ve kritik bir aşamadır. Ön-işleme sürecinde doğru veya yanlış tekniklerin uygulanması, oluşturulacak olan modelin doğruluğunu ve verimliliğini belirlemektedir (Pyle, 1999).

Veri hazırlama sürecinin kapsamı literatürde farklı şekillerde ele alınmış olup veri temizleme, veri birleştirme, veri dönüştürme, veri indirgeme olmak üzere dört ana başlık altında değerlendirilmektedir (Han ve diğ., 2012). Bu süreçte uygulanacak yöntemler analiz edilen veri setine göre belirlenmektedir.

2.3.3.1. Veri Temizleme (Data Cleaning)

Eksik verinin tamamlanması, aykırı ve gürültülü değerlerin tespiti ve düzeltilmesi, verideki tutarsızlıkların giderilmesi işlemleri veri temizleme kapsamında değerlendirilmektedir (Rahm ve Do, 2000).

Veri setinde çeşitli nedenlerden dolayı eksik değerler bulunabilmektedir. Bu nedenler arasında eksik veri girişi, sistemsel hatalar, anket yoluyla veri toplanması halinde kişilerin bazı bilgileri vermek istememesi gibi durumlar sayılabilir. Eksik değerlerin tamamlanması için çeşitli yaklaşımlar mevcuttur. Bu yöntemlerin bir kısmı aşağıdaki gibi sıralanabilir (Han ve Kamber, 2006; Akpınar, 2017;).

- Veri setinde aynı satırda (bir örneğe ait) çok sayıda eksik değer mevcut ise o satır tümüyle silinebilir.

- Eksik değerler, ait oldukları nitelik kategorik yapıda ise ilgili sütunun mod değeri (en sık tekrar eden değer) ile tamamlanabilir.
- Eksik kayıtların bulunduğu nitelik sütununa ait ortalama değer veya medyan değeri eksik değerlerin yerine atanabilir. Bu yöntem genellikle ilgili niteliğin sürekli (nümerik) yapıda olduğu durumlarda kullanılmaktadır.
- Eksik verinin tamamlanması işleminde bazı makine öğrenmesi algoritmalarından da faydalanılabilmektedir. Bu yaklaşımda eksik değerler sınıflandırma veya regresyon problemi gibi ele alınarak tahmin edilmektedir. Eksik değerler diğer örneklerden ayrılarak hedef nitelik gibi düşünülmektedir. Örnek olarak Scikit-Learn kütüphanesinde bulunan KnnImputer fonksiyonu K-En Yakın Komşu algoritmasını kullanarak eksik değerlerin tamamlanmasını sağlamaktadır.
- Tüm eksik değerler "NAN" gibi sabit bir değer ile doldurulabilir.

Hatalı ve aykırı değerler gürültülü veri olarak değerlendirilmektedir. Gürültülü değerlerin tespiti ve sorunun giderilmesi veri kalitesi açısından önemlidir (Zu ve Wu, 2004; Garcia ve diğ., 2016; Akpınar, 2017). Aykırı değerler bulundukları uzayda diğer örneklerden belirgin şekilde farklılık gösteren örneklerdir (Grubbs, 1969; Barnett ve Lewis, 1994). Örneğin veri setinde bir insanın boyu 3 metre olarak görünüyorsa bu aykırı bir değerdir. Veri setinde bulunan aykırı değerler, oluşturulacak olan modelin yanlış eğitilmesine sebep olabilmektedirler (Patel, 2019).

Gürültülü verinin düzeltilmesi için üç farklı yöntem izlenebilmektedir (Akın, 2008).

- Kutulama (binning) yönteminde değerler sıralanır, eşit genişlik ve eşit derinlik ile bölünür. Her bölme ortalama değerle veya bölmenin alt ve üst sınırları ile temsil edilir.
- Kümeleme yönteminde benzer örnekler aynı kümede olacak şekilde gruplanarak kümelerin dışında kalan örnekler aykırı olarak kabul edilerek silinir.
- Eğri uydurma yönteminde ise veri bir fonksiyona ait eğriye uydurulur, bir değişkenin değeri diğer değişkenler yardımı ile tespit edilir.

Tutarsız veri ise veri setlerinde sık karşılaşılan, değerlerin birbirleriyle uyuşmazlık içerisinde oldukları bir durumdur. Örneğin önceki dönemlerde bir nitelik alanına "A,B" şeklinde girilen değerlerin sonraki dönemde "1,2" şeklinde girilmesi halinde o niteliğe ait tutarsız değerler ortaya çıkmaktadır. Tekrar eden kayıtlar da tutarsız veri kapsamında değerlendirilebilir ve veri setinden çıkartılmalıdır (Buttrey ve Whitaker, 2017).

2.3.3.2. Veri Dönüştürme (Data Transformation)

Veri setinde yer alan niteliğe/niteliklere ait değerlerin dönüştürülmesi işlemidir (Han ve diğ.,2012). Veri ayrıklaştırma ve normalizasyon olmak üzere iki başlık altında incelenmektedir.

Veri ayrıklaştırma işlemi sürekli yapıya sahip nitelik değerlerinin kategorik değerlere dönüştürülmesidir (Chmielewski ve Grzymala-Busse, 1996; Garcia ve diğ., 2015). Örneğin bir veri setinde yer alan fiyat nitelik alanındaki nümerik değerler "düşük-orta-yüksek" olacak şekilde üçlü kategorik bir yapıya dönüştürülebilir. Ayrıklaştırma işlemi yapılırken veri kaybının minimum olması hedeflenir (Jin ve diğ., 2007).

Veri setinde yer alan niteliklerin değer aralıkları farklılık gösterebilmektedir. Örnek olarak bir nitelik alanındaki değerler 1000-100000 aralığında iken diğer bir nitelik alanında ise 0.1-0.9 aralığında olabilir. Değer aralığındaki farklılıklar bazı durumlarda geniş aralıklı niteliğin baskın olması sebebiyle uygulanacak olan modeli yanıltabilir. Bu durumu düzeltmek amacıyla normalizasyon yönteminden faydalanılmaktadır. Normalizasyon işlemi, tüm niteliklerin aynı değer aralığında olacak şekilde ölçeklendirilmesidir. Min-Max normalizasyou ve z-skor normalizasyonu sık kullanılan normalizasyon yöntemleridir.

Min-Max normalizasyonunda tüm nitelik değerleri 0-1 aralığında olacak şekilde düzenlenmektedir. X nitelik alanlarına ait değerler olmak üzere Denklem 2.1'de gösterildiği şekilde hesaplanmaktadır.

$$x_{normal} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{2.1}$$

z-skor normalizasyonunda ortalama değer 0, standart sapma 1 değerini almakta ve dağılım normale yaklaşmaktadır. İşlem sonucunda değerlerin büyük bölümü -1,1 aralığında yer almaktadır. μ ortalama, σ standart sapma olmak üzere Denklem 2.2'de gösterilmektedir.

$$x_{normal} = \frac{X - \mu}{\sigma} \tag{2.2}$$

2.3.3.3. Veri Birleştirme (Data Integration)

Veri birleştirme farklı veri tabanlarında ya da veri kaynaklarında bulunan verinin birleştirilmesi işlemidir (Chakrabarti ve diğ., 2008; Agarwal, 2013; Özkan, 2013). Veri birleştirme işlemi yapılırken veri kaynakları arasındaki farklılıklara dikkat edilmesi gerekmektedir (Akın, 2008). Bir veri tabanında "personel_id" olarak tanımlanan bir nitelik alanı diğer veri tabanında

"personel_numarası" olarak tanımlanmış olabilir. Benzer şekilde bir veri tabanında metre birimiyle tutulan kayıtlar bir başka veri tabanında santimetre birimiyle tutulmuş olabilir.

2.3.3.4. Veri İndirgeme (Data Reduction)

Makine öğrenmesi uygulamalarında veri setlerindeki veri boyutunun fazla olması model karmaşıklığını artırabilmektedir (Alpaydın, 2014). Bununla birlikte zaman ve kaynak kullanımını da olumsuz yönde etkilemektedir. Veri indirgeme işlemi, veri boyutunu problemin çözümüne yönelik olarak, veri setini en iyi şekilde temsil edebilecek minimum boyuta getirmeyi amaçlamaktadır (John ve diğ., 1994; Fayyad ve diğ., 1996). Bu işlem gerçekleştirilirken orijinal verinin temel yapısının ve bütünlüğünün korunması gerekmektedir. Özellik seçimi (*feature selection*) ve özellik çıkarımı (*feature extraction*) sık kullanılan veri indirgeme yöntemleridir.

Özellik seçimi; veri setinde gereksiz olarak görülen, sonuca etkisi olmayan veya çok az olan bağımsız değişkenlerin elenmesi işlemidir (Hall, 1999). Özellik seçiminde bağımlı değişkene etkisi en yüksek olan bağımsız değişkenlerin seçilmesi amaçlanmaktadır. Etki değeri yüksek niteliklerin belirlenebilmesi için çeşitli yaklaşımlar bulunmaktadır. Ki-kare testi, bilgi kazancı, One-R, t-skor özellik seçim yöntemlerinden bazılarıdır (Budak, 2018).

Özellik çıkarımı yönteminde ise, veri setinde mevcut olan bağımsız niteliklerin birleşiminden belirtilen sayıda yeni nitelikler üretilerek boyut azaltma işlemi gerçekleştirilmektedir (Han ve diğ., 2012). Veri, üretilen yeni nitelik uzayında temsil edilir. Temel Bileşen Analizi (*Principal Component Analysis*) ve Doğrusal Ayırt eden Analizi (*Linear Discriminant Analysis*) en iyi bilinen özellik çıkarımı yöntemleridir.

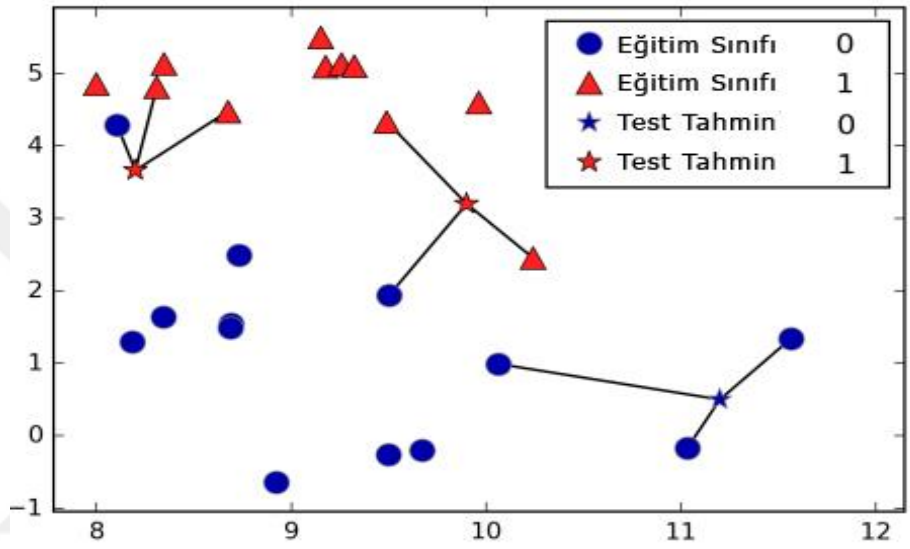
2.3.4. Modelleme

Ön işlenmiş veri üzerinde makine öğrenmesi algoritmalarının uygulanarak model oluşturulduğu aşamadır (Flach, 2012). Bu aşamada modellere ait parametre ayarlamaları yapılarak en başarılı sonuçlar elde edilemeye çalışılmaktadır (Azevedo ve Santos, 2008). Bu tez çalışmasında kullanılan algoritmalar ve çalışma prensipleri sırası ile anlatılmaktadır.

2.3.4.1. K-En Yakın Komşu Algoritması

İlk kez 1967 yılında Thomes M. Cover ve Peter E. Hart tarafından önerilmiş olan sınıflandırma yöntemidir (Cover ve Hart, 1967). Sınıf değeri bulunmak istenen örneğin, verilerin dağıldığı

uzayda kendisine en yakın sınıf değeri önceden bilinen k adet örneğe uzaklığı hesaplanarak tahmin edilmesi prensibine göre çalışmaktadır. Örneğin hedef niteliğin 0-1 olmak üzere ikili kategorik yapıda olduğu bir problem olsun. k=1 olarak alınırsa, sınıf değeri bilinmeyen örnek kendisine en yakın örneğin sınıf etiketini alır. k=3 olarak alındığında ise, kendisine en yakın üç örnekten sayıca fazla olan sınıf etiketini alır. Şekil 2.4'de K-En Yakın Komşu sınıflandırmasına ait bir örnek gösterilmektedir (Müller ve Guido, 2016).



Şekil 2.4: k=3 için örnek sınıflandırma modeli (Müller ve Guido, 2016).

Değerler arası uzaklıkların hesaplanmasında farklı yöntemler kullanılmaktadır. Öklit, Manhattan, Minkowski ve Chebyshev uzaklıkları sık kullanılan hesaplama yöntemleridir. Öklit uzaklığına göre, iki nokta arasındaki uzaklık bu iki noktayı birleştiren doğrunun uzunluğuna eşittir (2.3).

$$d(a,b) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{2.3}$$

Manhattan yönteminde uzaklık, noktalar arasındaki mutlak uzaklıkların toplamı kadardır (2.4).

$$d(a,b) = \sum_{i=1}^{n}|x_i - y_i| \tag{2.4}$$

Minkowski uzaklığı ise Öklit ve Manhattan uzaklıklarının genelleştirilmiş bir halidir (2.5).

$$d(a,b) = [\sum_{i=1}^{n}(|x_i - y_i|)^q]^{1/q} \tag{2.5}$$

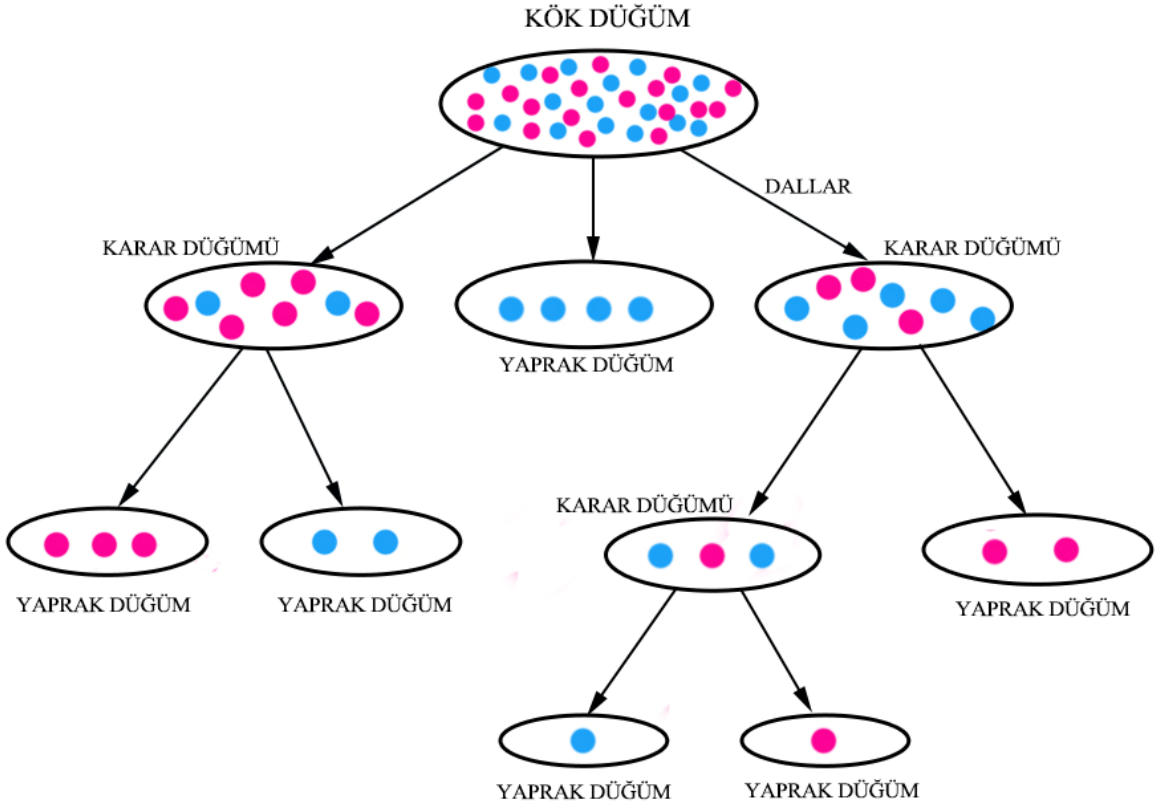
Chebyshev uzaklığına göre iki nokta arasındaki uzaklık, bu iki nokta arasındaki en uzak boyuttaki mesafe kadardır (2.6).

$$d(a,b) = \underset{i}{Max}(|x_i - y_i|) \tag{2.6}$$

2.3.4.2. Karar Ağaçları

Karar ağaçları kolay yorumlanabilmesi, işlem adımlarının basit ve hızlı olması gibi özellikleriyle oldukça popüler bir makine öğrenmesi modelidir (Han ve Kamber, 2006). Karar ağacı modellerinde öğrenme işlemi, veri setine art arda sorular sorularak gerçekleştirilmektedir (Quinlan, 1986; Dangeti, 2017). Soruları veri setinde bulunan nitelikler belirler.

Sınıflandırma sürecinde, ağaç yapısının oluşturulması için en belirleyici bağımsız nitelik tespit edilir. Bu nitelik kök düğümü (*root node*), bir başka deyişle ağacın tepe noktasını ifade eder. Kök düğümden itibaren ağaç hiyerarşik bir biçimde dallara ayrılarak karar düğümleri (*decision nodes*) ve yaprak düğümleri (*leaf nodes*) oluşur. Ağacın dallarında sınıflandırma işleminin gerçekleşme durumunda yaprak düğümleri, gerçekleşmeme durumunda ise karar düğümleri oluşmaktadır (Köktürk, 2012). Ayrıştırıcı nitelik kalmadığında dallanma işlemi sona erer. Örnek bir ağaç yapısı Şekil 2.5'de sunulmaktadır.



Şekil 2.5: Karar ağacı örneği (Quinlan, 1986).

Kök düğümün tespiti ve dallanma kurallarının belirlenmesi için çeşitli hesaplama yöntemleri kullanılmaktadır. Bilgi kazancı, Gini endeksi ve kazanç oranı bu yöntemler arasındadır.

Karar ağaçları modelleri oluşturmak için çeşitli algoritmalar kullanılmaktadır. CHAID (Kass, 1980), CART (Breiman ve diğ., 1984), ID3 (Quinlan, 1986), C4.5 (Quinlan, 1993), C5.0 (Quinlan, 1996) ve QUEST (Loh ve Shih, 1997) algoritmaları karar ağacı modellerine örnektir. Karar ağaçları algoritmalarından ID3 bilgi kazancını, C4.5 ve C5.0 algoritmaları kazanç oranını, CART algoritması ise Gini katsayısını kullanmaktadır.

Bilgi kazancı yönteminde en ayırt edici özelliklerin ve bölünme kurallarının belirlenmesi amacıyla her nitelik için bilgi kazancı hesaplanmaktadır. Bilgi kazancı hesaplanmasında entropi kullanılır. Entropi kavramı belirsizlik, rastgelelilik ve beklenmeyen durumların olma olasılığını ifade eder. Bir veri kümesi için T hedef nitelik, $T_i=\{T_1, T_2,\ldots,T_n\}$ niteliğe ait sınıflar, $P(T_i)$ sınıflara ait gözlemlerin olma olasılığı olmak üzere entropi formülü denklem (2.7)'de belirtildiği şekildedir (Shannon, 1948).

$$Entropi(T) = -\sum_{i=1}^{n} p(T_i)\ log_2\, p(T_i) \quad (2.7)$$

Örneğin T hedef niteliğine ait 10 örnekten;

- 4 örnek T_1 sınıfına,
- 3 örnek T_2 sınıfına,
- 1 örnek T_3 sınıfına,
- 2 örnek T_4 sınıfına ait olsun. Bu durumda entropi;

Entropi(T) = - (4/10)$\log_2 4/10$ - (3/10)$\log_2 3/10$ - (1/10)$\log_2 1/10$ - (2/10)$\log_2 2/10$ = 1,843 olarak hesaplanacaktır.

Herhangi bir A niteliğindeki değerleri alan örneklerin T hedef niteliğindeki hangi sınıfa ait olabileceğini belirlemek için denklem (2.8)'de belirtilen formül kullanılmaktadır.

$$Entropi_A(T) = \sum_{j=1}^{n} \frac{|T_j|}{|T|}\ Entropi(T_j) \quad (2.8)$$

Bu iki formülden yola çıkılarak A niteliğinin T hedef niteliği üzerindeki bilgi kazancı denklem (2.9)'daki şekilde hesaplanmaktadır.

$$Bilgi\ Kazancı(A) = Entropi(T) - Entropi_A(T) \quad (2.9)$$

Denklem (2.7)'de belirtilen formül A niteliğinden yapılacak bir dallanmada elde edilecek bilgi kazancını verir. Bilgi kazancı her bağımsız nitelik için yapılarak bölünme işlemi en yüksek bilgi kazancına sahip nitelikten başlar.

Gini katsayısı CART algoritması tarafından kullanılmakta olup, her bir nitelik için ikili bölünme yapılması prensibine göre çalışmaktadır (Breiman ve diğ., 1984). Denklem (2.10)'da belirtildiği şekilde hesaplanır.

$$Gini(T) = 1 - \sum_{i=1}^{n} {p_i}^2 \quad (2.10)$$

Her nitelik için ikili bölünme yapılacağından dolayı T niteliğine ait T_1 ve T_2 değerleri denklem (2.11)'deki şekilde hesaplanmaktadır.

$$Gini_A\,(T) = \frac{|T_1|}{|T|} Gini(T_1) + \frac{|T_2|}{|T|} Gini(T_2) \quad (2.11)$$

Her bir nitelik için Gini katsayısı hesaplanarak en küçük Gini değerine sahip nitelikten başlayarak bölünme işlemi gerçekleştirilir.

Kazanç oranı (*gain ratio*) yöntemi C4.5 ve C5.0 algoritmaları tarafından kullanılmakta olup tüm nitelikler için kazanç oranı hesaplanarak en yüksek kazanç oranına sahip nitelikten bölünme gerçekleştirilir. Bölünme bilgisi, A niteliğine ait $\{A_1,A_2,\ldots,A_n\}$ değerleri için T niteliği ayrıldığında elde edilecek olan bilgiyi göstermektedir. Bölünme bilgisi için denklem (2.12)'den faydalanılmaktadır.

$$Bölünme\ Bilgisi_A(T) = -\sum_{i=1}^{n} \frac{|T_i|}{|T|} log_2\left(\frac{T_i}{T}\right) \tag{2.12}$$

Kazanç oranı ise Denklem (2.13)'deki gibi hesaplanmaktadır.

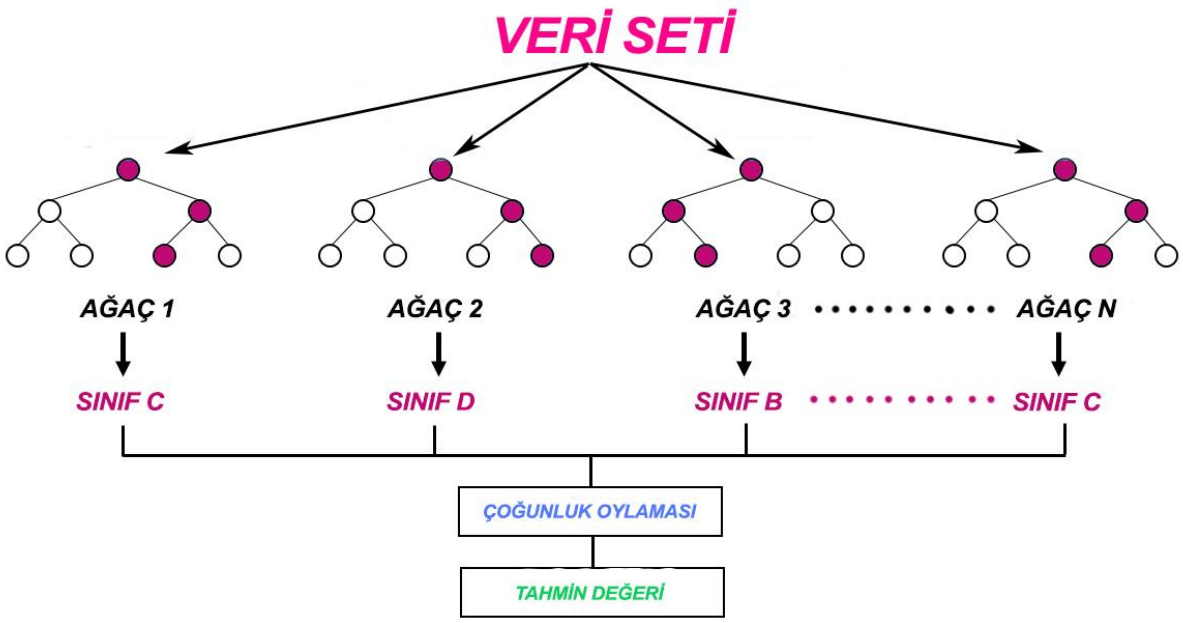
$$Kazanç\ Oranı(A) = \frac{Bilgi\ Kazancı(A)}{Bölünme\ Bilgisi_A(T)} \tag{2.13}$$

Karar ağaçları algoritmalarında bölünme (dallanma) olayı durdurulmadığı takdirde aşırı öğrenme (*overfitting*) olayı meydana gelebilmektedir. Bir başka deyişle model veriyi ezberleme yoluna gidebilmektedir. Bu durumu engellemek için karar ağaçlarında budama yöntemi kullanılarak ağaç yapısının daha sağlıklı bir hale getirilmesi sağlanmaktadır.

2.3.4.3. Rastgele Orman Algoritması

Rastgele Orman algoritması, özellikle yüksek boyutlu sınıflandırma problemleri için başarısı kanıtlanmış olan popüler kolektif (*ensemble*) makine öğrenmesi modellerinden biridir (Azar ve diğ., 2014). Kolektif öğrenme modelleri ile birden fazla modele ait çıktıların birleştirilerek daha sağlıklı sonuçlar elde edilmesi amaçlanmaktadır.

Rastgele orman modeli, birden fazla karar ağacı modelinden oluşmaktadır (Breiman, 2001). Her bir karar ağacı modelinden gelen sınıf değerleri tahminleri birleştirilir ve oylamaya tabi tutulur. En yüksek oyu alan sınıf değerleri nihai tahmin değeri olarak belirlenir (Şekil 2.6).



Şekil 2.6: Rastgele orman modeli (Belgiu ve Dragut, 2016).

Rastgele Orman modelinde özniteliklerin seçimi rastgele yapılır ve karar ağaçlarının bölünme işlemi Gini indeksi kullanılarak CART yöntemi ile gerçekleştirilmektedir (Pal, 2005; Emre, 2017). Breiman (1999), modelde çok sayıda ağaç kullanılması sebebiyle ağaçların budanmadan da aşırı öğrenme probleminin yaşanmayacağını ve genelleme hatalarının düşük olacağını ifade etmiştir.

2.3.4.4. Destek Vektör Makineleri

1992 yılında Bernhard B. Boser, Isabelle M. Guyon ve Vladimir N. Vapnik tarafından geliştirilmiş sınıflandırma yöntemidir (Boser ve diğ., 1992). Doğrusal olmayan sınıflandırma problemlerinde başarılı sonuçlar vermesi ve çok sayıda öznitelik ile çalışabilmesi açısından tercih edilen bir yöntemdir. Sınıfları birbirinden en iyi şekilde ayırabilecek, sınıflar arasında maksimum marjini sağlayan bir hiper düzlem bulunması prensibine göre çalışmaktadır (Cortes ve Vapnik, 1995; Harrington, 2012; Olson ve Wu, 2017).

Destek Vektör Makineleri (DVM) iki farklı şekilde uygulanabilmektedir. Verinin doğrusal olarak ayrılabildiği durumlarda doğrusal DVM, doğrusal olarak ayrılamadığı durumlarda ise doğrusal olmayan DVM kullanılmaktadır (Alpaydın, 2014; Filiz, 2019).

- **Doğrusal Sınıflandırma Problemi**

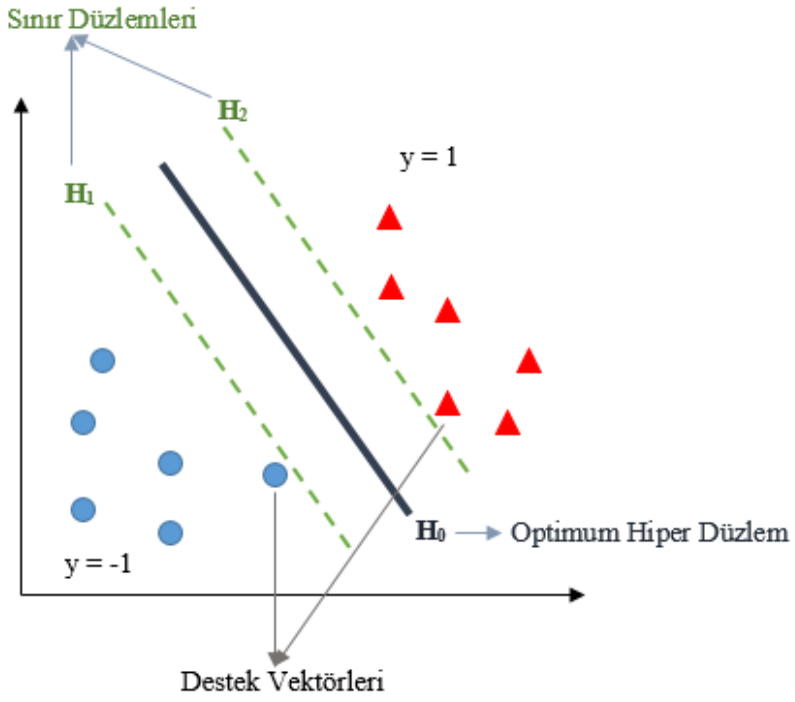
Sınıflar arasında doğrusal ayrımın yapabilecek optimal hiper düzlemin bulunabileceği durumları ifade etmektedir. Örnek olarak iki boyutlu bir veri seti denklem (2.14)'deki şekilde tanımlanmış olsun.

$$V = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n), x_i \in R^d, y_i \in \{-1, +1\}\ \} \quad (2.14)$$

Denklem 2.14'de x_i giriş vektörlerini, y_i ise x_i değerlerine karşılık gelen sınıf etiketlerini belirtmektedir. $W=\{w_1,w_2,...,w_n\}^T$ ağırlık vektörü, b sabit olmak üzere karar fonksiyonu hiper düzlemler ile ilişkilendirilmiş olarak denklem (2.15) ile gösterildiği şekilde tanımlanmaktadır.

$$f(x) = W^T x_i + b \quad (2.15)$$

Şekil 2.7'de doğrusal sınıflandırma için örnek DVM yapısı gösterilmektedir.



Şekil 2.7: DVM'de iki boyutlu uzayda ayırma düzlemleri (Cortes ve Vapnik, 1995).

Şekil 2.7'de H_1 ve H_2 iki sınıf arasındaki maksimum uzaklığı sağlayan sınır düzlemlerini, H_0 ise optimum ayırma düzlemini ifade etmektedir. H_0, H_1 ve H_2 düzlemleri sırasıyla denklem (2.16), (2.17), (2.18)'de gösterildiği şekilde hesaplanmaktadır.

$$H_0: \sum_{i=1}^{n} w_i x_i + b = 0 \quad (2.16)$$

$$H_1: \sum_{i=1}^{n} w_i x_i + b = -1 \quad (2.17)$$

$$H_2: \sum_{i=1}^{n} w_i x_i + b = +1 \quad (2.18)$$

Hiper düzlemler; sınıf değerleri H1 düzleminin altında $W^T x_i + b \leq -1$, ve H2 düzleminin üstünde $W^T x_i + b \geq +1$ eşitsizliklerini sağlayacak şekilde olmalıdır. H_1 ve H_2 düzlemlerinin arasında kalan bölge sınır bandı olarak adlandırılır. Sınır bandının genişliği $\frac{2}{\|w\|}$ formülü ile hesaplanmaktadır. Sınır düzlemlerinin optimum hiper düzleme uzaklığı ise $\frac{1}{\|w\|}$ formülü ile hesaplanmaktadır. Sınıfların en iyi şekilde ayrılabilmesi için düzlemler arasındaki mesafenin maksimum olması hedeflendiğinden dolayı (2.19)'daki amaç fonksiyonu ve (2.20)'deki kısıtlardan faydalanılmaktadır.

Amaç fonksiyonu: $min \frac{1}{2} \|w\|^2$ (2.19)

Kısıtlar: $y_i(W^T x_i + b) \geq +1, i \in \{1, \dots, n\}$ (2.20)

Amaç fonksiyonunun belirtilen kısıtlar için çözülebilmesi için Lagrange fonksiyonundan faydalanılmaktadır (Vapnik, 1999; Koçoğlu, 2017). Lagrange fonksiyonu kullanılarak düzenlenen denklem ile optimizasyon problemi kısıtlardan arındırılmaktadır (2.21).

$$L(w, b, a) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} a_i [y_i(w_i x_i + b) - 1] \quad (2.21)$$

Denklem (2.21)'de yer alan a_i değerleri Lagrange çarpanlarıdır. Lagrange fonksiyonu w ve b'ye karşı maksimize edilmeli ve $\alpha_i \geq 0$ durumu için maksimize edilmelidir (Vapnik, 1999). Denklem (2.21)'in çözülebilmesi için Karush-Kuhn-Tucker (KKT) koşullarından yararlanılmaktadır (2.22), (2.23).

$$\frac{\partial L(w_1, b, a)}{\partial w} \rightarrow W^T = \sum_{i=1}^{n} y_i a_i x_i \quad (2.22)$$

$$\frac{\partial L(w_1, b, a)}{\partial b} \rightarrow \sum_{i=1}^{n} y_i a_i = 0 \quad (2.23)$$

(2.22) ve (2.23) formülleri kullanılarak denklem (2.21) yeniden düzenlendiği takdirde denklem (2.24) bağıntısı elde edilmektedir.

$$L(w,b,a) = \sum_{i=1}^{n} a_i - \frac{1}{2}\sum_{i,j=1}^{n} y_i\, y_j\, a_i a_j \left(x_i{}^T x_j\right) \quad (2.24)$$

Denklem (2.24), $\alpha_i \geq 0$ olmak üzere bir optimizasyon problemi olarak çözülmektedir (Koçoğlu, 2017). H_0 optimum düzleminin elde edilebilmesi için L(w,b,α) dualinin maksimum olması gerekmektedir. L(w*,b*,α*) dualinin optimal noktasını sağlayan α_i değerleri w* ve b* parametrelerini belirler. x_i, α_i katsayılı destek vektörlerini, k destek vektörlerinin sayısını belirtmek üzere optimal çözüm parametreleri (2.25) ve (2.26) denklemleri ile bulunmaktadır.

$$w^* = \sum \alpha_i{}^* y_i x_i \ (i = 1, \dots, n) \quad (2.25)$$

$$b^* = \frac{1}{k}\left(\sum_{j=1}^{k}\left(\frac{1}{y_j} - x_j{}^T w^*\right)\right) \quad (2.26)$$

Tüm bu bilgilerden hareketle optimum hiper düzlemin hesaplanmasına yönelik nihai sınıflandırma fonksiyonu aşağıdaki gibidir (2.27).

$$f(x) = sign(\sum a_i{}^* y_i (x_i{}^T x) + b^*) \quad (2.27)$$

- **Doğrusal Olmayan Sınıflandırma Problemi**

Veri setinde bulunan örneklerin doğrusal olarak sınıflara ayrılamaması durumunu ifade eder. Bu durumun çözülebilmesi için soft hata değişkeni ve kernel (çekirdek) fonksiyonlarından faydalanılmaktadır (Koçoğlu, 2017). Doğrusal ayrılamayan örnekler için Denklem (2.20)'de belirtilen kısıtlar geçerliliğini kaybetmektedir. Bu durumun çözülebilmesi için formüle bir soft hata ξ_i değişkeni eklenir ve Denklem (2.28)'deki halini alır.

$$y_i(W^T x_i + b) \geq 1 - \xi_i \ (i = 1,2, \dots, n) \text{ ve } \xi_i \geq 0 \quad (2.28)$$

Burada amaç hatanın minimize edilmesidir. $\xi_i = 0$ ise xi örnekleri doğru sınıflandırılmıştır. $\xi_i \geq 1$ ise x_i örnekleri yanlış sınıflandırılmıştır. $0 < \xi_i < 1$ durumu verinin ayrılamadığını ifade etmekte olup x_i örneği sınır bantlarının içindedir. Maksimum sınırın tespit edilebilmesi için amaç fonksiyonuna ceza parametresi (C) eklenmektedir (Ayhan ve Erdoğmuş, 2014). Ceza parametresi Lagrange çarpanının alabileceği üst sınır değerini belirtir. C parametresi eklenmiş şekilde amaç fonksiyonu denklem (2.29)'daki gibidir.

$$L(W,\xi) = \frac{1}{2}\|W\|^2 + C(\sum_{i=1}^{n} \xi_i)^k \quad (2.29)$$

Ceza parametresinin eklenmesi ile Lagrange fonksiyonu denklem (2.30)'a dönüşür.

$$L(w,b,a,\beta,\xi) = \frac{1}{2}\|W\|^2 + C\left(\sum_{i=1}^{n}\xi_i\right) - \sum_{i=1}^{n} a_i[y_i(W^T x_i + b) - 1 + \xi_i] - \sum_{i=1}^{n}\beta_i\xi_i \quad (2.30)$$

KKT koşulları ile maksimize edilmek istenen dual Lagrange fonksiyonu ve kısıtları sırasıyla (2.31) ve (2.32)'deki gibidir.

$$\max L(a) = \sum_{i=1}^{n} a_i - \frac{1}{2}\sum_{i,j=1}^{n} y_i\, y_j\, a_i\, a_j {x_i}^T x_j \quad (2.31)$$

$$Kısıtlar: \sum_{i=1}^{n} a_i\, y_i = 0,\ 0 \le a_i \le C \quad (2.32)$$

Doğrusal ayrılamayan veri yapısının olduğu sınıflandırma problemlerinde uygulanan bir diğer yaklaşım haritalama yöntemidir (Kartal, 2015). Bu yöntemde çekirdek fonksiyonları kullanılarak veri seti doğrusal olarak ayrılabileceği yüksek boyutlu bir veri uzayına taşınmaktadır (Cortes ve Vapnik, 1995). Haritalama fonksiyonu veriyi bir üst uzaya taşır ve ϕ(x) şeklinde ifade edilmektedir (Akpınar, 2017). Çekirdek fonksiyonu denklem (2.33)'deki şekilde gösterilir.

$$K(x_i, x_j) = \Phi(x_i)^T \Phi(x_j) \quad (2.33)$$

Denklem (2.27)'de belirtilen formüle çekirdek fonksiyonu dahil edildiğinde denklem (2.34) elde edilmektedir.

$$f(x) = sign\left(\sum_{i=1}^{n} a_i\, y_i K(x, x_i)\right) \quad (2.34)$$

Tablo (2.1)'de doğrusal olmayan DVM problemlerinde kullanılan bazı çekirdek fonksiyonlarına yer verilmiştir.

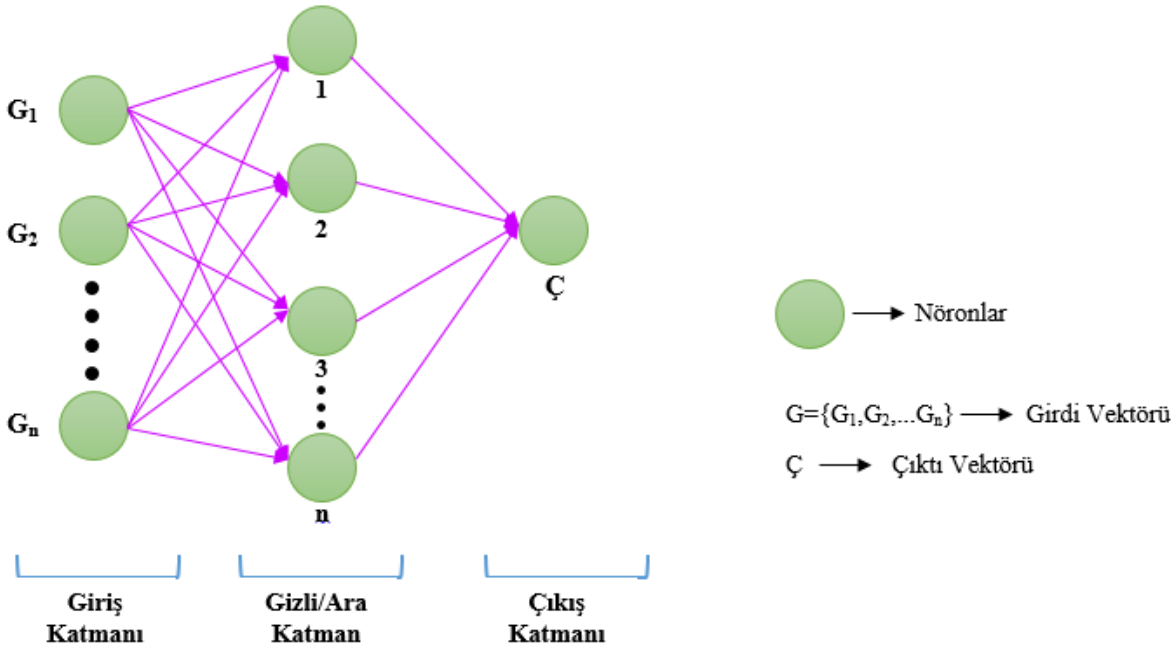
Tablo 2.1: Çekirdek fonksiyonları.

Çekirdek Fonksiyonu	Formül
Doğrusal	$K(x_i x_j) = {x_i}^T x_j$
Polinominal	$K(x_i x_j) = ({x_i}^T x_j + 1)^d$
Radial Basis Function (RBF)	$K(x_i x_j) = exp\left(-\gamma \left\| x_i - x_j \right\|^2\right), \gamma > 0$

2.3.4.5. Yapay Sinir Ağları

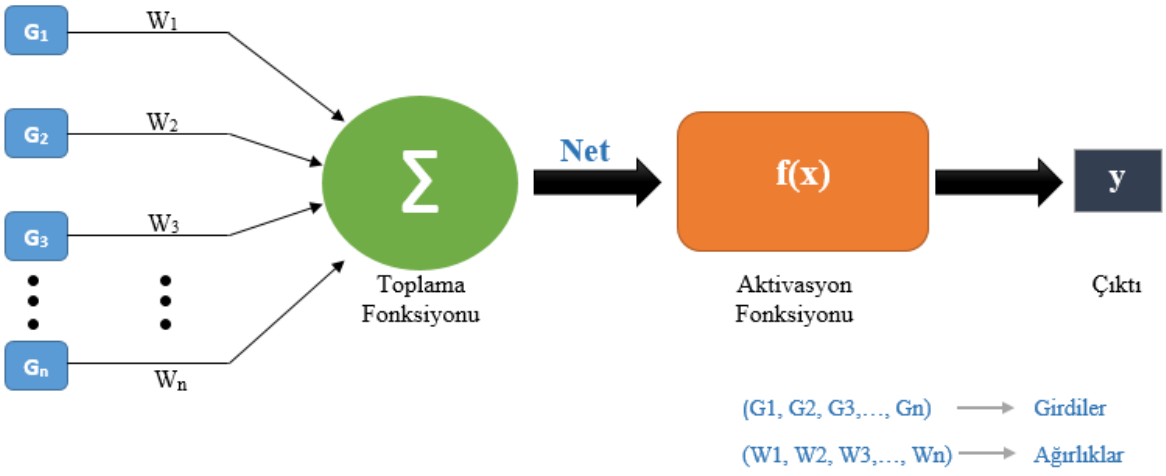
İnsan beyninin sinir ağı yapısından esinlenerek geliştirilmiş (Rosenblatt, 1958) ve birçok alanda başarıyla uygulanabilen bir yöntemdir (Alpaydın, 2014). Yüksek boyutlu veri ile çalışabilmesi, gürültüye karşı dayanıklı olması, verimlilik kaybı olmadan çoklu görevleri yapabilmesi gibi özellikleri bakımından avantajlı bir modeldir. Zaman ve kaynak kullanımının fazla olması yönünden dezavantajlı olabilmektedir.

Bir sinir ağı modeli temel olarak giriş katmanı, gizli (ara) katman/katmanlar ve çıkış katmanı olmak üzere üç bileşenden oluşmaktadır. Her katmanda nöron (proses elemanı) adı verilen yapay sinir hücreleri bulunmaktadır. Yapay sinir ağlarında (YSA) girdi bilgilerini çıktılara dönüştüren bir süreç işlemektedir. Veri setine ait örnekler giriş katmanı ile ağa gönderilirler. Daha sonra nöronlar arasında çeşitli bağlantılar kurularak bilinmeyen örneklerin hangi sınıf değerlerine ait olacağı belirlenir. Girdi değerleri nümerik yapıda olmalıdır. Örnek bir yapay sinir ağı modeli Şekil 2.8'de gösterildiği gibidir.



Şekil 2.8: Örnek yapay sinir ağı modeli (Gardner ve Dorling, 1998).

Her bir nöronun girdiler, ağırlıklar, toplama fonksiyonu, aktivasyon fonksiyonu ve çıktı olmak üzere beş bileşeni bulunmaktadır (Öztemel, 2012). Sürecin sinir hücrelerindeki işleyiş biçimi Şekil 2.9'da gösterildiği şekildedir.



Şekil 2.9: Yapay sinir hücresi işlem süreci (Zafari ve diğ., 2013).

Sinir hücresi bileşenleri aşağıdaki şekilde açıklanabilir:

- Girdiler: Sürecin ilk aşamasında sinir ağına dışarıdan gönderilen veri setindeki örneklere ait değerlerdir. YSA yapısında sinir hücreleri birbirlerine bağlı olduklarından dolayı bir nörona ait girdi aynı zamanda başka bir proses elemanından da gelebilmektedir (Yegnanarayana, 2009).
- Ağırlıklar: Sinir hücresine gelen girdinin önemini ve o hücre üzerindeki etkisini gösterir. Örneğin Şekil 2.9'daki W_1 ağırlığı, G_1 girdi değerinin hücre üzerindeki etkisini belirtmektedir.
- Toplama Fonksiyonu: Her bir nörona gelen net bilginin hesaplanması için kullanılan fonksiyondur. Nörona gelen girdiler ve girdilere ait ağırlıklar toplama fonksiyonu aracılığı ile çeşitli işlemlerden geçerek net girdi bilgisine (Net) dönüştürülmektedir. Tablo 2.2'de bazı toplama fonksiyonları ve formülleri gösterilmektedir (Çayıroğlu, 2015).

Tablo 2.2: Toplama fonksiyonları.

Toplama Fonksiyonu	**Formül**
Toplam (Ağırlıklı Toplam)	$Net = \sum_{i=1}^{n} G_i W_i$
Çarpım	$Net = \prod_{i=1}^{n} G_i W_i$
Maksimum	$Net = Max(G_i W_i \mid i = 1,2, \dots n)$
Minimum	$Net = Min(G_i W_i \mid i = 1,2, \dots n)$
Çoğunluk	$Net = \sum_{i=1}^{n} Sgn(G_i W_i)$

Ağırlıklı toplam fonksiyonu en sık kullanılan toplama fonksiyonlarından biridir.

- Aktivasyon Fonksiyonu: Toplama fonksiyonu ile hesaplanarak sinir hücresine gelen net girdi bilgisinin işlenerek çıktı değerinin hesaplanması için kullanılan fonksiyondur. Çeşitli aktivasyon fonksiyonları bulunmakta olup bazıları aşağıda verildiği gibidir (Tablo 2.3).

Tablo 2.3: Aktivasyon fonksiyonları (Zhang ve diğ., 2018).

Aktivasyon Fonksiyonu	Formül
Lineer	$f(x) = x$
Step	$f(x) = \begin{cases} 0, & x < 0 \; için \\ 1, & x \geq 0 \; için \end{cases}$
Sigmoid	$f(x) = \frac{1}{1 + e^{-x}}$
Hiperbolik Tanjant	$f(x) = \tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$
ReLU	$f(x) = \begin{cases} 0, & x < 0 \; için \\ x, & x \geq 0 \; için \end{cases}$

Yapay sinir ağlarının işleyiş biçimlerine göre farklı türleri bulunmaktadır. Katman sayıları, katmanlardaki nöron sayıları ve etkileşim biçimleri ağın yapısını belirleyen unsurlardır (Nabari, 2020). Bu tez çalışmasında Çok Katmanlı Algılayıcı modeli kullanıldığından dolayı detaylı olarak anlatılmaktadır.

- **Çok Katmanlı Algılayıcı (Multilayer Perceptron)**

Doğrusal olarak ayrılamayan veri kümelerinin sınıflandırılması problemine çözüm olarak Rumelhart ve diğ. (1986) tarafından geliştirilmiş en yaygın kullanılan YSA modellerinden biridir. Çok katmanlı algılayıcı yapısı giriş katmanı, iki veya daha fazla gizli katman ve çıkış katmanından oluşmaktadır. Sinir ağı mimarisinin (katman sayısı, nöron sayısı vb.) oluşturulması için kesin kurallar olmamakla birlikte deneme yanılma yöntemi ile belirlenmektedir.

Çok katmanlı algılayıcı modelinde geri yayılım algoritması kullanılmaktadır (Kotsiantis ve diğ., 2007). Geri yayılım yönteminde, girdiler rastgele ağırlıklar ile giriş katmanına verilir. Gizli katmanlardaki proses elemanlarında işlenerek çıkış katmanına ulaşırlar. Çıkış katmanında elde edilen çıktı değerleri ile beklenen (gerçek) çıktı değerleri karşılaştırılarak hatalar tespit edilir. Geriye doğru yayılım yapılarak ağırlıklar güncellenir. Hatalar minimize edilene kadar her iterasyonda (*epoch*) ağırlıklar güncellenerek yeni çıktı değerleri elde edilir.

Çok katmanlı algılayıcı aşağıda belirtilmekte olan öğrenme algoritmasını kullanır (Han ve diğ., 2012; Noriega, 2005).

- Girdileri (-1,+1) aralığında rastgele ağırlıklandır ve ağa ver.
- İlk iterasyonu çalıştır ve çıktı üret.
- Beklenen çıktı ile elde edilen çıktıyı karşılaştırarak hataları tespit et.

j indeksli çıktı hücresi için n indeksli eğitim verisi sonrası $b_j(n)$ beklenen çıktı değeri olmak üzere hata formülü Denklem 2.35'deki şekilde hesaplanmaktadır.

$$e_j(n) = b_j(n) - y_j(n) \tag{2.35}$$

Çıkış katmanındaki toplam hata denklem 2.36'daki gibi hesaplanmaktadır.

$$E(n) = \frac{1}{2}\sum_{j=1}^{k} {e_j}^2(n) \tag{2.36}$$

- Hataları geriye doğru düzelt.

Çıkış katmanının ağırlıklarını düzeltmek için (2.37)'den faydalanılır.

$$w_{ij}(t) = w_{ij}(t-1) + \left(\eta \delta_{pj} o_{pj}\right) \tag{2.37}$$

Bu formülde; $w_{ij}(t\text{-}1)$, t-1 zamanındaki i'ninci düğümden j'ninci düğüme kadar olan ağırlıkları; η, öğrenme oranını, δ_{pj} ise j düğümündeki p olayı için, hata terimini göstermekte olup denklem (2.38)'deki gibi hesaplanmaktadır.

$$\delta_{pj} = ko_{pj}\left(1 - o_{pj}\right)\left(t_{pj} - o_{pj}\right) \tag{2.38}$$

Gizli katmanlardaki ağırlıkların güncellenmesi için de (2.37) formülünden yararlanılmaktadır. Gizli katmanlar için δ_{pj} değerinin hesaplanması için denklem (2.39)'dan faydalanılmaktadır.

$$\delta_{pj} = ko_{pj}(1 - o_{pj}) \sum_k \delta_{pk} w_{jk} \tag{2.39}$$

2.3.4.6. Lojistik Regresyon

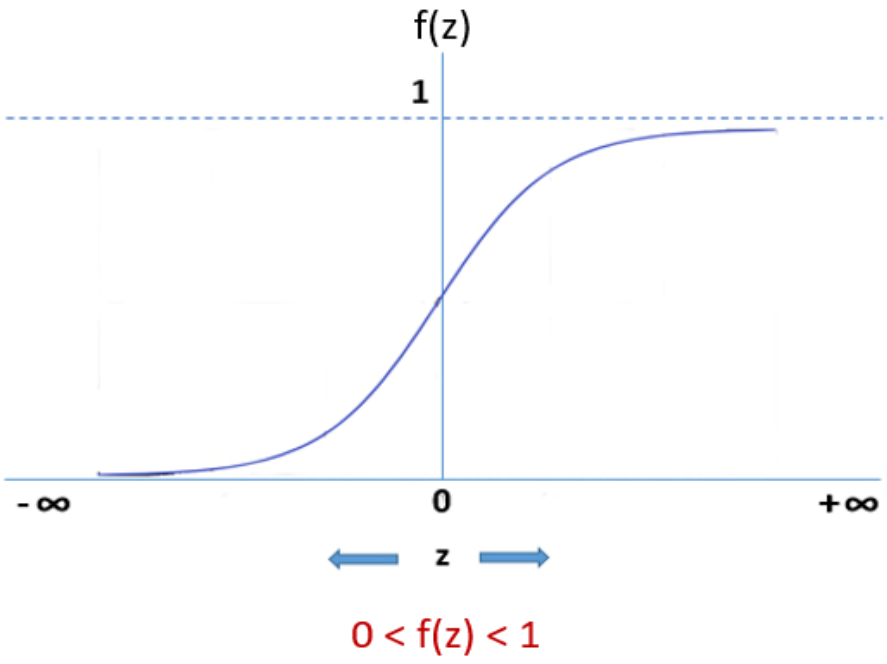
Bağımsız değişkenler ile bağımlı değişken arasında olasılıksal bir ilişki kurulması temeline dayanan sınıflandırma modelidir (Kleinbaum ve Klein, 2010; Hosmer ve diğ., 2013). Amaç, hedef niteliğin sınıf değerinin bulunması değil, belirli bir sınıf değerini alma olasılığının

hesaplanmasıdır. Lojistik regresyon hedef niteliğin yapısına göre ikili lojistik regresyon, sıralı (ordinal) lojistik regresyon ve çok kategorili (multinominal) lojistik regresyon olmak üzere üç farklı şekilde uygulanmaktadır (Karabulut ve Alpar, 2011; Kartal, 2015).

İkili lojistik regresyon için P olasılığı ile bağımsız değişkenler (X) arasındaki ilişki denklem (2.40)'da belirtildiği şekilde modellenebilir. Denklem tek bağımsız değişken olduğu varsayılarak ifade edilmektedir.

$$\pi(x) = P(Y = 1|X = x) = \frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}} = \frac{1}{1+e^{-(\beta_0+\beta_1 x)}} \tag{2.40}$$

Formülde X=x durumunda Y=1 olma olasılığı π(x) hesaplanmaktadır. "*e*" doğal logaritma tabanını ifade etmektedir (*e* = 2,718). Lojistik fonksiyonunun örnek gösterimi Şekil (2.10)'daki gibidir.



Şekil 2.10: Lojistik fonksiyonu (Kleinbaum ve Klein, 2010).

Bir olayın gerçekleşme olasılığının gerçekleşmeme olasılığına oranı üstünlük (odds) olarak tanımlanmaktadır ve denklem (2.41)'deki şekilde ifade edilir.

$$\frac{\pi(x)}{1-\pi(x)} = e^{\beta_0+\beta_1 x} \tag{2.41}$$

Doğrusal olmayan modelin doğrusal hale getirilmesi için üstünlüklerin (odds) doğal logaritması alınarak logit dönüşüm uygulanmaktadır (Denklem 2.42). Modelin doğrusal hale getirilmesi tahmin açısından kolaylık sağlamaktadır.

$$logit\ \pi(x) = \ln\left(e^{\beta_0+\beta_1 x}\right) = \beta_0 + \beta_1 x \tag{2.42}$$

Lojistik regresyon modelinde β katsayılarının tahmini için çeşitli yöntemler bulunmaktadır. Maksimum olabilirlik yöntemi, yeniden ağırlıklandırılmış en küçük kareler yöntemi, minimum logit ki-kare yöntemi bu yöntemlerden bazılarıdır (Erişlik, 2020). En sık kullanılan yöntem ise maksimum olabilirlik yöntemidir. Bu metotta gözlenen örneğin olasılığı bilinmeyen katsayıların bir fonksiyonu olarak belirtilir. Fonksiyonu maksimum yapan değerler parametreler için en çok olabilirlik kestiricileridir. Bir başka deyişle bir olayın gerçekleşme olasılığı maksimize edilmeye çalışılmaktadır (Ulutürk Akman, 2004).

Birden fazla bağımsız değişkenin olduğu durumlarda tek değişkenli model genelleştirilmektedir. Bağımsız değişkenler $X=\{x_1,x_2,\ldots,x_n\}$ olmak üzere denklem (2.43) elde edilir.

$$\pi(x) = P(Y = 1|x_1, x_2, \ldots, x_n) = \frac{e^{\left(\beta_0+\sum_{i=1}^{n}\beta_i x_i\right)}}{1+e^{\left(\beta_0+\sum_{i=1}^{n}\beta_i x_i\right)}} = \frac{1}{1+e^{-\left(\beta_0+\sum_{i=1}^{n}\beta_i x_i\right)}} \tag{2.43}$$

Üstünlük formülü (2.44) şeklinde ifade edilir.

$$\frac{\pi(x)}{1-\pi(x)} = e^{\left(\beta_0+\sum_{i=1}^{n}\beta_i x_i\right)} \tag{2.44}$$

Logit dönüşüm uygulanarak model doğrusal hale getirilir (2.45).

$$logit\ \pi(x) = \ln\left(\frac{\pi(x)}{1-\pi(x)}\right) = \beta_0 + \sum_{i=1}^{n}\beta_i x_i \tag{2.45}$$

Hedef niteliğin ikiden fazla kategoriye sahip olduğu durumlarda bir dizi ikili lojistik regresyon uygulanarak her grup birbiri ile karşılaştırılır. Bağımlı değişkenin ilk veya son sınıf değerinin referans kategori olarak belirlenmesi modelin yorumlanmasını kolaylaştırır (Osborne, 2017; Erişlik, 2020). Referans kategori ile diğer kategoriler sırasıyla ikili lojistik regresyon ile karşılaştırılmaktadır. Bu durumda bağımlı değişkenin sınıf değerleri $Y=\{Y_1,Y_2,\ldots Y_n\}$ olmak üzere n-1 adet logit fonksiyonu hesaplanmaktadır.

2.3.4.7. Naive Bayes

Bayes teoremini esas alan istatistiksel bir sınıflandırma yöntemidir (Bayes ve Price, 1763). Sınıf değeri bilinmeyen bir örneğin hedef niteliğin sınıf değerlerine ait olma olasılıklarının hesaplanması prensibine dayalı olarak çalışmaktadır (Han ve Kamber, 2006). X=(x_1, x_2,…,x_n) örnekler kümesi, S= (s_1,s_2,…,s_n) hedef niteliğe ait sınıflar kümesi olsun. Temel olarak bilinmeyen X örneğinin sınıf değeri aşağıdaki formül ile hesaplanmaktadır (2.46).

$$P(S_i|X) = \frac{P(S_i)P(X|S_i)}{P(X)} \quad (2.46)$$

Formülde $P(S_i|X)$ sonrasal olasılıktır ve Naive Bayes sınıflandırma modelinde temel hedef bu değerin maksimize edilmesidir. P(X) tüm sınıflar için eşittir ve sabit olarak kabul edilir. $P(X|S_i)$, X'in Si ile ilişkilendirilmesi sonucu oluşan koşullu olasılıktır. X'in sahip olduğu bağımsız niteliklere göre hedef niteliğin sınıf değerlerinden birine ait olma olasılığını belirtir. Bir diğer ifadeyle nitelik değeri x_j olan S_i sınıfına ait örneklerin sayısının, S_i sınıfına ait tüm örneklerin sayısına oranı olarak tanımlanabilir. X'e bağlı her bağımsız nitelik değeri için ayrı ayrı olasılıklar hesaplanmalıdır. Örneğin Şekil 2.11'deki gibi tek bağımsız değişkene sahip bir veri seti olsun.

Hava Durumu	Basketbol Oynama Durumu
Güneşli	Evet
Güneşli	Hayır
Yağmurlu	Hayır
Bulutlu	Evet
Bulutlu	Hayır
Güneşli	Evet
Yağmurlu	Hayır
Yağmurlu	Evet
Güneşli	Hayır
Güneşli	Evet

Şekil 2.11: Naive Bayes örnek veri seti.

Veri setinde havanın yağmurlu olduğunda basketbol oynanıp oynanmayacağı Naive Bayes modeli ile tahmin edilmek istendiğinde hesaplama aşağıda belirtildiği şekilde yapılmaktadır.

- $P(S_i)$=P(Evet) = 6/10

- P(X)=P(Yağmurlu)=3/10
- P(X|S_i)=P(Yağmurlu|Evet)= 1/6
- P(S_i|X)=P(Evet|Yağmurlu)= (6/10)(1/6)/(3/10) =0.33 olarak hesaplanacaktır. Sonucun sıfıra daha yakın olması sebebiyle Naive Bayes modeli tahmini hayır olarak gerçekleşecektir.

Çok sayıda niteliğin bulunduğu veri setlerinde P(X|S_i) olasılığının hesaplanması zor olmaktadır. Bu nedenle niteliklerin birbirinden bağımsız olduğu kabul edilerek sınıf koşullu bağımsızlık varsayımı (*assumption of class conditional independence*) uygulanmaktadır (denklem 2.47).

$$P(X|S_i) = \prod_{k=1}^{n} P(X_k|S_i) = P(X_1|S_i)P(X_2|S_i) \dots P(X_n|S_i) \quad (2.47)$$

P(X|S_i) niteliğin kategorik veya nümerik olma durumlarına göre farklı şekilde hesaplanmaktadır. Eğer nitelik kategorik yapıda ise denklem 2.29'daki şekilde kolaylıkla hesaplanabilmektedir. Eğer nitelik nümerik yapıda ise niteliğin Gauss dağılımına (Gaussian Distribution) sahip olduğu varsayılır ve (2.48) formülündeki şekilde hesaplanır.

$$P(X_k|S_i) = f(X_k, \mu_i, \sigma_i) = \frac{1}{\sigma_{S_i}\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{X_k-\mu_{S_i}}{\sigma_{S_i}}\right)^2} \quad (2.48)$$

Formülde μ nümerik değerin gösterdiği sınıfa ait ortalama değeri, σ ise nümerik değerin gösterdiği sınıfa ait standart sapmayı ifade etmektedir.

2.3.5. Model Değerlendirme

2.3.5.1. Model Geçerleme Yöntemleri

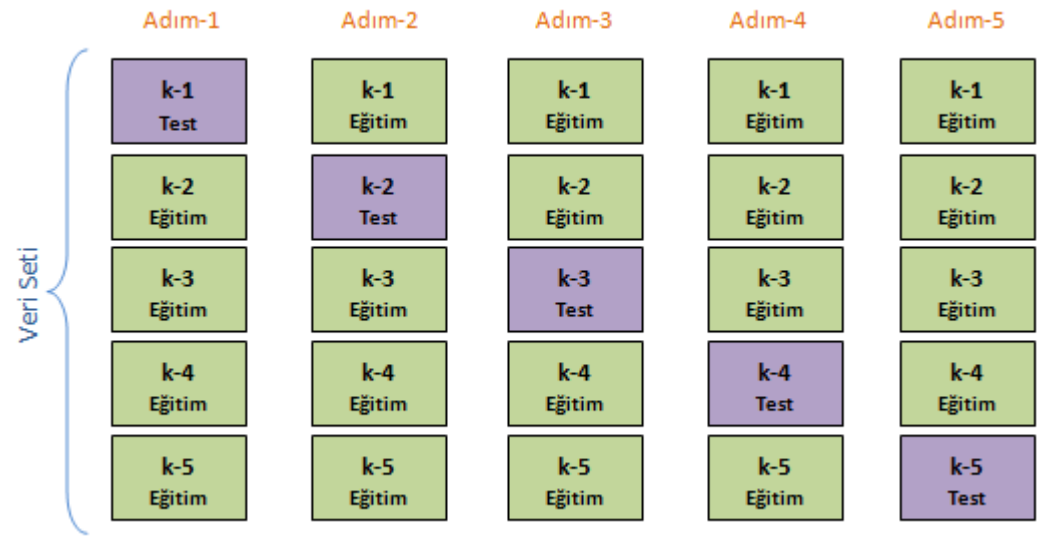
Gözetimli öğrenme modellerinin performans değerlendirmesinin yapılabilmesi için veri setinin eğitim ve test kümelerine bölünmesi gerekmektedir (Nordman, 2011). Eğitim kümesi, sınıflandırıcının öğrenmesi ve hedef niteliğin tahminine yönelik model kurabilmesi amacıyla kullanılmaktadır. Test kümesi ise oluşturulan modelin başarısının test edilebilmesi amacıyla kullanılmaktadır. Verinin ne şekilde bölüneceğine karar vermek için çeşitli yaklaşımlar bulunmaktadır. Hold-out, çapraz geçerleme ve bootstrap literatürde sıkça bahsedilen yöntemler arasındadır (Akpınar, 2017).

Hold-out (belirli oranlarda bölme) yönteminde, veri belirlenen oranlarda eğitim ve test kümelerine ayrılır. Genellikle ⅔ eğitim kümesi, ⅓ test kümesi olarak belirlenmektedir (Kohavi,

1995; Emre, 2017). Veri setinin yapısına bağlı olarak bölme oranları değişkenlik gösterebilmektedir (Han ve diğ., 2012). Hold-out yönteminin bazı dezavantajları olabilmektedir. Seçilen eğitim ve test kümeleri veri setini tam anlamıyla temsil edemeyebilirler. İki kümedeki sınıf dağılımları oransal olarak eşit olmayabilir ya da bazı sınıf değerlerinin hiç olmaması gibi durumlarla karşılaşılabilir. Bu gibi durumları engellemek için;

- Eğitim ve test kümelerinin sınıf dağılımları kontrol edilebilir.
- Tabakalı örnekleme (*stratified hold-out*) yöntemi tercih edilebilir.
- Farklı eğitim ve test kümeleri ile birden fazla tekrarlanarak gerçekleştirilen tekrarlı hold-out (*repeated hold-out*) yöntemi kullanılabilir.

Çapraz geçerleme yöntemleri k-kat çapraz geçerleme (*k-fold cross validation*) ve birini dışarıda bırak (*leave-one-out* - LOOCV) olmak üzere iki şekilde uygulanabilmektedir (Saharidis ve diğ., 2011). k-kat çapraz geçerleme yönteminde veri setindeki örnekler k eşit parçaya ayrılır. Ayrılan parçalardan 1 tanesi test kümesi, geriye kalan (k-1) parçalar ise eğitim kümesi olarak tanımlanarak model k kez çalıştırılır. İşlem her tekrarlandığında test kümesi değişir. Örneğin k=5 olarak seçildiğinde; veri seti {k1, k2, k3, k4, k5} olmak üzere 5 parçaya ayrılır. İlk adımda k1 test kümesi, {k2, k3, k4, k5} parçaları eğitim kümesi olarak belirlenir. İkinci adımda k2 test kümesi, {k1, k3, k4, k5} parçaları eğitim kümesi olarak belirlenir, tüm parçalar test kümesi olana kadar işlem devam eder (Şekil 2.12). Model doğruluğu k-kez elde edilen doğruluk oranlarının ortalaması alınarak hesaplanır.



Şekil 2.12: 5-kat çapraz geçerleme (Rogers ve Girolami, 2011).

Birini dışarıda bırak çapraz geçerleme yönteminde ise her adımda veri setine ait yalnızca 1 adet örnek test kümesi, kalan örnekler eğitim kümesi olarak tanımlanmaktadır. Veri setindeki kayıtların çok az olduğu durumlarda kullanılmaktadır. Örnek sayısının fazla olduğu durumlarda, işlem adımı sayısı çok fazla olacağı için süre ve maliyet açısından dezavantajlı olmaktadır (Kim, 2009)

Bootstrap yöntemi genellikle veri setindeki örnek sayısının az olduğu durumlarda kullanılmaktadır (Alpaydın, 2014). n adet örnekten oluşan veri seti için n defa rastgele örnek seçimi yapılarak eğitim kümesi oluşturulur ve seçilmeyen örnekler test kümesi olarak tanımlanır. Eğitim kümesi oluşturulurken seçilen örnekler elenmez ve dolayısı ile aynı örnekler birden fazla seçilebilmektedir. Eğitim kümesinde tekrar eden kayıtlar bulunurken test kümesindeki kayıtlar tekrar etmezler.

2.3.5.2. Model Performans Değerlendirme Ölçütleri

Veri üzerinde makine öğrenmesi algoritmaları uygulandıktan sonra, modellerin başarım oranlarının ölçülebilmesi için başvurulan ölçütlerdir. Model performans değerlendirme ölçütleri kullanılarak uygulanan modelin ne ölçüde performans sağladığı görülebilmektedir. Bu sayede;

- Geliştirilen modelin uygulamaya geçirilip geçirilmeyeceğine,

- Birden fazla modelin geliştirildiği durumlar için, hangi modelin daha başarılı olduğuna karar verilebilmektedir.

Sınıflandırma modelleri için çeşitli performans değerlendirme ölçütleri geliştirilmiştir. Değerlendirme ölçütlerinden sık kullanılanlar doğruluk (*accuracy*), kesinlik (*precision*), duyarlılık (*recall*), F-ölçütü (*F-Score*) ve Roc eğrisi (*reciever operator characteristics curve*) altında kalan alanı ifade eden Roc-Auc skoru sayılabilir (Kantardzic, 2011). Bu ölçütler karmaşıklık matrisinden (*confusion matrix*) yararlanılarak hesaplanmaktadır. Karmaşıklık matrisi her bir sınıfa ait doğru ve yanlış tahmin edilen örnek sayılarını gösteren tablodur. İkili sınıflandırmaya ait karmaşıklık matrisi Tablo 2.4'de gösterilmektedir (Akpınar, 2017).

Tablo 2.4: Karmaşıklık matrisi.

Karmaşıklık Matrisi (*Confusion Matrix*)		**Tahmin Edilen Değerler**		
		Pozitif	Negatif	Toplam
Gerçek Değerler	Pozitif	**TP** **doğru pozitif**	**FN** **yanlış negatif**	P^g
	Negatif	**FP** **yanlış pozitif**	**TN** **doğru negatif**	N^g
	Toplam	P^t	N^t	$P^g + N^g = P^t + N^t$

P^g: Gerçek değeri pozitif olan örneklerin sayısı
N^g: Gerçek değeri negatif olan örneklerin sayısı
P^t: Tahmin değeri pozitif olan örneklerin sayısı
N^t: Tahmin değeri negatif olan örneklerin sayısı

Doğru pozitif (*true positive - TP*) : Gerçekte pozitif sınıfa ait olup model tarafından da pozitif olarak tahmin edilen örnek sayısını belirtir.

Doğru negatif (*true negative - TN*): Gerçekte negatif sınıfa ait olup model tarafından da negatif olarak tahmin edilen örnek sayısını belirtir.

Yanlış pozitif (*false positive - FP*) : Gerçekte negatif sınıfa ait olup model tarafından pozitif olarak tahmin edilen örnek sayısını belirtir.

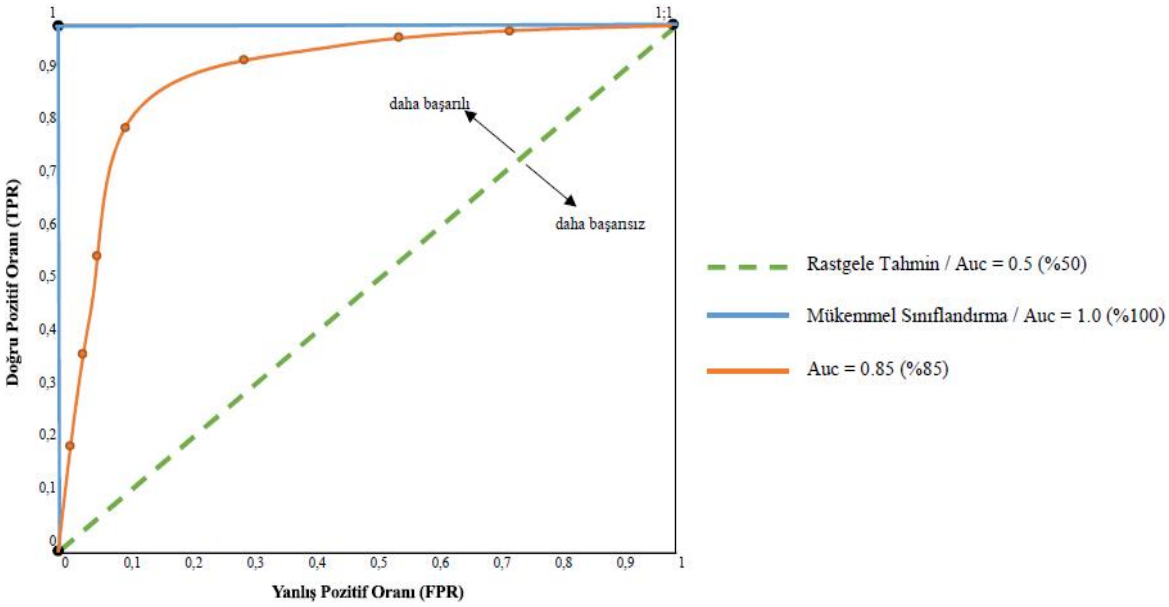
Yanlış negatif (*false negative - FN*): Gerçekte pozitif sınıfa ait olup model tarafından negatif olarak tahmin edilen örnek sayısını belirtir.

Doğru pozitif (TP) ve doğru negatif (TN) değerleri sınıflandırma modeli tarafından doğru sınıflandırılan örnek sayılarını, yanlış pozitif ve yanlış negatif değerleri ise sınıflandırma modeli tarafından yanlış sınıflandırılan örnek sayılarını ifade etmektedirler. Karmaşıklık matrisinden yararlanılarak hesaplanan değerlendirme ölçütleri formülleri Tablo 2.5'deki gibidir.

Tablo 2.5: Model değerlendirme ölçütleri (Gorunescu, 2011; Han ve diğ., 2012).

Değerlendirme Ölçütü	Açıklama	Formül
Doğruluk (*accuracy*)	Doğru tahmin edilen örnek sayısının, tüm örneklerin sayısına oranını ifade eder.	$\frac{TP + TN}{TP + FP + TN + FN}$
Kesinlik (*precision*)	Doğru tahmin edilen pozitif örnek sayısının, pozitif olarak tahmin edilen örneklerin sayısına oranını ifade eder.	$\frac{TP}{TP + FP}$
Duyarlılık / doğru pozitif oranı (*recall/true positive rate*)	Doğru tahmin edilen pozitif örnek sayısının, gerçekte pozitif olan örneklerin sayısına oranını ifade eder.	$\frac{TP}{TP + FN}$
Belirleyicilik / doğru negatif oranı (*specifity / true negative rate*)	Doğru tahmin edilen negatif örnek sayısının, gerçekte negatif olan örneklerin sayısına oranını ifade eder.	$\frac{TN}{TN + FP}$
F-Ölçütü (*F-Score*)	Kesinlik ve duyarlılık ölçütlerinin harmonik ortalamasıdır.	$\frac{2\ x\ kesinlik\ x\ duyarlılık}{kesinlik\ +\ duyarlılık}$
Yanlış pozitif oranı (*false positive rate*)	Yanlış tahmin edilen negatif örnek sayısının, gerçekte negatif olan örneklerin sayısına oranını ifade eder.	$\frac{FP}{FP + FN}$
Yanlış negatif oranı (*false negative rate*)	Yanlış tahmin edilen pozitif örnek sayısının, gerçekte pozitif olan örneklerin sayısına oranını ifade eder	$\frac{FN}{FN + TP}$

Tablo 2.5'deki ölçütlere ek olarak ROC eğrisi, makine öğrenmesi algoritmalarının başarısını belirlemek için kullanılan alternatif bir değerlendirme ölçütüdür (Huang ve Ling, 2005). 2. Dünya savaşı yıllarında radar sinyallerinin sınıflandırılması amacıyla geliştirilen, 1969 yılından itibaren medikal tanı testlerinde kullanılmakta olan (Ertorsun ve diğ., 2009) ROC eğrisi bir olasılık eğrisidir ve eğrinin altında kalan alan Auc (*area under the curve*) olarak tanımlanmaktadır. Auc skoru sınıflandırma için sınıflar arası ayrılabilirliğin ölçüsünü ifade eder. 0-1 arası değer almakta olup değer 1'e yaklaştıkça sınıflandırma performansı artmaktadır. Roc eğrisinde X ekseni yanlış pozitif oranını (FPR) ve Y ekseni doğru pozitif oranını (TPR) temsil etmektedir (Fawcet, 2006). Örnek bir Roc eğrisi Şekil 2.13'de gösterilmektedir.



Şekil 2.13: Örnek ROC eğrisi (Fogarty ve diğ., 2005).

Auc skoru, olası sınıflandırma eşikleri için toplu bir performans değerlendirmesi yapmaktadır. Belirli eşik değerlere göre FPR ve TPR oranları hesaplanarak grafiğe yerleştirilir. Eşik değerler F-ölçütünü en büyük yapan değerlere göre belirlenmektedir. Her eşik değere karşılık gelen FPR ve TPR ikilisi ROC uzayında bir nokta ile temsil edilir (Fawcett, 2006; Kartal, 2015). Noktaların birleştirilmesi ile eğri oluşturulmaktadır. Auc değerinin 0.5 olması sınıflandırmanın rastgele yapıldığını, altında olması tahmin başarısının çok kötü olduğunu, 1'e yakın olması ise başarılı sınıflandırmayı ifade etmektedir.

2.3.6. Uygulama

Geliştirilen makine öğrenmesi modelinin uygulamaya geçirildiği adımdır. Model değerlendirme ölçütleri kullanılarak veri üzerinde uygulanan makine öğrenmesi algoritmalarının başarım oranları hesaplanır. Model değerlendirmesi sonucunda en başarılı performans gösteren tahmin modeli seçilerek model uygulamaya konulmaktadır.

2.4. LİTERATÜR TARAMASI

Bu tez çalışmasında öğrenci başarılarının önceden tahmin edilmesi problemi ile ilgili olarak ulusal ve uluslararası alanda yapılmış olan benzer çalışmalara ait yayınlar araştırılmıştır. Bu kapsamda Web of Science ve Google Akademik veri tabanlarında 01/01/2000-31/12/2020

tarihleri aralığında *Education, Machine Learning, Prediction, Classification, Academic Success* anahtar kelimeleri ile yapılan literatür taraması sonucunda eğitim verisi ile yapılan çalışmalara ait 32 adet yayın incelenmiştir.

İncelenen çalışmalarda kullanılan sınıflandırma algoritmaları analiz edildiğinde en fazla kullanılan algoritmanın 25 çalışmada kullanılan Karar Ağaçları algoritması olduğu görülmektedir. Sırasıyla 23 çalışmada YSA, 18 çalışmada Naive Bayes, 13 çalışmada Rastgele Orman, 13 çalışmada DVM, 10 çalışmada Knn, 9 çalışmada Lojistik Regresyon algoritmaları kullanılmıştır. Bu algoritmaların haricinde Sıralı Minimal Optimizasyon, REPtree, OneR, ZeroR ve Doğrusal Regresyon algoritmaları kullanılan diğer modeller arasındadır.

Çalışmalar kullanılan araçlar ve geliştirme ortamları açısından incelendiğinde WEKA programının yoğun olarak kullanıldığı görülmektedir. Kullanılan araçlar sırası ile WEKA (11) SPSS (5), MATLAB (3), R-Studio (3), Python-Scikit Learn (2), Rapid Miner (2), Orange (1) şeklindedir.

Performans değerlendirme ölçütü olarak doğruluk ölçütü kullanımı ağırlıktadır. 28 çalışmada doğruluk ölçütünden faydalanılmıştır. Diğer değerlendirme ölçütlerinin kullanım sayıları; F-Skoru (16), ROC-Auc (13), karmaşıklık matrisi (11), kesinlik (9), duyarlılık (8)'dir. TPR, FPR, Kappa Skoru, hassasiyet ve ortalama kare hatası çalışmalarda kullanılan diğer değerlendirme ölçütleridir.

Minaei-Bigdoli ve diğ. (2003), çalışmalarında uzaktan eğitim görmekte olan lisans öğrencilerine ait veriyi kullanarak akademik başarı tahminine yönelik makine öğrenmesi modelleri geliştirmişlerdir. Çalışmada kullanılan veri seti 10 bağımsız nitelik ve 227 kayıttan oluşmaktadır. Hedef nitelik olan sene sonu not ortalaması 2 seviyeli, 3 seviyeli ve 9 seviyeli olmak üzere üç farklı şekilde derecelendirilmiştir. Veri indirgeme yöntemi olarak temel bileşen analizinden (PCA) yararlanılmıştır. Karar Ağaçları (C5.0, CART, QUEST, CRUISE), K-En Yakın Komşu, Naive Bayes, Genetik Algoritmalar ve Yapay Sinir Ağları modelleri MATLAB yazılımı ortamında uygulanmıştır. Veriyi bölme yöntemi olarak 10-kat çapraz geçerleme metodu kullanılmış olup model değerlendirme için doğruluk ölçütü ve ROC eğrisinden faydalanılmıştır. En yüksek başarım oranı Genetik Algoritmalar ile elde edilmiştir.

Aydın (2007), çalışmasında uzaktan eğitim lisans öğrencilerinin akademik başarı durumlarının tahmini için sınıflandırma modelleri geliştirmiş ve öğrenci profillerinin belirlenmesine yönelik

kümeleme çalışması gerçekleştirmiştir. Analiz edilen veri seti öğrencilere ait demografik, sosyoekonomik ve not bilgilerinden oluşan 129 adet bağımsız nitelik ve toplam 180.554 kayıttan oluşmaktadır. Hedef nitelik geçti-kaldı şeklinde ikili kategorik yapıdadır. SPSS yazılımı ortamında gerçekleştirilen çalışmada özellik seçimi uygulanarak sınıflandırma için 11 adet bağımsız nitelik kullanılmıştır. Veriyi bölme yöntemi olarak hold-out (%50/%50) ve 5-kat çapraz geçerleme uygulanmıştır. Karar Ağaçları (C5.0, CHAID, CART, Quest), Lojistik Regresyon ve Yapay Sinir Ağları çalışmada kullanılan sınıflandırma modelleridir. Model değerlendirme için doğruluk ve karmaşıklık matrisinden faydalanılmıştır. En başarılı sonuçlar Karar Ağaçları C5.0 algoritması ile elde edilmiştir. Geçmiş başarı ve e-öğrenme kullanım günceleri özniteliklerinin akademik başarının tahmin edilmesinde en belirleyici nitelikler olduğu sonucuna varılmıştır.

Ön lisans öğrencilerine yönelik akademik başarı tahmini için model geliştirilen bir çalışmada (Ogor, 2007), 78 bağımsız değişken ve 1.369 kayıttan oluşan veri seti kullanılmıştır. Bağımsız değişkenler demografik ve ağırlıklı olarak not bilgilerinden oluşmaktadır. Hedef nitelik ise zayıf-orta-iyi olmak üzere 3 seviyeli kategorik yapıdadır. SPSS yazılımı aracılığı ile gerçekleştirilen çalışmada modelleme için Karar Ağaçları (CART, C.5.0, CHAID) ve Yapay Sinir Ağları algoritmaları uygulanmıştır. Model değerlendirme için doğruluk ölçütünden faydalanılmıştır. Karar Ağaçları C5.0 algoritması en yüksek başarım oranını vermiştir. Kümülatif not ortalaması özniteliği en belirleyici değişken olarak tespit edilmiştir.

Cortez ve Silva (2008), ortaokul öğrencilerine ait demografik, sosyoekonomik, geçmiş dönem not bilgileri ve okul bilgilerini kullanarak akademik başarı tahminine yönelik sınıflandırma ve regresyon modelleri geliştirmişlerdir. Çalışmada kullanılan veri seti 32 bağımsız nitelikten oluşmaktadır. Hedef nitelik alanları ise Matematik ve Portekizce dersleri yılsonu ortalamaları olarak belirlenmiştir. Hedef nitelikler sınıflandırma problemi için 2 seviyeli ve 5 seviyeli olmak üzere iki farklı şekilde düzenlenmiştir. Matematik veri seti 395 örnek, Portekizce veri seti ise 649 örnek ile analiz edilmiştir. Modelleme için Karar Ağaçları, Yapay Sinir Ağları, Destek Vektör Makineleri ve Rastgele Orman algoritmaları R programlama dili kullanılarak uygulanmıştır. Veriyi bölme metodu olarak çapraz geçerleme yönteminden faydalanılmıştır. Model değerlendirme yöntemi olarak doğruluk ölçütü ve karmaşıklık matrisi kullanılmıştır. Matematik dersi başarı tahmini için Naive Bayes algoritması, Portekizce dersi başarı tahmini için ise Karar Ağaçları algoritması en yüksek performası göstermiştir. Geçmiş dönem not

bilgileri, anne eğitim durumu, baba eğitim durumu, anne meslek, baba meslek ve anne-baba alkol tüketimi öznitelikleri başarıya en fazla etki eden faktörler olarak belirlenmiştir.

Ramaswami ve Bhaskaran (2009), ortaöğretim öğrencilerine ait demografik, sosyoekonomik ve akademik verilerinin kullanılarak lise seviyesi genel başarı durumlarının ikili sınıflandırma ile tahmin edilmesini hedefledikleri çalışmalarında özellik seçim yöntemlerinden faydalanmışlardır. 32 bağımsız nitelik ve 1.969 örnekten oluşan veri seti kullanılmıştır. Çalışmada 5 farklı özellik seçim yöntemi uygulanmış ve her bir seçim yöntemi ile ayrı ayrı modeller oluşturularak sonuçlar karşılaştırılmıştır. Modelleme için Naive Bayes algoritması kullanılmış ve en iyi sonucu veren özellik seçim yönteminin OneR yöntemi olduğu görülmüştür. Doğruluk, F-ölçütü ve ROC eğrisi model değerlendirmede yararlanılan metriklerdir. Ortaokul not ortalaması, ikincil müfredat türü, baba meslek, öğrenim dili, okul lokasyonu, okula ulaşım şekli ve cinsiyet öznitelikleri özellik seçimi sonucunda elde edilen belirleyici değişkenler olmuştur.

Delen (2010), çalışmasında lisans öğrencilerinin okula devam etme/ayrılma durumlarını veri madenciliği yöntemleri ile tahmin etmeyi amaçlamıştır. Veri seti 39 bağımsız nitelik ve 23.000 kayıttan oluşmakta olup bağımlı değişken ikili kategorik yapıya sahiptir. Veriyi bölme yöntemi olarak 10-kat çapraz geçerleme kullanılmıştır. Modelleme için tekil ve kolektif yöntemler birlikte kullanılarak sonuçlar karşılaştırılmıştır. Değerlendirme ölçütü olarak doğruluk ve karmaşıklık matrisi kullanılmıştır. Çalışmada kullanılan veri için kolektif yöntemlerin tekil yöntemlere göre daha yüksek başarım oranı sağladığı tespit edilmiştir. Destek Vektör Makineleri tahmin için en yüksek performansı göstermiştir. Güz dönemi not ortalaması, bahar dönemi kredisi, burs alma durumu, medeni durum ve okula kayıt türü değişkenlerinin en belirleyici öznitelikler olduğu belirtilmiştir.

Lisans öğrencilerinin sosyoekonomik özelliklerinin kullanılarak uzaktan eğitim sistemindeki başarı durumlarının ikili sınıflandırma (geçti-kaldı) ile tahmin edildiği bir çalışmada aynı zamanda başarıya en çok etki eden niteliklerin belirlenmesi amaçlanmıştır (Kovacic, 2010). 9 bağımsız nitelik ve 450 kayıttan oluşan bir veri seti kullanılmış, veri SPSS ve Statistica araçları ile analiz edilmiştir. Çalışmada ki-kare yöntemi ile özellik seçimi yapılarak bağımsız nitelikler arasında en yüksek skora sahip olan 3 nitelik kullanılmıştır. Model oluşturmak için Karar Ağaçları CART ve CHAID algoritmalarından faydalanılmıştır. Karmaşıklık Matrisi ve Doğruluk, çalışmada kullanılan model değerlendirme ölçütleridir. Model değerlendirme

sonucunda CART algoritması daha yüksek performans göstermiştir. Başarıya en fazla etki eden niteliklerin etnisite, eğitim programı ve kurs bloğu değişkenleri olduğu belirtilmiştir.

Şen ve diğ. (2012), tarafından yapılan çalışmada ortaokul öğrencilerine ait veri kullanılarak ortaöğretime geçiş sınavı sonuçlarının veri madenciliği yöntemleri kullanılarak tahmini hedeflenmiştir. Sınav notları 5 aşamalı olarak derecelendirilmiş ve tahmin için sınıflandırma algoritmaları kullanılmıştır. Toplam 25 adet bağımsız nitelik kullanılmış olup, bu niteliklerden 9 adedi öğrencilerin ailevi ve demografik özelliklerinden, 16 adedi ise okuldaki derslerine ait geçmiş not ortalamalarından oluşmaktadır. Yapay Sinir Ağları, Lojistik Regresyon, Karar Ağaçları ve Destek Vektör Makineleri model kurmak için kullanılan yöntemlerdir. Model değerlendirme kriteri olarak doğruluk ölçütünden yararlanılmıştır. En yüksek sınıflandırma başarısı Karar Ağaçları modeli ile elde edilmiştir. Önceki yıllar not ortalaması, yerleştirme sınavı puanı, deneyim, burs alma durumu ve kardeş sayısı değişkenleri akademik başarı tahmininde en belirleyici nitelikler olarak belirlenmiştir.

Göker (2012), lise öğrencilerinin demografik, sosyoekonomik ve geçmiş akademik başarı özelliklerine ait veri kullanarak üniversite giriş sınavı sonuçlarının tahminine yönelik bir çalışma yapmıştır. Çalışmada 39 adet bağımsız nitelik ve 220 örnek kullanılmış olup Naive Bayes, Karar Ağaçları (J48) ve Yapay Sinir Ağları (RBF Network) ile hem sınıflandırma hem regresyon modelleri kurulmuştur. Bağımlı değişken regresyon için 0-500 puan aralığında, sınıflandırma için ise kazandı-kazanamadı şeklinde ikili kategorik yapıdadır. Özellik seçimi yapılarak bağımsız nitelik sayısı 20'ye düşürülmüştür. Karmaşıklık matrisi ve doğruluk sınıflandırma modelleri için, ortalama kare hatası ise regresyon modelleri için kullanılan değerlendirme ölçütleridir. Model değerlendirme sonuçlarına göre Naive Bayes algortiması en yüksek performansı gösteren model olmuştur. Tahmin için; 12., 11., 10. sınıf not ortalamaları,dershaneye gidip gitmediği, devamsızlık, 9. Sınıf not ortalaması, sürekli ilaç kullanım durumu, ilköğretim diploma notu, kardeş sayısı, baba meslek, kendine ait odasının olup olmaması ve aile gelir durumu değişkenlerinin belirleyici nitelikler olduğu belirtilmiştir.

Şengür (2013), lisans öğrencilerinin ders notları verisinin kullanılarak mezuniyet notlarının öngörülebilmesine yönelik bir çalışma gerçekleştirmiştir. Çalışmada 49 bağımsız nitelik ve 127 örnekten oluşan veri seti kullanılmıştır. Hedef nitelik olan mezuniyet notu alanı 0-4 arası değerlerden oluşmakta olup tahmin için regresyon yöntemi kullanılmıştır. Yapay Sinir Ağları ve Karar Ağaçları algoritmaları kullanılarak sonuçlar karşılaştırılmıştır. Başarım değerlendirme

kriteri olarak Ortalama Kare Hatası ve Ortalama Mutlak Hata ölçütleri kullanılmıştır. Model değerlendirme sonucunda Yapay Sinir ağları ile daha başarılı sonuçlar elde edildiği belirtilmiştir.

Acharya ve Sinha (2014), çalışmalarında Lisans koleji öğrencilerinin demografik, sosyoekonomik verisini kullanılarak 1. dönem başarı puanlarını tahmin etmişlerdir. Kullanılan veri seti 34 adet bağımsız nitelik ve 413 kayıttan oluşmaktadır. Hedef nitelik 6 seviyeli kategorik yapıdadır. Çalışmada öznitelik seçimi yapılarak en iyi sonucu veren 8 bağımsız nitelik karar niteliğinin tahmini için kullanılmıştır. Tahmin için Naive Bayes, Karar Ağaçları, Çok Katmanlı Yapay Sinir Ağları ve K-En Yakın Komşu yöntemlerinden faydalanılmıştır. En yüksek sınıflandırma başarısı Karar Ağaçları modeli ile elde edilmiştir. 1. Dönem not ortalaması, aile gelir durumu, ortaöğretim not ortalaması, hangi dine mensub olduğu, devamsızlık, sosyal sınıf, ortaöğretim okul türü değişkenlerinin başarıyı en fazla etkileyen nitelikler olduğu sonucuna varılmıştır.

Kaur ve diğ. (2015), lise seviyesindeki öğrencilere ait veriyi kullanarak akademik performansı iyi olmayan öğrencilerle ilişkili faktörlerin tanımlanmasını, yavaş öğrenenlerin belirlenerek eğitim kalitesinin iyileştirilmesini ve böylece öğretmenlerin öğrenci performanslarını geliştirmede bireysel olarak kendilerine yardımcı olabilmelerini amaçlamışlardır. Karar niteliği ikili sınıflandırma ile tahmin edilmiştir. Çalışmada kullanılan veri seti 13 adet bağımsız değişken ve 152 kayıttan oluşmaktadır. Başarı durumunun tahmini için Çok Katmanlı Yapay Sinir Ağları, Naive Bayes, Karar Ağaçları (J48), Azaltılmış Hata Oranlı Karar Ağaçları (REPTree) ve Sıralı Minimal Optimizasyon yöntemleri kullanılmıştır. En yüksek sınıflandırma performansı Çok Katmanlı Yapay Sinir Ağları modeli ile elde edilmiştir.

Naser ve diğ. (2015), çalışmalarında lisans öğrencilerine ait demografik ve ders notları verisini kullanarak akademik performans tahminini amaçlamışlardır. Veri seti 10 bağımsız nitelik ve 150 kayıttan oluşmakta olup bağımlı değişken 0-4 aralığında kategorik yapıdadır. Veriyi bölme işlemi için Hold-out ve çapraz geçerleme yöntemleri birlikte kullanılmışlardır. Çalışmada Yapay Sinir Ağları ile modelleme yapılmış ve %84,6 sınıflandırma başarısı elde edilmiştir. Model performans değerlendirme için doğruluk ölçütünden faydalanılmıştır.

Guo ve diğ. (2015), çalışmalarında lise öğrencilerine ait demografik, geçmiş not bilgileri, okul bilgileri, psikolojik durumları (hazırbulunuşluk düzeyi, ilgi ve dikkat durumu vb.) ve takviye

kurs durumları verisini kullanarak akademik başarı tahminine yönelik model geliştirmişlerdir. Çalışmada kullanılan veri seti 100 farklı okuldan alınan toplam 120.000 örnekten oluşmaktadır. 14'den fazla bağımsız nitelik kullanılmış olup öznitelik sayısı tam olarak belirtilmemiştir. Bağımlı değişken 5 seviyeli kategorik yapıda bulunmaktadır. ANSI C ve Theano yazılımları ile gerçekleştirilen çalışmada modelleme için Naive Bayes, Yapay Sinir Ağları, Destek Vektör Makineleri ve SPPN adını verdikleri derin öğrenme modeli kullanılmıştır. Model performans değerlendirmesi için doğruluk ve kesinlik ölçütlerinden faydalanılmıştır. En başarılı sonuçlar SPNN derin öğrenme modeli ile elde edilmiştir.

Özdemir (2016), lise öğrencilerinin demografik, sosyoekonomik verilerinin kullanılarak sene sonu ağırlıklı not ortalamalarının ikili sınıflandırma ile tahmininin amaçlandığı çalışmasında 61 bağımsız nitelik ve 1.706 örnekten oluşan veri seti kullanılmıştır. R-Studio ortamında gerçekleştirilen çalışmada veriyi bölme yöntemi olarak hold-out ve k-fold çapraz geçerleme yöntemlerinin birlikte kullanımı tercih edilmiştir. Tahmin modelleri Karar Ağaçları, Destek Vektör Makineleri, Naive Bayes, K-En Yakın Komşu ve Lojistik Regresyon algoritmaları kullanılarak geliştirilmiştir. Doğruluk, ROC eğrisi, F-Ölçütü ve hata oranı model değerlendirme için kullanılan yöntemlerdir. En yüksek tahmin başarısı Karar Ağaçları (C4.5) algoritması ile elde edilmiştir. Başarı tahmini için en belirleyici özniteliklerin devamsızlık, motivasyon, günlük ders çalışma süresi, sınıf mevcudu, sınıf tekrarı olup olmaması ve cinsiyet değişkenleri olduğu belirtilmiştir.

Devasia ve diğ. (2016), Hindistan'da lise öğrencilerine ait veriyi analiz ederek akademik başarı tahminini ve başarıya etki eden faktörlerin belirlenmesini hedefledikleri çalışmalarında 19 bağımsız değişken ve 700 örnekten oluşan veri seti kullanmışlardır. Özellik seçimi yapılarak modellemede kullanılan bağımsız değişken sayısı azaltılmıştır. Karar Ağaçları, Yapay Sinir Ağları ve Naive Bayes modelleri uygulanmış olup en başarılı sonuç Naive Bayes algoritması ile elde edilmiştir. Kullanılan veri seti için başarıya en fazla etki eden öznitelikler geçmiş dönem notları, anne ve babanın eğitim durumları ve ailenin gelir durumu olarak tespit edilmiştir.

Gök (2017), ortaokul öğrencilerine ait ailevi, demografik ve okul kaynaklı etmenlere dayalı niteliklerden oluşan veri setini kullanarak öğrencilerin Türkçe, Matematik ve dönem sonu genel başarı ortalamalarının tahmin edilmesini amaçlamıştır. Veri seti 24 adet bağımsız nitelik ve 1.492 kayıttan oluşmaktadır. Hedef nitelikler 0-5 arası kategorize edilmiş olup sınıflandırma yöntemi ile tahmin edilmiştir. Doğrusal Regresyon, Destek Vektör Makineleri, Rastgele

Orman, K-En Yakın Komşu, Naive Bayes ve Lojistik Regresyon algoritmaları kullanılmıştır. Genel başarı tahmini için Lojistik Regresyon, Türkçe ve Matematik dersleri başarı tahmini için ise Destek Vektör Makineleri en iyi sonuçların alındığı modeller olmuştur. Anne baba eğitim durumları, gelir durumu, kardeş sayısı, günlük uyku süresi, ders çalışma süresi ve ailenin ders konusundaki desteği değişkenlerinin sınıflandırma için en belirleyici nitelikler olduğu belirtilmiştir.

Asif ve diğ. (2017), lisans öğrencilerinin akademik başarı tahminini hedefledikleri bir çalışmalarında 46 nitelik ve 210 öğrenci verisinden oluşan veri seti kullanmışlardır. Karar niteliği kategorik yapıda olup 1-5 arası değerlere sahiptir. Çalışmada hold-out yöntemi kullanılarak veri eğitim ve test kümelerine ayrılmıştır. Tahmin için Rapid Miner yazılımı ortamında Karar Ağaçları, Rastgele Orman, Naive Bayes, K-En Yakın Komşu ve Yapay Sinir Ağları yöntemleri ile modeller oluşturulmuştur. Karmaşıklık matrisi, doğruluk ve duyarlılık model performans değerlendirmesi için kullanılan yöntemlerdir. En yi sınıflandırma başarısı Naive Bayes modeli ile elde edilmiş olup geçmiş not bilgilerine ait değişkenlerin en belirleyici öznitelikler olduğu belirtilmiştir.

Depren ve diğ. (2017), çalışmalarında Uluslararası Matematik ve Fen Eğilimleri Araştırması (TIMSS) veri setini kullanarak Türkiye'deki 8. sınıf öğrencilerinin Matematik dersleri başarılarının tahmin edilmesine yönelik model geliştirmişlerdir. Veri seti 11 bağımsız değişken ve 6.250 kayıttan oluşmaktadır. Hedef nitelik başarılı-başarısız olmak üzere ikili kategorik yapıdadır. Modelleme için Karar Ağaçları, Rastgele Orman, Naive Bayes, Lojistik Regresyon ve Yapay Sinir Ağları algoritmaları WEKA yazılımı kullanılarak uygulanmıştır. TPR, FPR, Kesinlik, MCC, F-ölçütü, ROC-Auc Skoru ve Kappa skoru çalışmada kullanılan model değerlendirme ölçütleridir. Çalışmada en yüksek başarım oranı Lojistik Regresyon algoritması ile elde edilmiştir. Anne-baba eğitim durumları, öğrencinin motivasyonu, evdeki eğitim olanakları, evdeki ders çalışma desteği ve öğrencinin derse katılımı değişkenleri model tahmini için en belirleyici nitelikler olarak bulunmuştur.

Aydemir (2017), çalışmasında meslek yüksekokulu öğrencilerinin genel not ortalamalarını ve mezuniyet yıllarını tahmin etmeyi amaçlamıştır. Veri setinde, 1.387 öğrencinin demografik, ailevi durumlarına ait 10 nitelik ve üniversite giriş puanı olmak üzere toplam 11 adet bağımsız nitelik bulunmaktadır. Akademik ortalama ve mezuniyet yılı nitelikleri ise tahmin edilmek istenen hedef nitelikler olarak belirlenmiştir. Akademik ortalama hedef niteliği kategorik

yapıda olup 5 farklı sınıf değerine sahiptir. WEKA yazılımı ortamında gerçekleştirilen çalışmada Karar Ağaçları, Naive Bayes, K-En Yakın Komşu, Sıralı Minimum Optimizasyon ve Yapay Sinir Ağları modelleri kullanılmıştır. Doğruluk, kesinlik, duyarlılık, Roc-Auc Skoru, F-Ölçütü, TPR ve FPR çalışmada kullanılan model değerlendirme ölçütleridir. Sıralı Minimum Optimizasyon algoritması ile geliştirilen model genel not ortalaması tahmini için, Karar Ağaçları J48 algoritması ile geliştirilen model ise mezuniyet yılı tahmini için en yüksek performansı sergilemiştir. Giriş yılı, giriş puanı, mezuniyet yılı, anne-babanın hayatta olma durumları, anne-baba eğitim durumları ve lise diploma notu değişkenleri akademik başarıya en fazla etki eden öznitelikler olarak belirlenmiştir.

Lisans öğrencilerinin akademik performanslarının önceden ölçülmesinin hedeflendiği bir çalışmada (Adejo ve Connoly, 2018), klasik sınıflandırma yöntemleri ile kolektif yöntemlerin birlikte kullanılarak bir hibrit model geliştirilmesinin başarı tahmininde ve risk faktörlerinin belirlenmesinde daha etkili olabileceği düşünülmüştür. 74 bağımsız nitelik ve 141 kayıttan oluşan veri setinin kullanıldığı çalışma SPSS ve Rapid Miner yazılımları ile gerçekleştirilmiştir. Model kurmak için Karar Ağaçları, Yapay Sinir Ağları, Destek Vektör Makineleri ve kolektif yöntemler kullanılmıştır. Model performans değerlendirme için doğruluk, kesinlik, sınıflandırma hata oranı ve F-ölçütünden faydalanılmıştır. Önerilen hibrit modelin akademik başarı tahmininde başarılı olduğu ve bu alanda kullanılabileceği belirtilmiştir.

Öğrenci verileri kullanılarak akademik başarının tahmini ve başarıya etki eden faktörlerin belirlenmesine yönelik bir çalışmada (Uzel ve diğ., 2018), veri üzerinde sınıflandırma ve birliktelik kuralları yöntemlerini birlikte uygulamışlardır. Çalışmada kullanılan veri seti ilkokul, ortaokul ve lise seviyesindeki öğrencilere ait 16 bağımsız nitelik ve 480 kayıttan oluşmaktadır. Bağımsız değişken üçlü kategorik yapıda olup sınıflandırma için Karar Ağaçları, Rastgele Orman, Naive Bayes ve Yapay Sinir Ağları yöntemleri WEKA yazılımı ortamında uygulanmıştır. Bölme yöntemi olarak 10-kat çapraz geçerleme yönteminden yararlanılmıştır. Doğruluk, kesinlik, duyarlılık, F-Ölçütü ve karmaşıklık matrisi modellerin değerlendirilmesi için kullanılan yöntemlerdir. En yüksek başarım oranı Yapay Sinir Ağları ile elde edilmiştir. Çalışmada elde edilen sonuçlara göre veri setinde bulunan özniteliklerden cinsiyet, devamsızlık, ailenin okul memnuniyeti ve ek kurs alma durumunun akademik başarıya diğer özniteliklere göre daha fazla etki ettiği görülmüştür.

Polyzou ve Karypis (2019), çalışmalarında lisans öğrencilerinin bir sonraki dönem akademik başarılarının öngörülebilmesini ve akademik başarı bakımından risk altında olan öğrencilerin tespit edilebilmesini hedeflemişlerdir. Bu sayede üniversite yönetimine ve öğrenci danışmanlarına önceden bilgi verilerek ilgili öğrenciler için gerekli tedbirlerin alınabileceği düşünülmüştür. Çalışmada 42 adet bağımsız nitelik 8 farklı kategoriye ayrılarak kullanılmış olup her bir kategorinin karar niteliğine etkisi incelenerek başarım oranları karşılaştırılmıştır. Karar niteliği üniversite harf not sistemi ölçeğine göre olup ikili kategorik şekle dönüştürülmüştür. Karar Ağaçları, Çok Katmanlı Yapay Sinir Ağları, Rastgele Orman ve Gradyan Artırma (Gradient Boosting) algoritmaları ile modelleme yapılmıştır. Sonuçların değerlendirilmesi için F-Ölçütü, Roc Eğrisi ve Kesinlik ölçütleri kullanılmıştır. En yüksek sınıflandırma başarısını Gradyan Artırma modeli göstermiştir. Geçmiş not bilgileri ve dönemlik ders yükü bilgilerine ait değişkenlerin başarı tahmininde önemli rol oynadıkları belirtilmiştir.

Jalota ve Agrawal (2019), tarafından yapılan çalışmada uzaktan eğitim lisans öğrencilerine ait veri kullanılarak akademik performanslarının tahmini amaçlanmıştır. Çalışmada kullanılan veri seti; öğrencilerin akademik geçmişi, demografik özellikleri ve psikolojik alt yapıları olmak üzere 3 farklı gruba ayrılmış 16 bağımsız nitelik ve 480 kayıttan oluşmaktadır. Çalışma Karar Ağaçları, Çok Katmanlı Yapay Sinir Ağları, Rastgele Orman ve Destek Vektör Makineleri yöntemleri kullanılarak WEKA yazılımı ortamında gerçekleştirilmiştir. Modellerin değerlendirilmesinde doğruluk ölçütü kullanılmış ve en yüksek doğruluk oranı Yapay Sinir Ağları ile elde edilmiştir.

Youssef ve diğ. (2019), çalışmalarında hazırlık amaçlı uzaktan çevrimiçi lisans eğitim programları için öğrencilerin bırakma eğilimlerinin erken tespitini hedeflemişlerdir. Bırakma eğilimlerinin öngörülebilmesi için 37 bağımsız nitelik ve toplam 4.757 kayıttan oluşan veri seti makine öğrenmesi yöntemleri ile analiz edilmiştir. Hedef nitelik bırakma riski olan, başarısız olma ihtimali yüksek, başarılı olma ihtimali yüksek şeklinde 3 kategoriye ayrılmış ve tahmin için sınıflandırma algoritmalarından yararlanılmıştır. Python programlama dili ile gerçekleştirilen çalışmada model kurmak için Destek Vektör Makineleri, K-En Yakın Komşu, Karar Ağaçları, Naive Bayes, Rastgele Orman ve Lojistik Regresyon algoritmaları kullanılmıştır. Modellerin performans başarımları ROC Eğrisi, F-ölçütü, kesinlik ve duyarlılık ölçütleri ile değerlendirilmiştir. En başarılı sonuçlar Rastgele Orman algoritması ile elde edilmiştir.

Filiz (2019), çalışmasında Uluslararası Matematik ve Fen Eğilimleri Araştırması (TIMSS) veri setini kullanarak Türkiye'deki 8. sınıf öğrencilerinin Fen Bilimleri ve Matematik dersleri başarılarının tahmin edilmesini hedeflemiştir. Çalışmada Fen Bilimleri ve Matematik olmak üzere iki farklı veri seti kullanılmıştır. Fen Bilimleri veri seti 4.481, Matematik veri seti ise 4.577 kayıttan oluşmaktadır. Her iki veri setinde de 35 bağımsız değişken ve 1 bağımlı değişken bulunmaktadır. Bağımlı değişken "başarılı-başarısız" olmak üzere ikili kategorik yapıda olup tahmin için sınıflandırma yöntemi kullanılmıştır. Özellik seçimi uygulanarak Fen bilimleri veri seti için 13, Matematik veri seti için 10 nitelik sınıflandırma modellerinde kullanılmıştır. WEKA yazılımı ile gerçekleştirilen analizlerde sınıflandırma yöntemi olarak Destek Vektör Makineleri, K-En Yakın Komşu, Karar Ağaçları, Naive Bayes, Rastgele Orman, Lojistik Regresyon ve Yapay Sinir Ağları algoritmalarından faydalanılmıştır. Doğruluk, TPR, FPR, kesinlik, F-ölçütü, ROC eğrisi, Kappa skoru ve Mathews Korelasyon Katsayısı (MCC) çalışmada kullanılan model başarım değerlendirme ölçütleridir. En başarılı sonuçlar Fen Bilimleri veri seti için Lojistik Regresyon algoritması, Matematik veri seti için ise Destek Vektör Makineleri ile elde edilmiştir. Evdeki eğitim olanakları, öğrencinin özgüveni ve öğrencinin ek kurs alma durumu akademik başarı için en belirleyici nitelikler olarak tespit edilmiştir.

Salal ve diğ. (2019), çalışmalarında Portekiz'deki ortaokul öğrencilerinin demografik, sosyoekonomik, okul bilgileri ve geçmiş dönemlere ait not bilgilerini kullanarak akademik başarı tahmini için model geliştirmişlerdir. Veri seti 32 bağımsız nitelik ve 649 kayıttan oluşmaktadır. Sene sonu not ortalaması hedef nitelik olup 1-5 arası kategorik yapıdadır. WEKA yazılımı ile gerçekleştirilen çalışmada Karar Ağaçları, Naive Bayes, Lojistik Regresyon, Rastgele Orman, REPTree, OneR ve ZeroR algoritmaları ile modelleme yapılmıştır. Özellik seçimi yapılarak bağımsız değişken sayısı 10'a indirgenmiştir. Modellerin performans değerlendirmesi için doğruluk ölçütünden faydalanılmıştır. Model değerlendirme sonucunda en yüksek performansı gösteren modeller OneR ve REPTree algortimaları olmuştur. 2. Dönem not ortalaması, 1. Dönem not ortalaması, okul ismi ve çalışma süresi değişkenlerinin sınıflandırma başarısı üzerinde belirleyici nitelikler olduğu belirtilmiştir.

Lisans öğrencilerinin akademik performanslarının tahmin edildiği bir çalışmada, öğrencilerin sosyoekonomik geçmişleri ve ulusal sınav puanları verisinden oluşan veri kümesi kullanılmıştır (Lau ve diğ., 2019). Kullanılan veri seti 11 bağımsız nitelik ve 1.000 kayıttan oluşmakta olup

hedef nitelik olan genel not ortalaması ikili kategorik yapıdadır. WEKA aracı ortamında gerçekleştirilen çalışmada modelleme için Yapay Sinir Ağlarından faydalanılmıştır. ROC Eğrisi, hata oranı, kesinlik ve duyarlılık performans değerlendirme için kullanılan ölçütlerdir. Çalışmada ayrıca istatistiksel analiz yapılarak başarıya en çok etki eden faktörlerin tespit edilmesi amaçlanmış olup hedef nitelik ile en yüksek korelasyonun İngilizce dersi sınav notu arasında olduğu görülmüştür.

Gomes ve Jelihovschi (2020), çalışmalarında ortaöğretim öğrenci verisi kullanarak öğrencilerin ulusal sınav başarılarını regresyon yöntemi ile tahmin etmeyi ve başarıya etki eden nitelikleri tespit etmeyi hedeflemişlerdir. Çalışmada toplam 53 nitelik ve 3.670.089 kayıttan oluşan oldukça büyük bir veri seti kullanılmıştır. Veri seti öğrencilerin demografik özellikleri, sosyoekonomik özellikleri ve geçmiş akademik başarılarına ait niteliklerden oluşmaktadır. Veriyi bölme yöntemi olarak 3-kat çapraz geçerleme kullanılmış olup tahmin için Karar Ağaçları CART algoritması ile modelleme yapılmıştır. Okul türü, aile gelir durumu ve sınav motivasyonu değişkenlerinin sınav başarısı tahmini için belirleyici nitelikler olduğu belirtilmiştir.

Adekitan ve Salau (2020), lise seviyesinden mezun olan öğrencilerin üniversiteye kabul kriteri puanlarının sınıflandırma ve regresyon yöntemleri ile tahmin edilmesini amaçlamışlardır. Aynı zamanda bağımsız niteliklerin karar niteliğine olan etkileri tespit edilmeye çalışılmıştır. Çalışmada 2.413 öğrenci verisi kullanılmıştır. Orange yazılımı kullanılarak gerçekleştirilen çalışmada Karar Ağaçları, Yapay Sinir Ağları, Rastgele Orman ve Naive Bayes kullanılan makine öğrenmesi modelleridir. Modellerin uygulanması sonucunda en yüksek sınıflandırma başarısı (%53,2) Yapay Sinir Ağları modeli ile elde edilmiştir. Aşırı Örnekleme (*Oversampling*) metodu ile veri boyutu artırılarak başarım oranı %79,8'e çıkarılmış olmasına rağmen bu yöntemin çok sağlıklı olmadığı ve mümkün olduğunca kaçınılması gerektiği belirtilmiştir. Aynı zamanda etnik köken niteliğinin akademik başarıya önemli bir etkisinin olmadığı tespit edilmiştir.

Tomasevic ve diğ. (2020), açık öğretim üniversitesi öğrencilerine ait verinin kullanılarak final notlarının tahminini hedefledikleri çalışmalarında sınıflandırma ve regresyon yöntemlerini birlikte kullanmışlardır. 32.593 öğrenciye ait veri seti 16 bağımsız nitelikten oluşmaktadır. Karar niteliği olan final notu alanı ise sınıflandırma problemi için ikili kategorik yapıda olup regresyon problemi için 0-100 puan aralığındadır. Bağımsız nitelikler demografik, etkileşim ve

performans verileri olarak 3 kategoriye ayrılmış ve kategorilerin model başarısına etkileri ayrı ayrı değerlendirilmiştir. Destek Vektör Makineleri, Yapay Sinir Ağları, K-En Yakın Komşu ve Bayesian Lineer Regresyon modelleme için kullanılan yöntemlerdir. Sınıflandırma için performans değerlendirme kriteri olarak F-ölçütü, regresyon için ise Ortalama Kare Hatası ölçütü kullanılmıştır. Her iki durumda da Yapay Sinir Ağları modeli en yüksek başarım oranlarına ulaşmıştır. Demografik özeliklerin model başarısına anlamlı bir etki etmediği sonucuna varılmıştır. Öğrencilerin geçmiş akademik başarıları ve derslere katılım durumlarına ait özniteliklerin tahmin için belirleyici değişkenler olduğu belirtilmiştir.

Ortaokul öğrencilerinin akademik başarı tahminlerine yönelik yapılan bir çalışmada (Abbasoğlu, 2020), e-okul veri tabanından elde edilen 27 bağımsız değişken ve 1395 kayıttan oluşan veri seti kullanılmıştır. "Yıl sonu genel not ortalaması" niteliği bağımlı değişken olup 0-5 arası kategorik değerlere sahiptir. WEKA yazılımı ortamında gerçekleştirilen çalışmada sınıflandırma modelleri Destek Vektör Makineleri, Yapay Sinir Ağları, K-En Yakın Komşu, Rastgele Orman, Lojistik Regresyon ve Naive Bayes algoritmalarından faydalanılarak geliştirilmiştir. Veri öncelikle veri indirgeme işlemi uygulanmadan analiz edilmiş, daha sonra özellik seçimi ve temel bileşen analizi uygulanarak analiz edilmiş ve sonuçlar karşılaştırmalı olarak sunulmuştur. Model değerlendirme yöntemi olarak doğruluk ölçütü kullanılmıştır. Çalışmada geliştirilen modellerden Lojistik Regresyon algoritması %64.13 doğruluk oranı ile en başarılı sonucu vermiştir. Yaş, devamsızlık, anne-babanın birlikte veya ayrı olması, anne ve baba eğitim durumları, ailenin gelir durumu, kendine ait odasının olup olmadığı ve Fen Bilimleri kursu alma durumunun akademik başarı üzerinde en belirleyici olduğu sonucuna varılmıştır.

Literatür araştırması kapsamında incelenen yayınlara ait detaylar Tablo 2.6'da gösterilmektedir.

Tablo 2.6: Literatür incelemesi.

Çalışma Başlığı / Kaynak	**Çalışma Türü**	**Veri Seti**	**Yöntemler**	**Kullanılan Araçlar**	**Değerlendirme Ölçütleri**
Using Data Mining to Predict Secondary School Student Performance / Cortez ve Silva (2008)	Makale	• 32 Nitelik • 1.044 Kayıt	• Karar Ağaçları • Yapay Sinir Ağları • Destek Vektör Makineleri • Rastgele Orman	• R Programlama Dili	• Doğruluk • Karmaşıklık Matrisi
A Study on Feature Selection Techniques in Educational Data Mining / Ramaswami ve Bhaskaran (2009)	Makale	• 32 Nitelik • 1.969 Kayıt	• Naive Bayes	• WEKA	• Doğruluk • F-Ölçütü • ROC Eğrisi
A comparative analysis of machine learning techniques for student retention management / Delen (2010)	Makale	• 39 Nitelik • 23.000 Kayıt	• Karar Ağaçları • Yapay Sinir Ağları • Destek Vektör Makineleri • Lojistik Regresyon • Rastgele Orman • Boosted Trees • Information Fusion	• Belirtilmemiş	• Doğruluk • Karmaşıklık Matrisi
Predicting and analyzing secondary education placement-test scores: A data mining approach / Şen ve diğerleri (2012)	Makale	• 25 Nitelik • 5.000 Kayıt	• Karar Ağaçları • Yapay Sinir Ağları • Destek Vektör Makineleri • Lojistik Regresyon	• Belirtilmemiş	• Doğruluk • Karmaşıklık Matrisi • Hassasiyet

Tablo 2.6: Literatür incelemesi (devam).

Çalışma Başlığı / Kaynak	**Çalışma Türü**	**Veri Seti**	**Yöntemler**	**Kullanılan Araçlar**	**Değerlendirme Ölçütleri**
Early Prediction of Students Performance using Machine Learning Techniques / Acharya ve Sinha (2014)	Makale	• 34 Nitelik • 413 Kayıt	• Karar Ağaçları • Yapay Sinir Ağları • Naive Bayes • K-En Yakın Komşu	• WEKA	• Kappa Değeri • F-ölçütü
Predicting Student Performance Using Artificial Neural Network: in the Faculty of Engineering and Information Technology / Naser ve diğerleri (2015)	Makale	• 10 Nitelik • 150 Kayıt	• Yapay Sinir Ağları	• Belirtilmemiş	• Doğruluk
Makine Öğrenmesi Yöntemleri İle Akademik Başarının Tahmin Edilmesi / Gök (2017)	Makale	• 24 Nitelik • 1492 kayıt	• Doğrusal Regresyon • Destek Vektör Makineleri • Rastgele Orman • K-En Yakın Komşu • Naive Bayes • Lojistik Regresyon	• Belirtilmemiş	• Doğruluk • Karmaşıklık Matrisi • Ortalama Kare Hatası
Analyzing undergraduate students' performance using educational data mining / Asif ve diğerleri (2017)	Makale	• 46 Nitelik • 210 Kayıt	• Karar Ağaçları • Naive Bayes • Yapay Sinir Ağları • Rastgele Orman • K-En Yakın Komşu	• Rapid Miner	• Doğruluk • Duyarlılık • Karmaşıklık Matrisi

Tablo 2.6: Literatür incelemesi (devam).

Çalışma Başlığı / Kaynak	**Çalışma Türü**	**Veri Seti**	**Yöntemler**	**Kullanılan Araçlar**	**Değerlendirme Ölçütleri**
Identifying the Classification Performances of Educational Data Mining Methods: A Case Study for TIMSS / Depren ve diğerleri (2017)	Makale	• 11 Nitelik • 6250 Kayıt	• Karar Ağaçları • Rastgele Orman • Naive Bayes • Lojistik Regresyon • Yapay Sinir Ağları	• WEKA	• TPR • FPR • Kesinlik • MCC • F-ölçütü • ROC-Auc Skoru • Kappa skoru
Predicting student academic performance using multi-model heterogeneous ensemble approach / Adejo ve Connoly (2018)	Makale	• 74 Nitelik • 141 Kayıt	• Karar Ağaçları • Yapay Sinir Ağları • Destek Vektör Makineleri • Kolektif Modeller (Belirtilmemiş)	• SPSS • Rapid Miner	• Doğruluk • Kesinlik • F-Ölçütü • Sınıflandırma Hata Oranı
Feature Extraction for Next-Term Prediction of Poor Student Performance / Polyzou ve Karypis (2019)	Makale	• 42 Nitelik • 94.364 Kayıt	• Karar Ağaçları • Yapay Sinir Ağları • Rastgele Orman • Gradyan Artırma	• Python – Sci-Learn	• Doğruluk • Kesinlik • F-ölçütü • ROC Eğrisi
A predictive approach based on efficient feature selection and learning algorithms' competition: Case of learners' dropout in MOOCs / Youssef ve diğerleri (2019)	Makale	• 37 Nitelik • 4.757 Kayıt	• Karar Ağaçları • Naive Bayes • K-En Yakın Komşu • Destek Vektör Makineleri • Rastgele Orman	• Python – Sci-Learn	• Doğruluk • Duyarlılık • F-ölçütü • ROC Eğrisi

Tablo 2.6: Literatür incelemesi (devam).

Çalışma Başlığı / Kaynak	**Çalışma Türü**	**Veri Seti**	**Yöntemler**	**Kullanılan Araçlar**	**Değerlendirme Ölçütleri**
Modelling, prediction and classification of student academic performance using artificial neural networks / Lou ve diğerleri (2019)	Makale	• 11 Nitelik • 1.095 Kayıt	• Yapay Sinir Ağları	• Belirtilmemiş	• Doğruluk • Hassasiyet • F-ölçütü • ROC Eğrisi • Karmaşıklık Matrisi
Presenting the Regression Tree Method and It's application in a Large Scale Educational Dataset / Gomes ve Jelihovschi (2020)	Makale	• 53 Nitelik • 3.670.089 kayıt	• Karar Ağaçları (CART)	• R-Studio • SPSS	• Ortalama Kare Hatası
Toward an improved learning process: the relevance of ethnicity to data mining prediction of students' performance / Adekitan ve Salau (2020)	Makale	• 6 Nitelik • 2.413 Kayıt	• Karar Ağaçları • Yapay Sinir Ağları • Rastgele Orman • Naive Bayes	• Orange	• Doğruluk • Duyarlılık • F-ölçütü • ROC Eğrisi • Karmaşıklık Matrisi
An overview and comparison of supervised data mining techniques for student exam performance prediction / Tomasevic ve diğerleri (2020)	Makale	• 16 Nitelik • 32.593 Kayıt	• K-En Yakın Komşu • Yapay Sinir Ağları • Destek Vektör Makineleri • Bayesian Lineer Regresyon	• MATLAB	• Doğruluk • F-ölçütü • Ortalama Kare Hatası

Tablo 2.6: Literatür incelemesi (devam).

Çalışma Başlığı / Kaynak	**Çalışma Türü**	**Veri Seti**	**Yöntemler**	**Kullanılan Araçlar**	**Değerlendirme Ölçütleri**
Ortaokul Öğrencilerinin Akademik Başarılarının Eğitsel Veri Madenciliği Yöntemleri ile Tahmini / Abbasoğlu (2020)	Makale	• 27 Nitelik • 1395 Kayıt	• Destek Vektör Makineleri • Yapay Sinir Ağları • K-En Yakın Komşu • Rastgele Orman • Lojistik Regresyon • Naive Bayes	• WEKA	• Doğruluk
Predicting Student Performance: an Application of Data Mining Methods With an Educational Web-Based System / Minaei-Bigdoli ve diğerleri (2003)	Bildiri	• 10 Nitelik • 227 Kayıt	• Karar Ağaçları (CART - C5.0 – QUEST - CRUISE) • Yapay Sinir Ağları • Naive Bayes • K-En Yakın Komşu	• MATLAB	• Doğruluk • ROC Eğrisi
Student Academic Performance Monitoring and Evaluation Using Data Mining Techniques / Ogor ve diğerleri (2007)	Bildiri	• 78 Nitelik • 1.369 Kayıt	• Karar Ağaçları (CART - C5.0 - CHAID)	• SPSS	• Doğruluk
Early Prediction of Student Success: Mining Students Enrolment Data / Kovacic (2010)	Bildiri	• 9 Nitelik • 450 Kayıt	• Karar Ağaçları (CART - CHAID)	• SPSS • Statistica	• Karmaşıklık Matrisi • Doğruluk

Tablo 2.6: Literatür incelemesi (devam).

Çalışma Başlığı / Kaynak	**Çalışma Türü**	**Veri Seti**	**Yöntemler**	**Kullanılan Araçlar**	**Değerlendirme Ölçütleri**
Classification and prediction based data mining algorithms to predict slow learners in education sector / Kaur ve diğerleri (2015)	Bildiri	• 13 Nitelik • 152 kayıt	• Karar Ağaçları • Yapay Sinir Ağları • Naive Bayes • Raptree • Sıralı Minimal Optimizasyon	• WEKA	• Doğruluk • Kesinlik • Duyarlılık • F-ölçütü • ROC Eğrisi
Predicting Students Performance in Educational Data Mining / Guo ve diğerleri (2015)	Bildiri	• 14+ Nitelik • 120.000 Kayıt	• Destek Vektör Makineleri • Naive Bayes • Yapay Sinir Ağları • SPNN (Derin Öğrenme Modeli)	• ANSI C • Theano	• Doğruluk • Kesinlik
Prediction of Students Performance using Educational Data Mining / Devasia ve diğerleri (2016)	Bildiri	• 19 Nitelik • 700 kayıt	• Karar Ağaçları • Naive Bayes • Yapay Sinir Ağları	• Belirtilmemiş	• Doğruluk
Prediction of Students' Academic Success Using Data Mining Methods / Uzel ve diğerleri (2018)	Bildiri	• 16 Nitelik • 480 Kayıt	• Karar Ağaçları • Naive Bayes • Yapay Sinir Ağları • Rastgele Orman	• WEKA	• Doğruluk • Kesinlik • Duyarlılık • Karmaşıklık Matrisi • F-Ölçütü
Analysis of Educational Data Mining using Classification / Jalota ve Agrawal (2019)	Bildiri	• 16 Nitelik • 480 Kayıt	• Karar Ağaçları • Yapay Sinir Ağları • Rastgele Orman • Destek Vektör Makineleri	• WEKA	• Doğruluk • Kesinlik • Duyarlılık • F-ölçütü

Tablo 2.6: Literatür incelemesi (devam).

Çalışma Başlığı / Kaynak	**Çalışma Türü**	**Veri Seti**	**Yöntemler**	**Kullanılan Araçlar**	**Değerlendirme Ölçütleri**
Educational Data Mining: Student Performance Prediction in Academic / Salal ve diğerleri (2019)	Bildiri	• 32 Nitelik • 649 kayıt	• Karar Ağaçları • Naive Bayes • Lojistik Regresyon • Rastgele Orman • Raptree • OneR • ZeroR	• WEKA	• Doğruluk •
Veri Madenciliği Ve Anadolu Üniversitesi Uzaktan Eğitim Sisteminde Bir Uygulama / Aydın (2007)	Tez	• 129 Nitelik • 180.554 kayıt	• Karar Ağaçları • Yapay Sinir Ağları • Lojistik Regresyon	• SPSS	• Doğruluk • Karmaşıklık Matrisi
Üniversite Giriş Sınavında Öğrencilerin Başarılarının Veri Madenciliği Yöntemleri ile Tahmin Edilmesi / Göker (2012)	Tez	• 39 Nitelik • 220 Kayıt	• Karar Ağaçları • Yapay Sinir Ağları (RBF) • Naive Bayes	• WEKA	• Doğruluk • Duyarlılık • F-ölçütü • ROC Eğrisi • Karmaşıklık Matrisi
Öğrencilerin Akademik Başarılarının Veri Madenciliği Metotları ile Tahmini / Şengür (2013)	Tez	• 49 Nitelik • 127 Kayıt	• Karar Ağaçları • Yapay Sinir Ağları	• MATLAB	• Ortalama Kare Hatası • Ortalama Mutlak Hata
Eğitimde Veri Madenciliği Ve Öğrenci Akademik Başarı Öngörüsüne İlişkin Bir Uygulama / Özdemir (2016)	Tez	• 61 Nitelik • 1.706 Kayıt	• Karar Ağaçları • Naive Bayes • K-En Yakın Komşu • Destek Vektör Makineleri • Lojistik Regresyon	• R-Studio	• Doğruluk • Hata Oranı • F-ölçütü • ROC Eğrisi

Tablo 2.6: Literatür incelemesi (devam).

Çalışma Başlığı / Kaynak	**Çalışma Türü**	**Veri Seti**	**Yöntemler**	**Kullanılan Araçlar**	**Değerlendirme Ölçütleri**
Veri Madenciliği Yöntemleri Kullanarak Meslek Yüksek Okulu Öğrencilerinin Akademik Başarı Tahmini / Aydemir (2017)	Tez	• 11 Nitelik • 1387 Kayıt	• Karar Ağaçları, • Naive Bayes, • K-En Yakın Komşu • Sıralı Minimum Optimizasyon • Yapay Sinir Ağları	• WEKA	• Doğruluk • Kesinlik • Duyarlılık • Roc-Auc Skoru, • F-Ölçütü, • TPR • FPR
Makine Öğrenmesi Yöntemleri ve Eğitim Verisi Üzerine Bir Uygulama: Uluslararası Matematik ve Fen Eğilimleri Araştırması 2015 Türkiye Örneği / Filiz (2019)	Tez	• 35 Nitelik • 4577 kayıt	• Karar Ağaçları • Naive Bayes • K-En Yakın Komşu • Destek Vektör Makineleri • Rastgele Orman • Yapay Sinir Ağları • Lojistik Regresyon	• WEKA	• Doğruluk • TPR • FPR • Kesinlik • F-ölçütü • ROC eğrisi • Kappa skoru • MCC

Benzer çalışmalarda, hedef nitelik derecesinin 3 ve üzeri olduğu durumlarda maksimum performansı sağlayan modellerin başarım oranları Tablo 2.7‘de verilmektedir.

Tablo 2.7: Benzer çalışmalarda geliştirilen en başarılı modellerin başarım oranları.

Çalışma	Hedef Nitelik Derecesi	En Başarılı Model	Başarım Oranı Yüzdesi (Doğruluk)
Ogor, 2007	3	Karar Ağaçları	97,90
Cortez ve Silva, 2008	5	Naive Bayes	78,50
Şen ve diğerleri, 2012	5	Karar Ağaçları	94,50
Acharya ve Sinha, 2014	6	Karar Ağaçları	66,00
Naser ve diğerleri, 2015	4	Yapay Sinir Ağları	84,60
Guo ve diğerleri, 2015	5	Hibrit Model	77,20
Gök, 2017	5	Lojistik Regresyon	63,81
Asif ve diğerleri, 2017	3	Naive Bayes	83,60
Aydemir, 2017	5	Karar Ağaçları	59,98
Uzel ve diğerleri, 2018	3	Yapay Sinir Ağları	80,60
Youssef ve diğerleri, 2019	3	Rastgele Orman	95,40
Salal ve diğerleri, 2019	5	Karar Ağaçları	76,73
Abbasoğlu, 2020	5	Lojistik Regresyon	64,13

Literatürdeki benzer çalışmalarda geliştirilen modellerde akademik başarı tahminine en fazla etki eden bağımsız değişkenler Tablo 2.8'de belirtilmektedir.

Tablo 2.8: Benzer çalışmalarda tespit edilmiş olan belirleyici nitelikler.

Benzer Çalışmalarda Tespit Edilen Belirleyici Nitelikler	Sayı
Geçmiş Başarı Durumu	16
Anne Eğitim Durumu	6
Baba Eğitim Durumu	6
Gelir Durumu	6
Devamsızlık	5
Motivasyon	5
Kurs Alma Durumu	4
Kardeş Sayısı	3
Cinsiyet	3
Odası Var mı	3
Baba Meslek	3
Aile Ders Desteği	3
Anne-Baba Birlikte mi	2
Anne Meslek	2
Burs Alma Durumu	2

3. MALZEME VE YÖNTEM

Bu bölümde veri analizi ve makine öğrenmesi modellerinin geliştirilmesi süreçleri anlatılmaktadır. Model geliştirme işlemi CRISP-DM süreç modeline ait adımlar izlenerek gerçekleştirilmiştir.

3.1. MAKİNE ÖĞRENMESİ AŞAMALARININ UYGULANMASI

Bu bölümde çalışmada makine öğrenmesi algoritmalarının uygulanması süreci anlatılmaktadır. Tüm modeller Anaconda Navigator editörü ortamında Python programlama dili kullanılarak geliştirilmiştir. Python programlama dili son yıllarda popülerliği ve kullanımı giderek artan bir programlama dilidir (Harrington, 2012). Açık kaynak kodlu olması, kolay yazım şekli ve veri bilimi alanı için geniş kütüphane desteği olması nedeniyle makine öğrenmesi uygulamalarında büyük avantaj sağlamaktadır (Paper, 2018). Çalışmada kullanılan geliştirme ortamı, programlama dili ve kütüphaneler sürüm bilgileri ile birlikte Tablo 3.1'de gösterilmektedir.

Tablo 3.1: Çalışmada kullanılan programlar ve kütüphaneler.

Program/Kütüphane Adı	Sürüm Bilgisi	Kullanım Amacı	Kaynak
Anaconda Navigator	3.6	Geliştirme Ortamı	Anaconda, 2020
Microsoft Excel	2016	Verinin saklanması	Microsoft Corparation, 2018
Python	3.6.5	Programlama Dili	Van Rossum ve Drake, 2009
Numpy	1.14.3	Bilimsel Hesaplamalar için kullanılan kütüphane	Harris ve diğ., 2020
Pandas	0.23.0	Veri seçme ve dönüştürme işlemleri	McKinney., 2010
Scikit-Learn	0.19.1	Makine Öğrenmesi yöntemlerinin uygulanması	Pedregosa ve diğ., 2011
Seaborn	0.8.1	Veri görselleştirme	Waskom ve diğ., 2017
MatPlotLib	2.2.2	Veri görselleştirme	Hunter., 2007

3.1.1. Problemin Tanımlanması

Makine öğrenmesi ve veri madenciliği yöntemlerinin eğitim alanında kullanılması eğitim-öğretim stratejilerinin belirlenmesine katkı sağlayabilmektedir (Bilen ve diğ., 2014). Tahmin edici modellerden elde edilen anlamlı sonuçların eğitim sürecine olumlu katkıları eğitimciler ve karar verici konumundaki devlet kurumları açısından şu şekilde özetlenebilir;

- Eğitim kurumları ve eğitimciler, öğrencilere yönelik tedbirleri (takviye kurs, danışmanlık vb.) önceden alma imkanına sahip olabilirler, öğretim yöntemlerini düzenleyebilirler (Romero ve Ventura, 2007).
- Eğitim politikası belirleyen merkezi kurumlar, makine öğrenmesi modellerinin çıktılarını değerlendirerek strateji geliştirebilir veya mevcut stratejilerini güncelleyebilirler. Örneğin ortaokul seviyesinde akademik başarı tahmini analiz edilerek öğrencilerin bir sonraki kademe olan lise seviyesindeki okullara türlerine göre daha sağlıklı yönlendirilmesi sağlanabilir.

Tez çalışmasına başlamadan önce eğitim alanında yapılmış olan makine öğrenmesi ve veri madenciliği çalışmaları incelenmiştir. Akademik başarı tahminine yönelik çalışma sayısının oldukça fazla olduğu ancak ortaokul seviyesinde akademik başarı tahmini yapılan çalışmaların çok az sayıda olduğu görülmüştür.

Bu kapsamda "Sınıflandırmaya dayalı makine öğrenmesi yöntemleri kullanılarak ortaokul öğrencilerinin akademik başarılarının önceden tahmin edilebilmesi ve başarıya en fazla etki eden faktörlerin tespit edilmesi" tez çalışmasının ana problemi olarak belirlenmiştir.

3.1.2. Veri Setinin İncelenmesi

Çalışmada kullanılan veri seti ortaokul 6., 7. ve 8. Sınıfta öğrenim görmekte olan 328 öğrencinin sosyoekonomik özellikleri ve bir önceki yıla (2018-2019) ait notlarından oluşmaktadır. Veri seti Milli Eğitim Bakanlığı'na (MEB) bağlı İstanbul İl Milli Eğitim Müdürlüğü'nden izin alınarak örneklem olarak belirlenmiş olan eğitim kurumundan temin edilmiştir. Veri seti kurum yöneticileri tarafından ad, soy ad, öğrenci numarası gibi kişisel veriler silinerek Microsoft Excel 2016 dosyası olarak teslim edilmiştir. Öğrenci velilerinden imzalı onam formu alınmıştır.

Çalışmada kullanılan veri setine ait öznitelikler ve açıklamaları Tablo 3.2'de gösterildiği gibidir.

Tablo 3.2: Veri setine ait öznitelikler ve açıklamaları.

No	Nitelik Adı	Nitelik Açıklaması	Nitelik Türü
1	CİNSİYET	Öğrencinin Cinsiyeti	Kategorik
2	ANNE SAĞ	Anne Sağ mı?	Kategorik
3	BABA SAĞ	Baba Sağ mı?	Kategorik
4	A-B BİRLİKTE	Anne-Baba Birlikte mi?	Kategorik
5	ANNE EĞT D	Anne Eğitim Durumu	Kategorik
6	BABA EĞT D	Baba Eğitim Durumu	Kategorik
7	ANNE HSTLK	Annenin Sürekli Hastalığı Var mı?	Kategorik
8	BABA HSTLK	Babanın Sürekli Hastalığı Var mı?	Kategorik
9	ANNE MESL	Annenin Mesleği	Kategorik
10	BABA MESL	Babanın Mesleği	Kategorik
11	KARDEŞ S	Kardeş Sayısı (Kendisi Hariç)	Nümerik
12	GELİR D	Ailenin Gelir Durumu (Çok Kötü-Düşük-Orta-İyi-Çok İyi)	Kategorik
13	KİMİNLE YAŞIYOR?	Öğrencinin Kiminle Yaşadığı (Aile-Anne-Baba-Akraba-Yurt)	Kategorik
14	ODA VAR	Kendine Ait Odası Var mı?	Kategorik
15	EV KİRA MI?	Oturdukları Ev Kira mı Kendilerinin mi?	Kategorik
16	ISINMA	Yaşadıkları Evin Isınma Şekli	Kategorik
17	SÜREKLİ HST	Öğrencinin Sürekli Bir Hastalığı Var mı?	Kategorik
18	DEVAMSIZLIK	Devamsızlık Yaptığı Gün Sayısı	Nümerik
19	TÜRKÇE	1. Dönem Türkçe Dersi Ortalaması	Nümerik (0-100)
20	MATEMATİK	1. Dönem Matematik Dersi Ortalaması	Nümerik (0-100)
21	İLAÇ KUL	Öğrencinin Sürekli Kullandığı İlaç Var mı?	Kategorik
22	AMELİYAT	Öğrenci Ameliyat Geçirdi mi?	Kategorik
23	BURSLU MU ?	Öğrenci Burslu mu?	Kategorik
24	ÖNCEKİ ORT	Geçmiş Yıllar Sene Sonu Not Ortalamalarının Ortalaması	Nümerik (0-100)
25	SON ORT	Sene Sonu Not Ortalaması (Karar Niteliği/Hedef Nitelik)	Nümerik(0-100)

Veri setinde toplam 25 nitelik bulunmaktadır. İlk 24 değişken bağımsız nitelikler, “SON ORT” değişkeni ise tahmin edilecek olan karar niteliğidir (hedef nitelik). Bağımsız niteliklerin 19 tanesi kategorik, 5 tanesi ise nümerik veri türüne sahiptir. Hedef nitelik ise nümerik yapıda olup önişleme adımında ordinal kategorik yapıya dönüştürülecektir.

Çalışmada kullanılan veri setinin Ms Excel programındaki örnek görünümü Şekil 3.1’deki gibidir.

ODA VAR	EV KİRA MI	ISINMA	SÜREKLİ HST	DEVAMSIZLIK	TÜRKÇE	MATEMATİK	İLAÇ KUL	AMELİYAT	BURSLU MU	ÖNCEKİ ORT	SON ORT
hayır	hayır	kalorifer	hayır	1,5	58,44	49,22	hayır	hayır	hayır	81,143333	72,83
evet	evet	kalorifer	hayır	2	79,22	66,55	hayır	hayır	hayır	93,896667	82,77
evet	evet	kalorifer	hayır	5	71,44	47	hayır	hayır	hayır	77,426667	74,65
evet	hayır	kalorifer	hayır	1	85,66	88	hayır	evet	hayır	94,84	93,5
hayır	hayır	kalorifer	hayır	18	64,66	49,44	hayır	hayır	hayır	76,403333	74,29
evet	hayır	kalorifer	hayır	1	71,44	66,88	hayır	hayır	hayır	86,19	74,01
hayır	hayır	soba	hayır				hayır	hayır	hayır	67,703333	70,56
evet	hayır	kalorifer	hayır	10,5	71,88	64,22	hayır	hayır	hayır	80,543333	80,71
evet	hayır	kalorifer	hayır	0	97,66	96,66	hayır	hayır	hayır	94,703333	96,17
evet	evet	kalorifer	hayır				hayır	hayır	hayır	78,823333	60,65
evet	evet	kalorifer	hayır				hayır	hayır	hayır	77,833333	60,19
evet	hayır	kalorifer	evet	3	85,66	87,44	hayır	hayır	hayır	86,013333	90,45
evet	hayır	kalorifer	evet				evet	hayır	hayır	59,79	58,83
evet	evet	kalorifer	hayır				hayır	hayır	hayır	52,18	50,12
evet	hayır	kalorifer	evet				hayır	evet	hayır	69,446667	71,06
evet	hayır	kalorifer	hayır	1,5	57,22	51,77	hayır	hayır	hayır	#SAYI/0!	68,95
evet	hayır	kalorifer	hayır	4	73,55	45	hayır	hayır	hayır	73,036667	71,73
evet	evet	kalorifer	hayır	16	63,33	57,55	hayır	hayır	hayır	79,276667	70,61
evet	hayır	kalorifer	hayır	2	78,88	52,11	hayır	hayır	hayır	83,386667	75,63
evet	hayır	kalorifer	hayır				hayır	evet	hayır	68,506667	63,52
evet	evet	kalorifer	hayır				hayır	evet	hayır	79,3	67,87

Şekil 3.1: Orijinal verinin Ms Excel programındaki örnek görünümü.

Veri setindeki kayıtlar Excel dosyasından veri çerçevesine (*data frame*) aktarılmış olup bu işlem için Pandas kütüphanesinde bulunan “*read_excel*” fonksiyonundan yararlanılmıştır. Pandas kütüphanesindeki “*info*” fonksiyonu kullanılarak veri setinin genel yapısı görüntülenmiştir (Şekil 3.2).

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 328 entries, 0 to 327
Data columns (total 25 columns):
CİNSİYET            328 non-null object
ANNE SAĞ            327 non-null object
BABA SAĞ            326 non-null object
A-B BİRLİKTE        326 non-null object
ANNE EĞT D          314 non-null object
BABA EĞT D          315 non-null object
ANNE HSTLK          318 non-null object
BABA HSTLK          318 non-null object
ANNE MESL           314 non-null object
BABA MESL           312 non-null object
KARDEŞ S            322 non-null float64
GELİR D             318 non-null object
KİMİNLE YAŞIYOR     321 non-null object
ODA VAR             317 non-null object
EV KİRA MI          319 non-null object
ISINMA              320 non-null object
SÜREKLİ HST         322 non-null object
DEVAMSIZLIK         255 non-null float64
TÜRKÇE              255 non-null float64
MATEMATİK           255 non-null float64
İLAÇ KUL            324 non-null object
AMELİYAT            324 non-null object
BURSLU MU           326 non-null object
ÖNCEKİ ORT          327 non-null float64
SON ORT             328 non-null float64
dtypes: float64(6), object(19)
memory usage: 64.1+ KB
```

Şekil 3.2: Veri seti genel yapısı.

Şekil 3.2 incelendiğinde veri setinin 25 sütun ve 328 satırdan oluştuğu, ayrıca tüm sütunlara ait kayıtların sayıları ve sütunların veri türleri görülmektedir**.**

Veri setinde bulunan nümerik değerlere sahip sütunların istatistiksel özeti Şekil 3.3'de gösterilmiştir. Bu işlem için Pandas kütüphanesindeki "*describe*" fonksiyonu kullanılmıştır.

Index	KARDEŞ S	DEVAMSIZLIK	TÜRKÇE	MATEMATİK	ÖNCEKİ ORT	SON ORT
count	322	255	255	255	327	328
mean	1.67391	4.19137	75.7538	72.6188	80.9008	76.0739
std	1.1005	4.12296	12.3647	15.9268	10.7822	12.1141
min	0	0	44.55	37.66	52.18	35.79
25%	1	1	66.44	59.22	73.3475	68.715
50%	2	3	76.66	73.11	81.76	75.695
75%	2	6	85.22	85.995	89.4975	85.4025
max	9	19.5	99.33	100	99.42	98.88

Şekil 3.3: Nümerik değerlere sahip sütunların istatistiksel özeti.

Şekil 3.3'deki özet tabloda "count" ilgili sütuna ait kayıtların sayısını, "mean" ortalama değeri, "std" standart sapmayı, "min" en küçük değeri, "%25" birinci çeyrek değerini, "%50" ikinci çeyrek değerini, "%75" üçüncü çeyrek değerini, "max" ise en yüksek değeri ifade etmektedir.

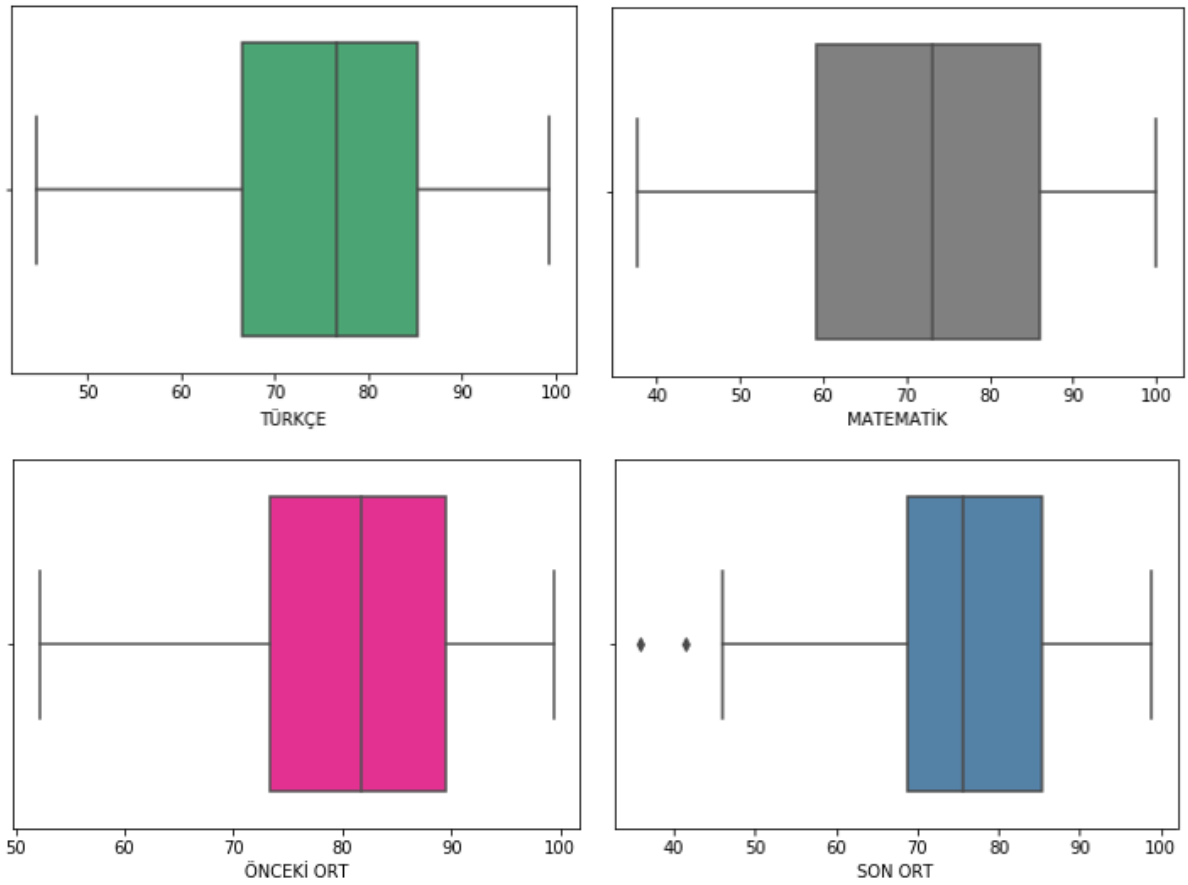
Eksik verinin tespiti için pandas kütüphanesinde bulunan "*isnull*" fonksiyonundan faydalanılmıştır. Eksik değerlerin niteliklere göre dağılımı Şekil 3.4'de gösterilmiştir.

```
Eksik Değerler Listesi:
 CİNSİYET            0
ANNE SAĞ             1
BABA SAĞ             2
A-B BİRLİKTE         2
ANNE EĞT D          14
BABA EĞT D          13
ANNE HSTLK          10
BABA HSTLK          10
ANNE MESL           14
BABA MESL           16
KARDEŞ S             6
GELİR D             10
KİMİNLE YAŞIYOR      7
ODA VAR             11
EV KİRA MI           9
ISINMA               8
SÜREKLİ HST          6
DEVAMSIZLIK         73
TÜRKÇE              73
MATEMATİK           73
İLAÇ KUL             4
AMELİYAT             4
BURSLU MU            2
ÖNCEKİ ORT           1
SON ORT              0
```

Şekil 3.4: Eksik değerler listesi.

Şekil 3.4 incelendiğinde veri setinde toplam 369 adet eksik değer olduğu görülmektedir.

Veri setinde tutarsız ve tekrar eden kayıtlara rastlanmamıştır. Aykırı/uç değerlerin tespiti için Seaborn kütüphanesinde bulunan "boxplot" fonksiyonundan yararlanılmış ve sürekli yapıdaki niteliklerin dağılımları kutu grafikleri ile görüntülenmiştir (Şekil 3.5).



Şekil 3.5: Nümerik niteliklere ait kutu grafikleri.

Kutu grafikleri incelendiğinde "TÜRKÇE", "MATEMATİK" ve "ÖNCEKİ ORT" niteliklerine ait aykırı değere rastlanmamakla birlikte hedef nitelik "SON ORT" değişkenine ait 2 adet aykırı değer tespit edilmiştir. Niteliğe ait dağılımın alt sınırı olan 45.39 değerinin altında kalan 41.34 ve 35.79 değerleri aykırı gözlemler olarak saptanmıştır.

3.1.3. Veri Önişleme

Makine öğrenmesi algoritmalarının veri seti üzerinde uygulanabilmesi için öncelikle veri önişleme işleminin gerçekleştirilmesi gerekmektedir.

Makinenin veriyi sağlıklı ve hızlı bir şekilde yorumlayabilmesi için kategorik verilerin sayısal karşılık değerlerine çevrilmesi gerekmektedir. Bu amaç doğrultusunda Scikit-Learn

kütüphanesinde bulunan "*Label Encoder*" fonksiyonundan yararlanılmış ve kategorik veriler sayısal biçime dönüştürülmüştür. Örneğin Cinsiyet alanı kız-erkek olmak üzere ikili kategorik bir yapıya sahip iken işlem sonucunda 0 ve 1 olmak üzere sayısal biçime çevrilmiştir. Burada "0" değeri cinsiyeti "erkek" olan kayıtları, "1" değeri ise cinsiyeti "kız" olan kayıtları temsil etmektedir.

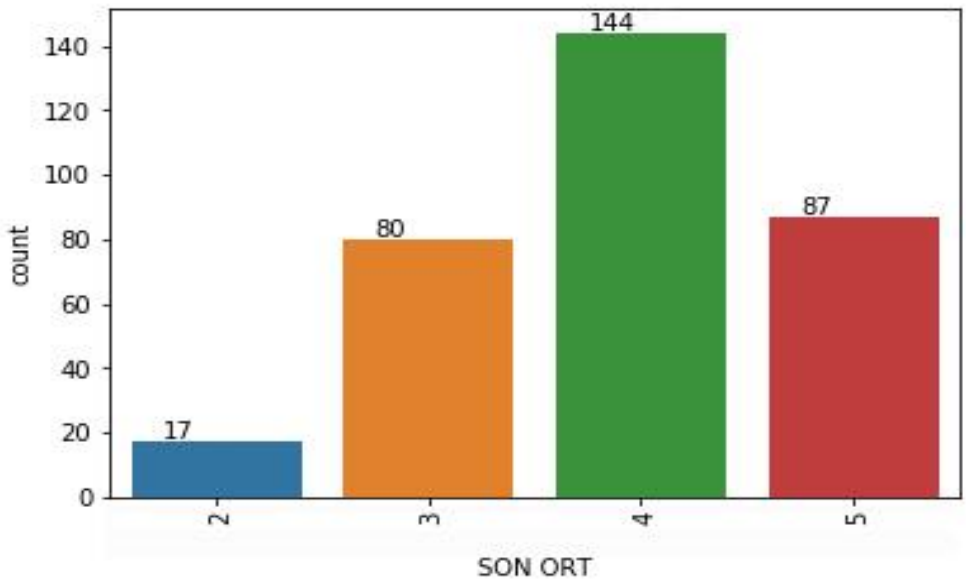
Eksik verinin tamamlanması işlemi değişkenlerin veri türleri ve içerik yapıları dikkate alınarak ayrı ayrı yapılmıştır. Kategorik veri türüne sahip olan sütunlardaki eksik değerler "en sık tekrarlanan değer" yöntemiyle, nümerik veri türüne sahip olan sütunlardaki değerler ise "ortalama değer" yöntemiyle tamamlanmıştır. Bu işlemler için "*SimpleImputer*" fonksiyonu kullanılmış ve "*strategy*" parametresi sırasıyla "*most_frequent*" ve "*mean*" olarak uygulanmıştır.

Hedef niteliğe ait aykırı değerler, kutulama yöntemlerinden baskılama işlemi uygulanarak alt sınır değeri ile değiştirilmiştir. "KARDES S" ve "DEVAMSIZLIK" niteliklerine ait aykırı değerler değiştirilmeden bırakılmıştır. Ayrıklaştırma işlemi kapsamında, veri setinde 100'lük sistemde bulunan bağımlı değişken, sınıflandırma problemi için MEB'in 5'lik not sistemine göre derecelendirilmiştir. 5'lik sistem not çizelgesi Tablo 3.3'de belirtilmektedir.

Tablo 3.3: MEB ortaöğretim kurumları not dönüşüm tablosu (MEB, 2004).

Puan Aralığı	**Not Değeri**
0-24	0
25-44	1
45-54	2
55-69	3
70-84	4
85-100	5

Ayrıklaştırma işlemi uygulandıktan sonra hedef niteliğe ait değerlerin dağılımı Şekil 3.6'daki gibidir. Hedef niteliğin dengesiz bir dağılıma sahip olduğu görülmektedir.



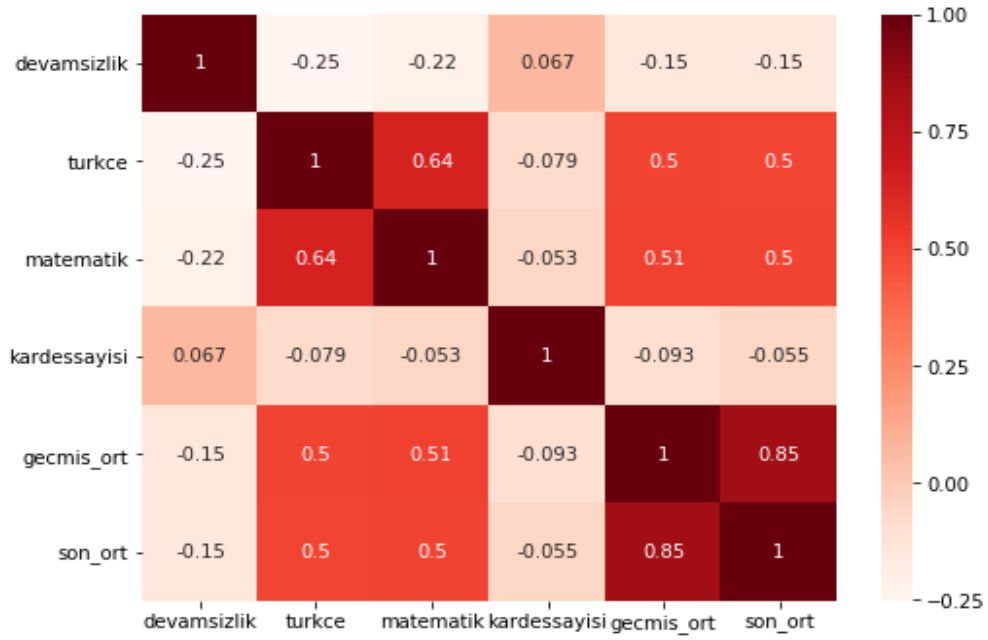
Şekil 3.6: Hedef niteliğe ait değerlerin dağılımı.

Eksik, aykırı değerler ve veri dönüşümü problemleri giderildikten sonra sütunlar ayrı ayrı veri çerçevesine dönüştürülmüş ve birleştirme işlemi yapılarak tek ve yeni bir veri çerçevesi elde edilmiştir. Son durumda veri setine ait nitelik alanları Tablo 3.4'de belirtildiği şekildedir.

Tablo 3.4: Önişleme uygulanmış veri setine ait nitelik alanları.

No	Nitelik Adı	Nitelik Açıklaması	Nitelik Türü
1	cinsiyet	Öğrencinin Cinsiyeti	Kategorik
2	annesag	Anne Sağ mı?	Kategorik
3	babasag	Baba Sağ mı?	Kategorik
4	abbirlikte	Anne-Baba Birlikte mi?	Kategorik
5	anneegt	Anne Eğitim Durumu	Kategorik
6	babaegt	Baba Eğitim Durumu	Kategorik
7	annehastalik	Annenin Sürekli Hastalığı Var mı?	Kategorik
8	babahastalik	Babanın Sürekli Hastalığı Var mı?	Kategorik
9	annemeslek	Annenin Mesleği	Kategorik
10	babameslek	Babanın Mesleği	Kategorik
11	gelir	Ailenin Gelir Durumu (Çok Kötü-Düşük-Orta-İyi-Çok İyi)	Kategorik
12	kiminleyasiyor	Öğrencinin Kiminle Yaşadığı (Aile-Anne-Baba-Akraba-Yurt)	Kategorik
13	odavarmi	Kendine Ait Odası Var mı?	Kategorik
14	evkirami	Oturdukları Ev Kira mı Kendilerinin mi?	Kategorik
15	isinma	Yaşadıkları Evin Isınma Şekli	Kategorik
16	sureklihastalik	Öğrencinin Sürekli Bir Hastalığı Var mı?	Kategorik
17	devamsizlik	Devamsızlık Yaptığı Gün Sayısı	Nümerik
18	turkce	1. Dönem Türkçe Dersi Ortalaması	Nümerik (0-100)
19	matematik	1. Dönem Matematik Dersi Ortalaması	Nümerik (0-100)
20	surekliilac	Öğrencinin Sürekli Kullandığı İlaç Var mı?	Kategorik
21	ameliyat	Öğrenci Ameliyat Geçirdi mi?	Kategorik
22	burslumu	Öğrenci Burslu mu?	Kategorik
23	kardessayisi	Kardeş Sayısı (Kendisi Hariç)	Nümerik
24	gecmis_ort	Geçmiş Yıllar Sene Sonu Not Ortalamalarının Ortalaması	Nümerik (0-100)
25	son_ort	Sene Sonu Not Ortalaması (Karar Niteliği/Hedef Nitelik)	Kategorik(2-3-4-5)

Nümerik değişkenler arasındaki korelasyonun incelenmesi amacıyla Seaborn kütüphanesi bünyesinde bulunan “heatmap” fonksiyonu kullanılarak ısı haritası oluşturulmuştur (Şekil 3.7).



Şekil 3.7: Nümerik değerler arası korelasyon ısı haritası.

Isı haritasında bağımlı değişken ile diğer nümerik değişkenler arasındaki korelasyon incelenmiştir. Buna göre bağımlı değişkenin; geçmiş yılların ortalaması değişkeni ile yüksek dereceli pozitif yönlü korelasyona, Türkçe ve Matematik not ortalamaları ile orta dereceli pozitif yönlü korelasyona, devamsızlık ve kardeş sayısı değişkenleri ile düşük dereceli negatif yönlü korelasyona sahip olduğu görülmektedir.

Veri indirgeme kapsamında veri setine özellik seçimi uygulanarak karar niteliğine en yüksek derecede etki eden bağımsız nitelikler belirlenmiş ve makine öğrenmesi modellerinde bu nitelikler kullanılmıştır. Özellik seçimi için ki-kare yöntemi kullanılmış ve en yüksek skora sahip olan 11 bağımsız nitelik seçilerek model geliştirmede kullanılmıştır. Scikit-Learn kütüphanesindeki “chi2” fonksiyonundan yararlanılarak gerçekleştirilen Ki-kare analizinin sonuçları Şekil 3.8’de gösterilmektedir.

	Specs	Score
18	matematik	375.455011
23	gecmis_ort	337.457983
17	turkce	200.230865
16	devamsizlik	38.159535
11	kiminleyasiyor	29.393945
3	abbirlikte	15.446569
9	babameslek	8.186024
8	annemeslek	8.017521
4	anneegt	6.680928
12	odavarmi	4.428096
22	kardessayisi	3.409855

Şekil 3.8: Ki-kare yöntemi ile özellik seçimi.

Veri, Scikit-Learn kütüphanesinden faydalanılarak eğitim kümesi ve test kümesi olmak üzere ikiye ayrılmıştır. Veriyi bölme yöntemleri olarak belirli oranlarda bölme yöntemi (hold-out) ve k-kat çapraz geçerleme yöntemi kullanılmıştır. Hold-out yöntemi için %67 eğitim-%33 test ve %60 eğitim-%40 test oranları kullanılmış ve örnek seçimi rastgele olarak ayarlanmıştır. k-kat çapraz geçerleme yöntemi ise k=5 ve k=10 alınarak gerçekleştirilmiştir. Çapraz geçerleme için değerlendirme ölçütü olarak tüm katların ortalamaları alınmıştır. Hold-out yöntemi için "train_test_split" fonksiyonu, çapraz geçerleme yöntemi için ise "cross_val_score" fonksiyonları kullanılmıştır.

Geliştirilecek olan modeller için 2 farklı senaryo hazırlanmıştır. 1. Senaryoda veri mevcut durumu ile analiz edilmiş, 2. Senaryoda ise veri normalize edilerek analiz edilmiştir. Her iki senaryo için aynı algoritmalar uygulanmış ve elde edilen model başarımları karşılaştırılmıştır. 2. Senaryo için veri dönüştürme işlemi yapılmış ve tahmin için kullanılacak olan bağımsız değişkenler normalize edilmiştir. Normalizasyon için "StandardScaler" fonksiyonu kullanılarak z-skor normalizasyonu uygulanmıştır.

3.1.4. Algoritma Seçimi ve Algoritmaların Uygulanması

Bu tez çalışması kapsamında tahmin modelleri geliştirmek için Karar Ağaçları Algoritması, Rastgele Orman Algoritması, K-En Yakın Komşu Algoritması, Destek Vektör Makineleri, Naive Bayes Algoritması, Lojistik Regresyon Algoritması ve Yapay Sinir Ağları kullanılmıştır. Çalışmada kullanılan sınıflandırma algoritmalarının uygulanabilmesi için Scikit-Learn kütüphanesinde kullanılan fonksiyonlar Tablo 3.5'de gösterildiği gibidir.

Tablo 3.5: Çalışmada kullanılan algoritmalara ait fonksiyonlar.

Algoritma	Scikit-Learn Fonksiyonu	Kaynak
Karar Ağaçları	DecisionTreeClassifier	Louppe ve diğ., 2020a
Rastgele Orman	RandomForestClassifier	Louppe ve diğ., 2020b
K-En Yakın Komşu	KNeighborsClassifier	Vanderplas ve diğ., 2020
Destek Vektör Makineleri	SVC	Chang ve Lin., 2011
Naive Bayes	BernoulliNB-GaussianNB vb.	Michel ve diğ., 2020
Lojistik Regresyon	LogisticRegression	Varoquaux ve diğ., 2020
Yapay Sinir Ağları	MLPClassifier	Laradji ve diğ., 2020

Makine öğrenmesi modelleri için parametre seçimi modelin başarım oranını belirlemede önemli rol oynamaktadır. Tüm algoritmalar çalışma prensiplerine bağlı olarak kendilerine özgü parametrelere sahiplerdir. Bu nedenle her bir algoritma için en uygun parametrelerin belirlenmesi amaçlanmış ve optimum parametre değerleri model geliştirmede kullanılmıştır.

K-En Yakın Komşu Algoritması için *k* değeri en temel parametredir. k değeri için [1, 10] aralığındaki değerler ayrı ayrı uygulanmıştır. Algoritma için bir diğer önemli parametre ise uzaklık ölçütüdür. Uzaklık ölçütü parametresi için Jaccard, Öklit, Minkowski ve Manhattan uzaklıkları uygulanmıştır.

Karar Ağaçları Algoritması için ağacın bölünme kriteri ("*split_criterion*") ve maksimum derinlik ("*max_depth*") parametreleri belirlenmiştir. Rastgele Orman Algoritması için Karar Ağaçları Algoritması parametrelerine ek olarak ağaç sayısı parametresi (*n_estimators*) belirlenmiştir. Naive Bayes Algoritması için karar niteliğinin (bağımlı değişken) çoklu sınıf yapısına sahip olasından dolayı *multiclass* parametresi değeri "*multinomial*" olarak seçilmiştir. Nümerik değişkenler arasında genel olarak doğrusal bir ilişki olması sebebi ile Destek Vektör Makineleri için çekirdek fonksiyonu "*linear*" olarak belirlenmiştir.

Yapay Sinir Ağları modeli için gizli katman sayısı ve aktivasyon fonksiyonu parametreleri belirlenmiştir.

Modellerin başarım oranlarını ve performanslarını değerlendirmek için Doğruluk, Kesinlik, Duyarlılık, F ölçütü, Roc-Auc skoru metriklerinden ve Karmaşıklık Matrisinden yararlanılmıştır. Veri setindeki karar niteliği dengesiz dağılıma sahip olduğundan dolayı Doğruluk ölçütü tek başına yeterli ve güvenilir bir ölçüt değildir. Bu nedenle diğer değerlendirme ölçütleri ile birlikte kullanılmıştır. Modellere ait performans değerlendirmesi sonuçları BULGULAR bölümünde anlatılmaktadır.

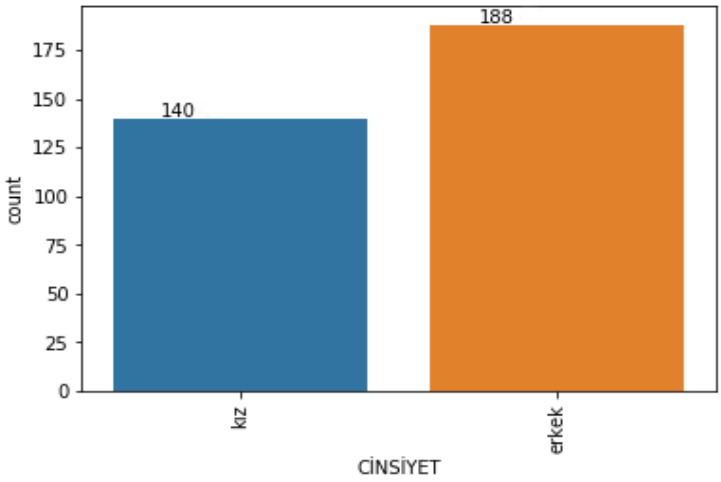
4. BULGULAR

Bu bölümde veri seti üzerinde uygulanan makine öğrenmesi modellerinin karşılaştırmalı sonuçlarına ve modelin uygulamaya alınması aşamasına yer verilmiştir.

4.1. VERİ SETİNE AİT İSTATİSTİKSEL BULGULAR

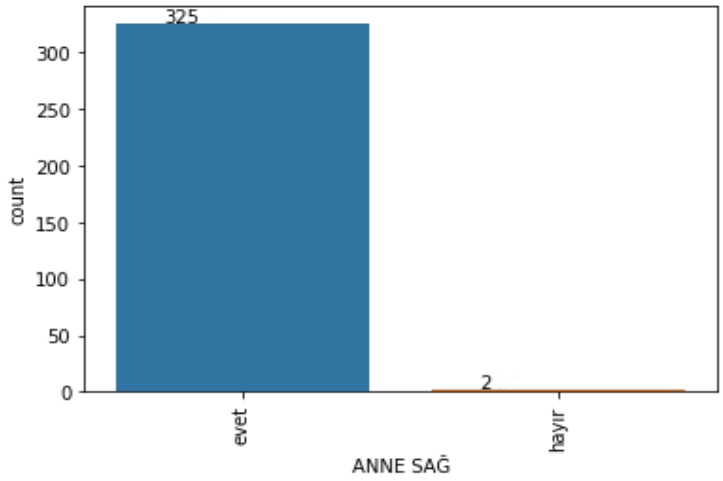
Kategorik niteliklere ait örneklerin sınıf değerlerine göre dağılım grafikleri, Seaborn kütüphanesinde bulunan "countplot" fonksiyonu kullanılarak görüntülenmiştir.

"CİNSİYET" niteliğinin değerlere göre dağılımı incelendiğinde veri setindeki öğrencilerin %42.7'sinin kız, %57.3'ünün erkek olduğu görülmektedir (Şekil 4.1).



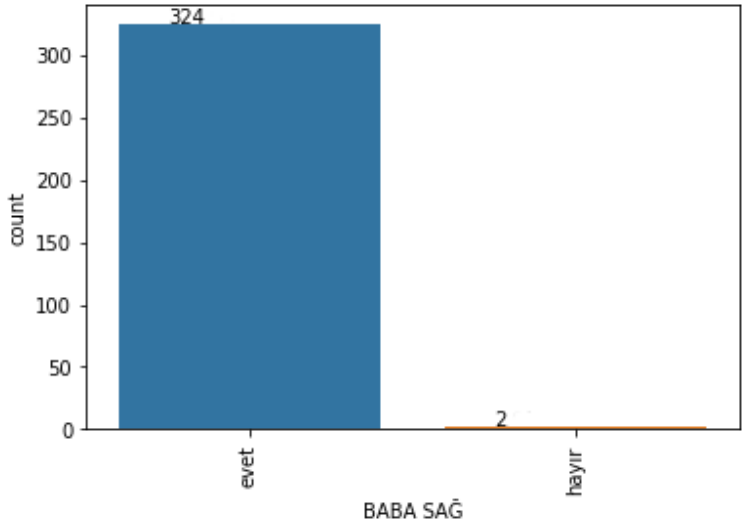
Şekil 4.1: "CİNSİYET" özniteliğinin sınıf değerlerine göre dağılım grafiği.

"ANNE SAĞ" özniteliğinin dağılımı incelendiğinde %99,4'ünün hayatta olduğu, %0,6'sının ise hayatta olmadığı görülmektedir (Şekil 4.2).



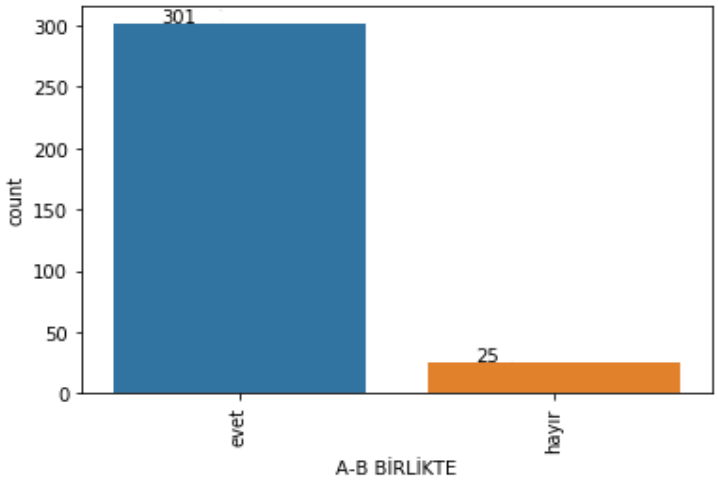
Şekil 4.2: "ANNE SAĞ" özniteliğinin sınıf değerlerine göre dağılım grafiği.

"BABA SAĞ" özniteliğinin dağılımı incelendiğinde %99,4'ünün hayatta olduğu, %0,6'sının ise hayatta olmadığı görülmektedir (Şekil 4.3).



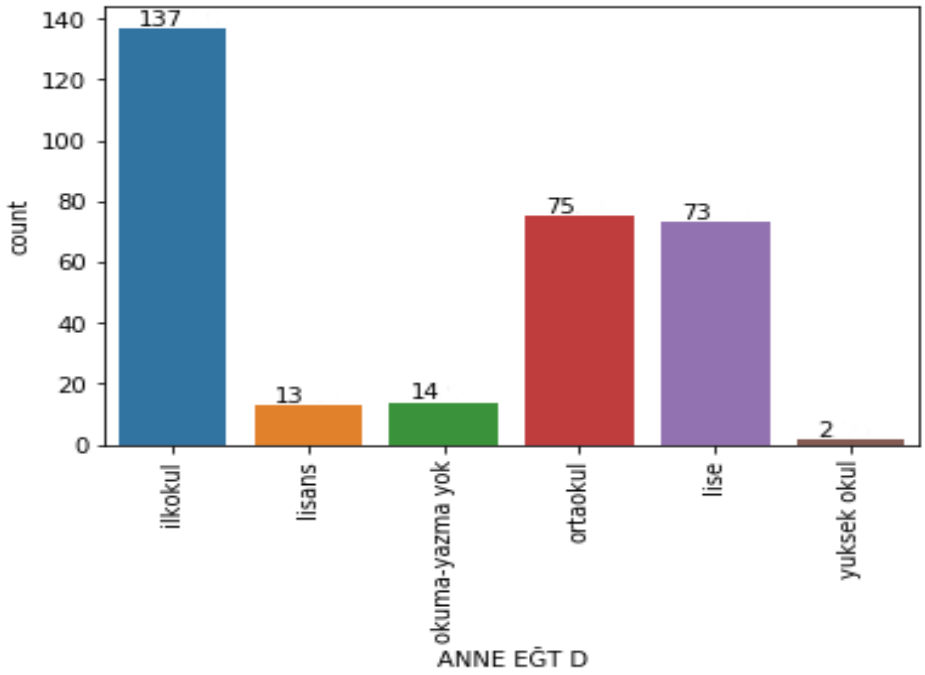
Şekil 4.3: "BABA SAĞ" özniteliğinin sınıf değerlerine göre dağılım grafiği.

"A-B BİRLİKTE" değişkeni incelendiğinde öğrencilerin ebeveynlerinin %92,3'ünün birlikte, %7,7'sinin ise ayrı olduğu görülmektedir (Şekil 4.4).



Şekil 4.4: "A-B BİRLİKTE" özniteliğinin sınıf değerlerine göre dağılım grafiği.

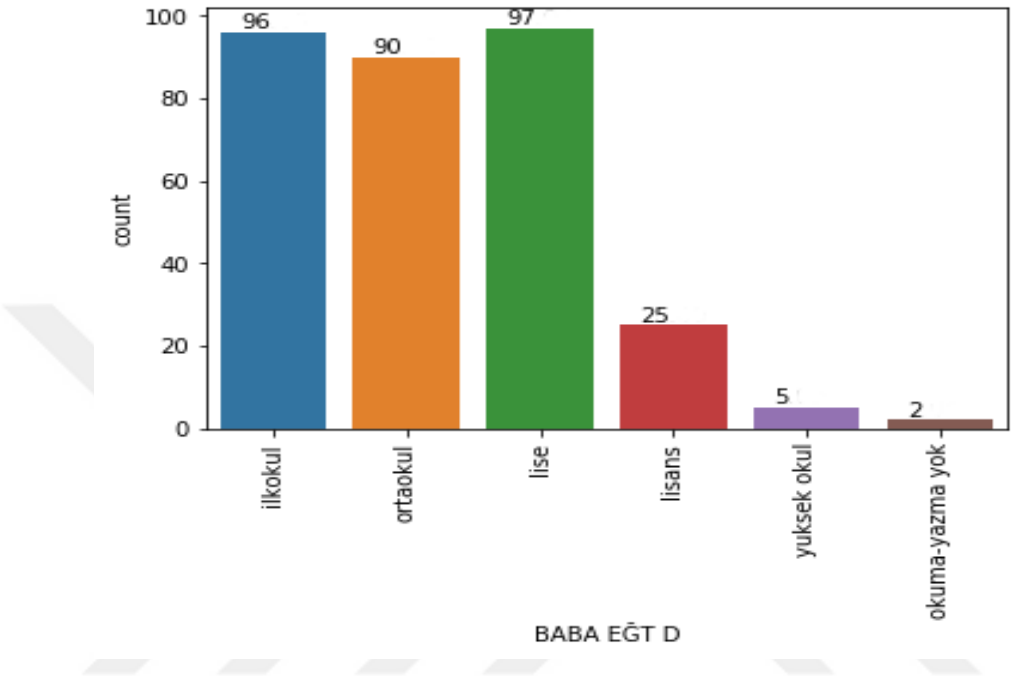
"ANNE EĞT D" niteliğinin sınıf değerlerine göre dağılımı incelendiğinde öğrenci annelerinin büyük bölümünün (%43,6) ilkokul mezunu, %23,9'unun ortaokul mezunu, %23,3'ünün lise mezunu, %4,1'inin lisans mezunu, %0,6'sının yüksekokul mezunu olduğu, %4,5'inin ise okuma yazma bilmediği görülmektedir (Şekil 4.5).



Şekil 4.5: "ANNE EĞT D" özniteliğinin sınıf değerlerine göre dağılım grafiği.

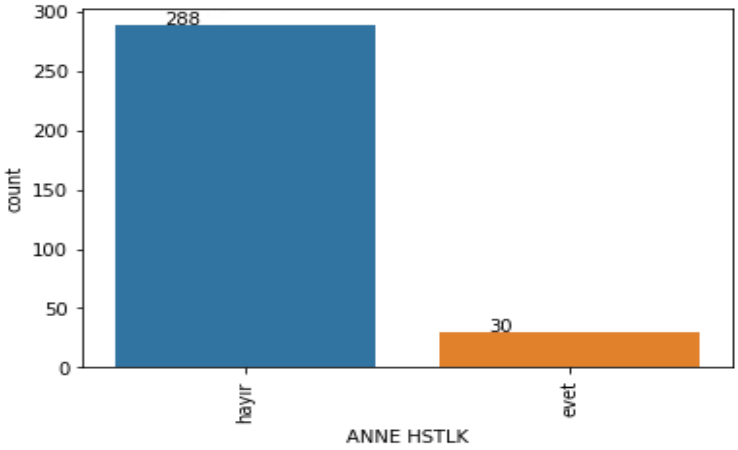
"BABA EĞT D" niteliğinin sınıf değerlerine göre dağılımına bakıldığında öğrenci babalarının %30,8'inin lise mezunu, %30,5'inin ilkokul mezunu, %28,6'sının ortaokul mezunu, %7,9'unun

lisans mezunu, %1,6'sının yüksekokul mezunu olduğu, %0,6'sının ise okuma yazma bilmediği görülmektedir (Şekil 4.6).



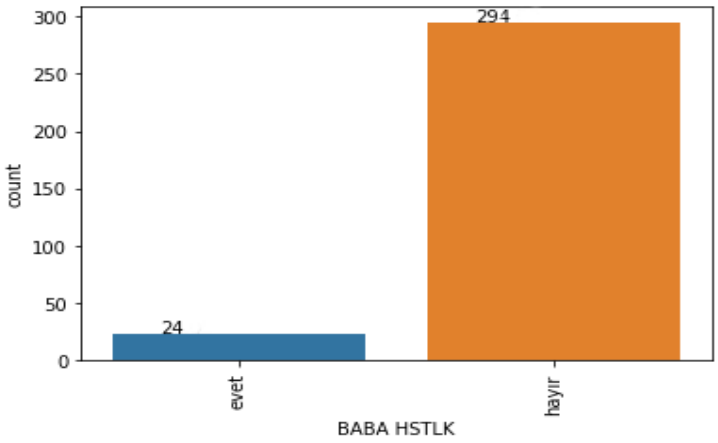
Şekil 4.6: "BABA EĞT D" özniteliğinin sınıf değerlerine göre dağılım grafiği.

"ANNE HSTLK" değişkeninin dağılımı incelendiğinde öğrencilerin annelerinin %90,6'sının herhangi bir sürekli hastalığı olmadığı, %9,4'ünün ise sürekli hastalığı olduğu görülmektedir (Şekil 4.7)



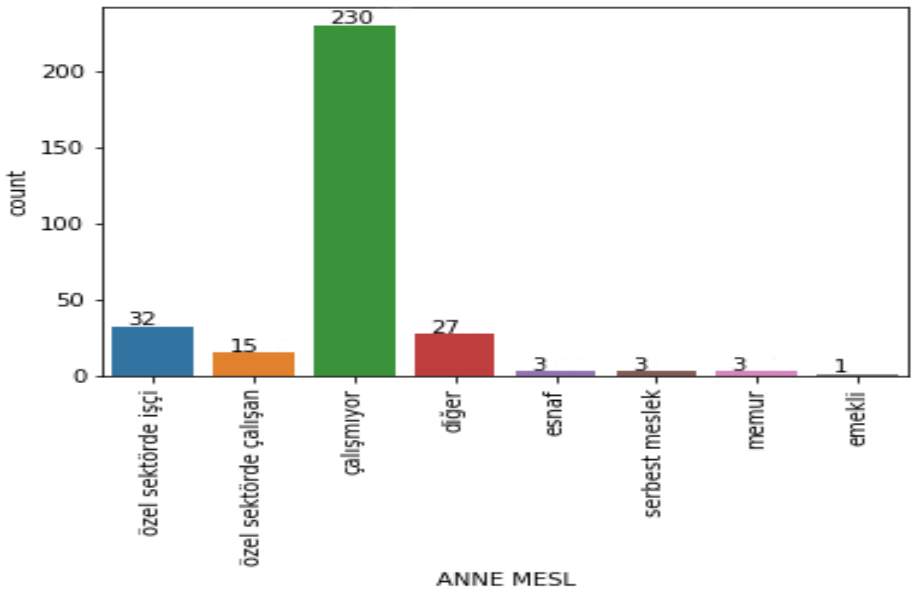
Şekil 4.7: "ANNE HSTLK" özniteliğinin sınıf değerlerine göre dağılım grafiği.

"BABA HSTLK" değişkeninin dağılımına bakıldığında öğrencilerin babalarının %94,3'ünün herhangi bir sürekli hastalığı olmadığı, %5,7'sinin ise sürekli hastalığı olduğu görülmektedir (Şekil 4.8)



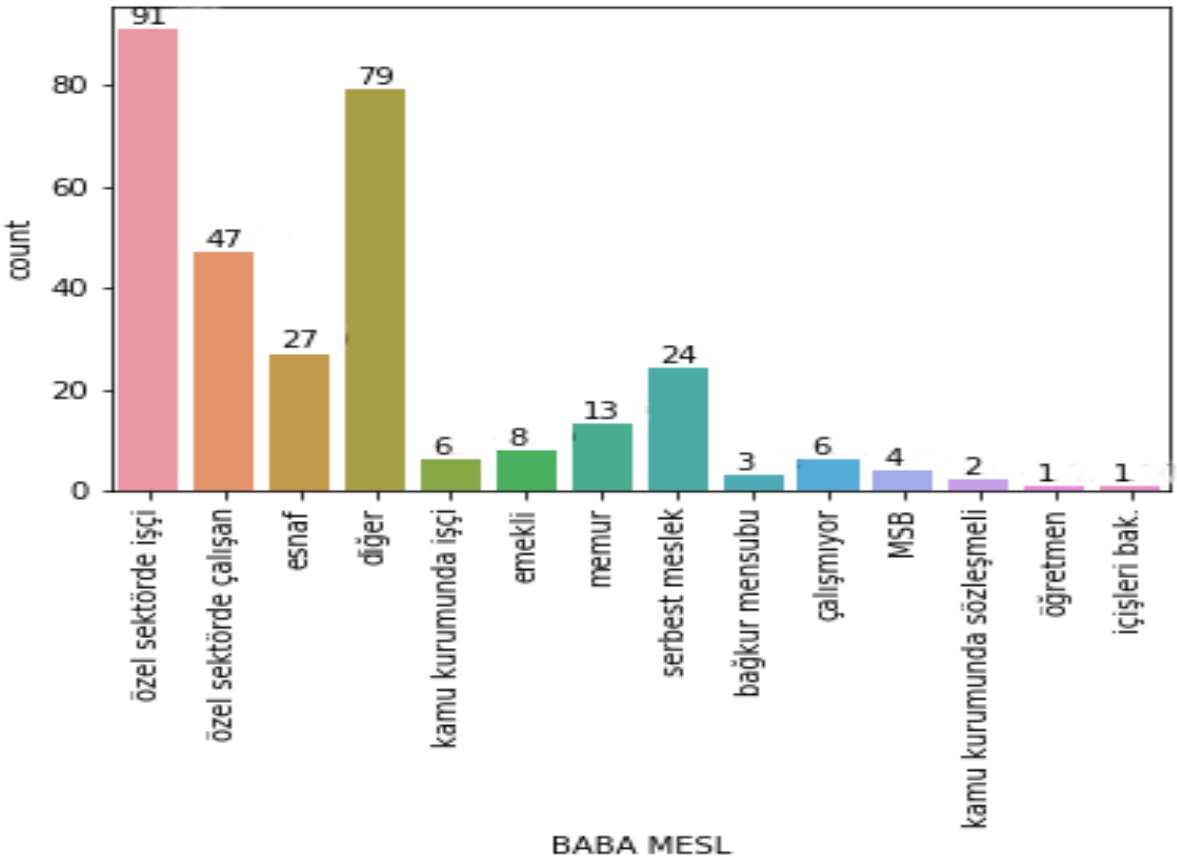
Şekil 4.8: "BABA HSTLK" özniteliğinin sınıf değerlerine göre dağılım grafiği**.**

"ANNE MESL" özniteliğinin sınıf değerlerine göre dağılımı incelendiğinde (Şekil 4.9) öğrenci annelerinin büyük çoğunluğunun herhangi bir işte çalışmadığı (%73,2), %10,2'sinin özel sektörde işçi, %4,8'inin özel sektörde çalışan olduğu, %8,6'sının ise listede olmayan bir işte çalışmakta olduğu (diğer) görülmektedir. Esnaf, serbest meslek, memur ve emekli kategorilerinde yer alanların sayılarının çok az olduğu gözlenmiştir.



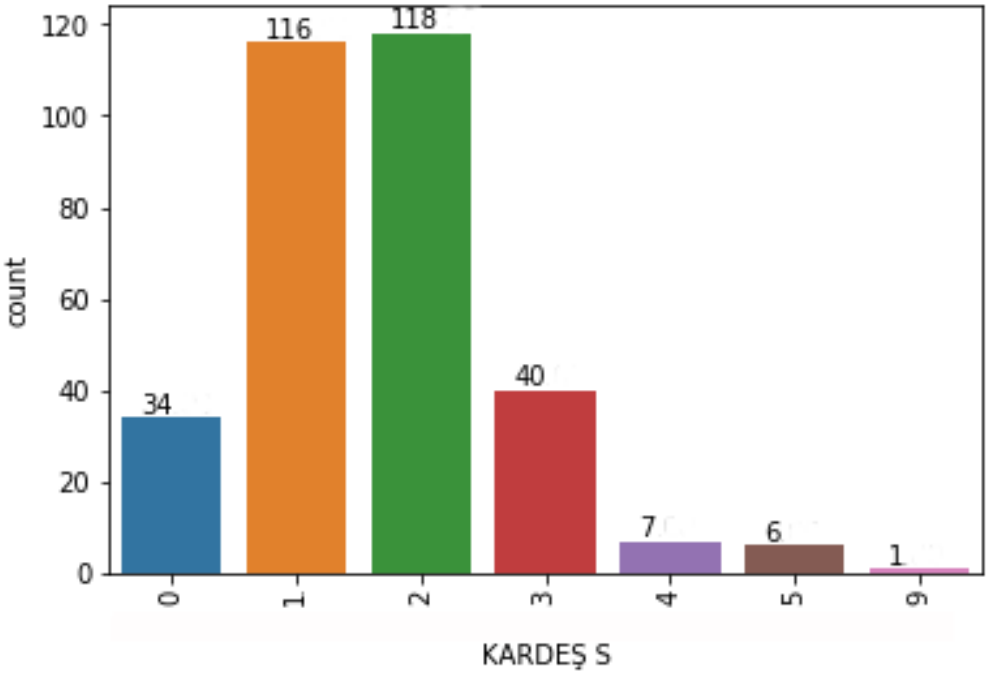
Şekil 4.9: "ANNE MESL" özniteliğinin sınıf değerlerine göre dağılım grafiği.

"BABA MESL" özniteliğinin sınıf değerlerine göre dağılımı incelendiğinde (Şekil 4.10) öğrenci babalarının %29,2'sinin özel sektörde işçi, %15,1'inin özel sektörde çalışan, %8,7'sinin esnaf, %7,7'sinin serbest meslek kategorilerinde olduğu görülmektedir. %25,3'ünün listede olmayan meslek gruplarında çalıştığı, diğer meslek kategorilerinde yer alanların sayılarının az olduğu görülmektedir.



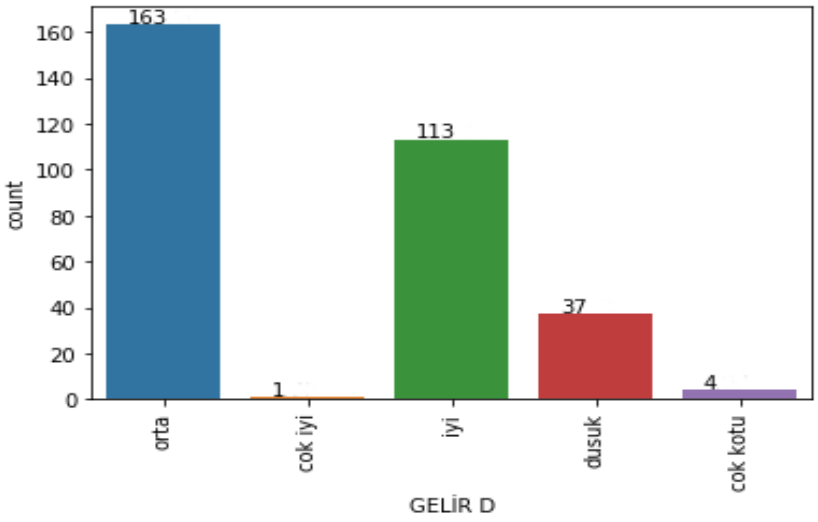
Şekil 4.10: "BABA MESL" özniteliğinin sınıf değerlerine göre dağılım grafiği.

"KARDEŞ S" özniteliğinin sınıf değerlerine göre dağılımına bakıldığında (Şekil 4.11); öğrencilerin %36,6'sının 2 kardeşi, %36'sının 1 kardeşi, %12,4'ünün 3 kardeşi olduğu, %10,6'sının kardeşi olmadığı, 4 ve üzeri kardeşi olanlarının sayılarının az olduğu görülmektedir.



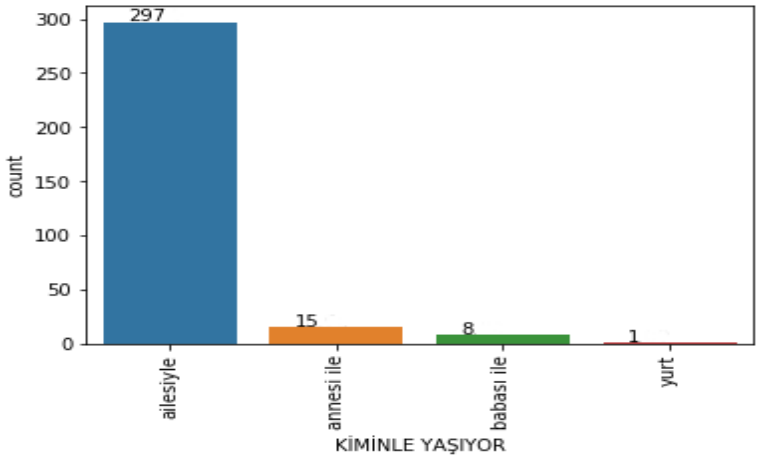
Şekil 4.11: "KARDEŞ S" özniteliğinin sınıf değerlerine göre dağılım grafiği.

"GELİR D" özniteliğinin sınıf değerlerine göre dağılımı incelendiğinde (Şekil 4.12) öğrenci ailelerinin büyük bölümünün gelir durumlarının orta seviyede (%51,3), %35,5'inin iyi seviyede, %11,6' sının düşük seviyede, çok kötü ve çok iyi seviyede olanların az sayıda olduğu görülmektedir.



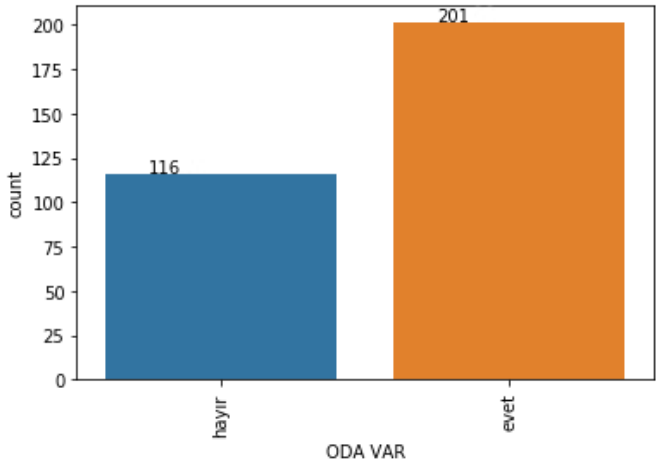
Şekil 4.12: "GELİR D" özniteliğinin sınıf değerlerine göre dağılım grafiği.

"KİMİNLE YAŞIYOR" özniteliğinin sınıf değerlerine göre dağılımı incelendiğinde (Şekil 4.13) öğrencilerin büyük çoğunluğunun ailesi ile (%92,5), %4,7'sinin annesi ile %2,5'inin ise babası ile yaşadığı görülmektedir.



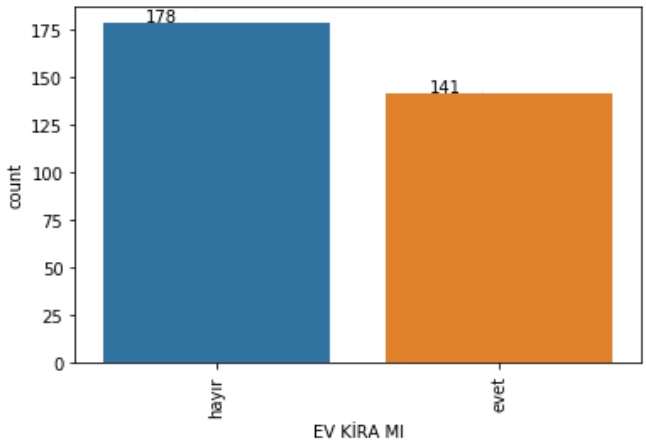
Şekil 4.13: "KİMİNLE YAŞIYOR" özniteliğinin sınıf değerlerine göre dağılım grafiği.

"ODA VAR" niteliğinin dağılımına bakıldığında (Şekil 4.14); öğrencilerin %63,4'ünün kendine ait odasının olduğu %36,6'sının ise kendine ait odasının olmadığı görülmektedir.



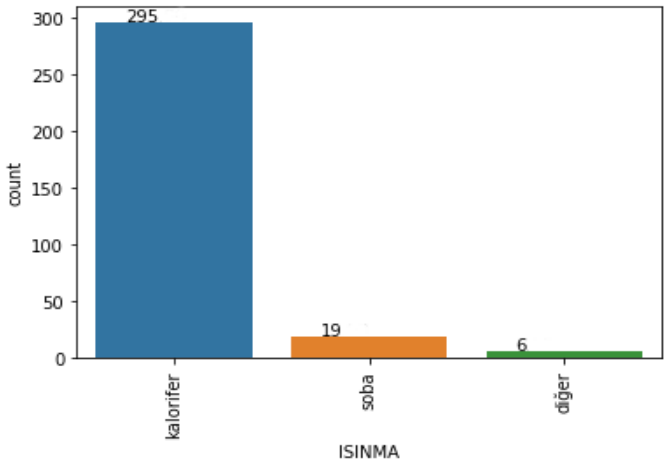
Şekil 4.14: "ODA VAR" özniteliğinin sınıf değerlerine göre dağılım grafiği.

"EV KİRA MI" özniteliğinin dağılımına bakıldığında (Şekil 4.15); öğrencilerin ailelerinin %55,8'inin kendi evinde yaşadığı ve %44,2'sinin kirada olduğu görülmektedir.



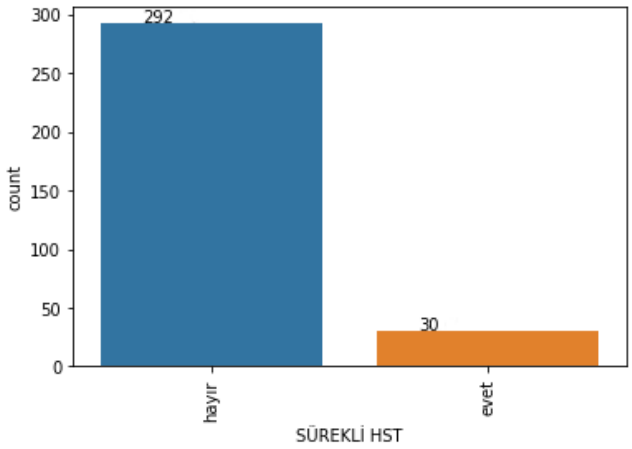
Şekil 4.15: "EV KİRA MI" özniteliğinin sınıf değerlerine göre dağılım grafiği.

"ISINMA" özniteliğinin sınıf değerlerine göre dağılımı incelendiğinde (Şekil 4.16) öğrencilerin yaşadıkları evlerin %92,2'sinin ısınma sisteminin kalorifer (doğalgaz), %5,9'unun soba olduğu görülmektedir.



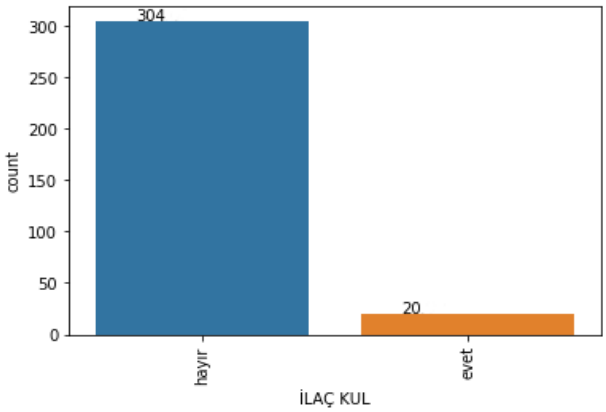
Şekil 4.16: "ISINMA" özniteliğinin sınıf değerlerine göre dağılım grafiği.

"SÜREKLİ HST" özniteliğinin sınıf değerlerine göre dağılımı incelendiğinde (Şekil 4.17) öğrencilerin büyük bölümünün (%90,7) sürekli bir hastalığı olmadığı %9,3'ünün ise sürekli bir hastalığı olduğu görülmektedir.



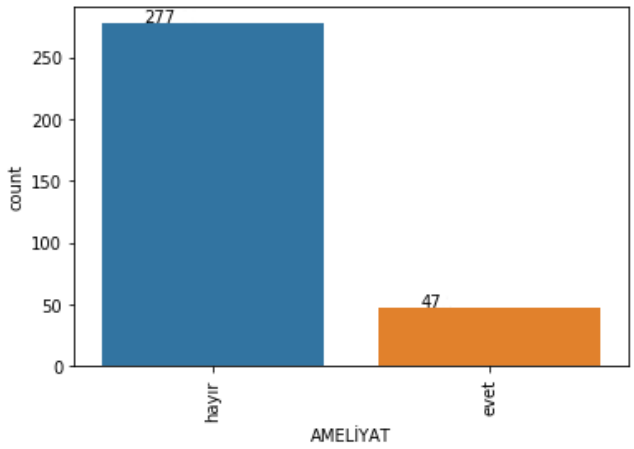
Şekil 4.17: "SÜREKLİ HST" özniteliğinin sınıf değerlerine göre dağılım grafiği**.**

"İLAÇ KUL" özniteliğinin sınıf değerlerine göre dağılımı incelendiğinde (Şekil 4.18) öğrencilerin %93.8'inin sürekli olarak bir ilaç kullanmadığı %6,2'sinin ise sürekli olarak ilaç kullandığı görülmektedir.



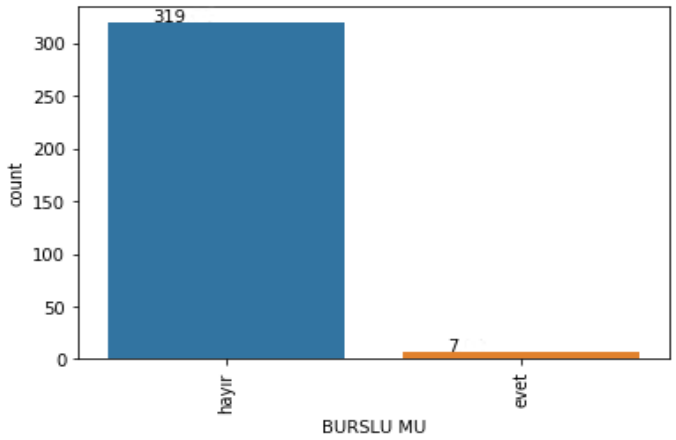
Şekil 4.18: "İLAÇ KUL" özniteliğinin sınıf değerlerine göre dağılım grafiği.

"AMELİYAT" özniteliğinin sınıf değerlerine göre dağılımı incelendiğinde (Şekil 4.19) öğrencilerin %85,5'inin herhangi bir ameliyat geçirmediği %14,5'inin ise ameliyat geçirdiği görülmektedir.



Şekil 4.19: "AMELİYAT" özniteliğinin sınıf değerlerine göre dağılım grafiği.

"BURSLU MU" özniteliğinin sınıf değerlerine göre dağılımı incelendiğinde (Şekil 4.20) öğrencilerin %97,9'unun herhangi bir burs almadığı %2,1'inin ise burs aldığı görülmektedir.



Şekil 4.20: "BURSLU MU" özniteliğinin sınıf değerlerine göre dağılım grafiği.

4.2. NORMALİZE EDİLMEMİŞ (TEMEL) VERİ SETİNE AİT SONUÇLAR

1. Senaryoda veri normalize edilmeden analiz edilmiştir. Temel veri setinin Hold-out %67/%33 oranlarında eğitim ve test kümelerine ayrılması ile kurulan modellere ait parametre seçimleri aşağıda belirtildiği gibidir.

- Rastgele Orman algoritması için ağaç sayısı parametresi (n_estimators) 20 olarak uygulandığında en iyi sonucu vermiştir.

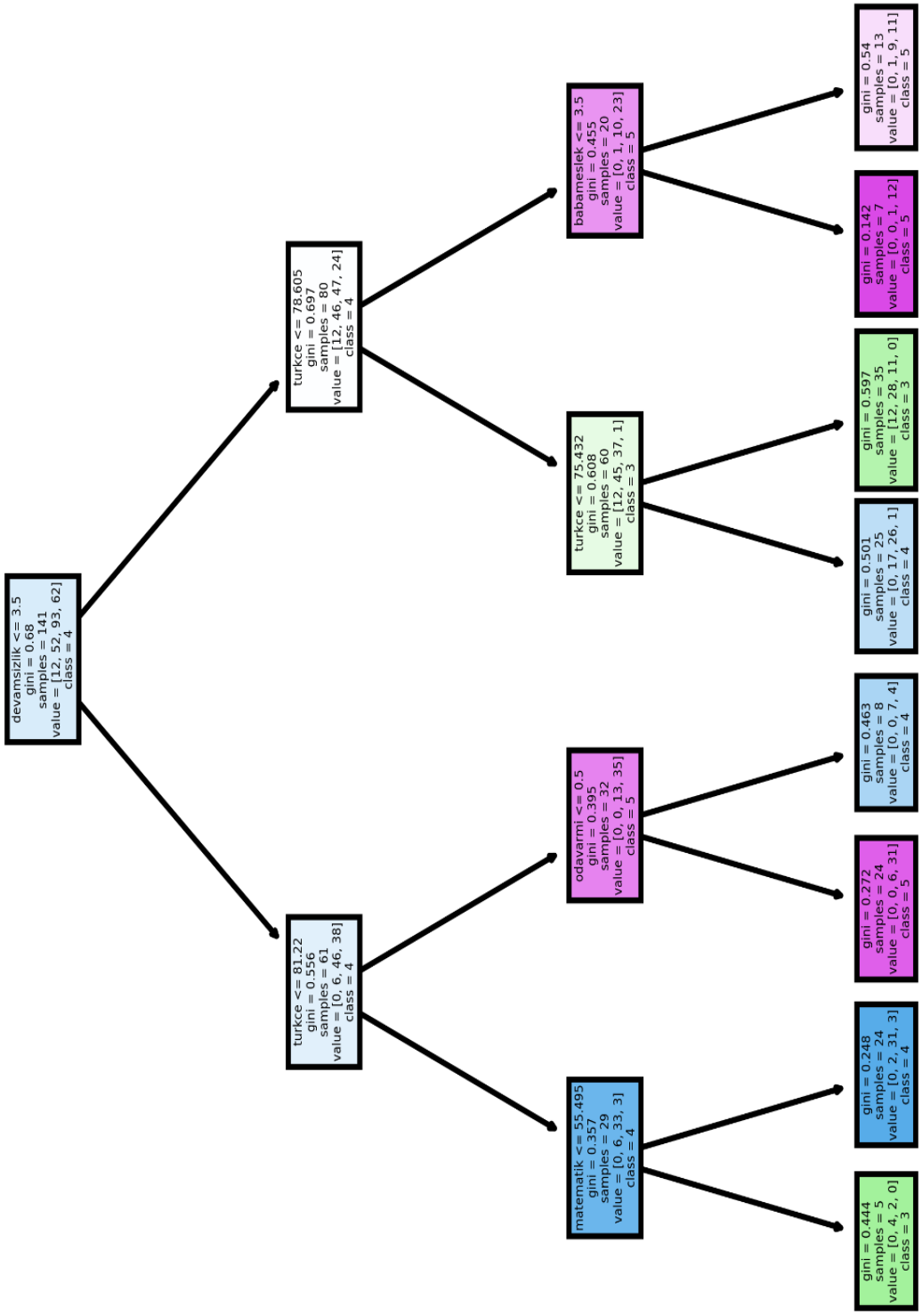
- K-En Yakın Komşu Algoritması için k=7 ve uzaklık ölçütü olarak "Manhattan" uzaklığı kullanılmıştır.
- Karar Ağaçları Algoritması için maksimum derinlik parametresi (max_depth) 3, bölünme kriteri (criterion) ise "gini" olarak uygulanmıştır.
- Yapay Sinir Ağları modeli giriş katmanı, üç adet gizli katman ve çıkış katmanından oluşmaktadır. Giriş katmanındaki nöron sayısı 11, gizli katmanlardaki nöron sayıları ise sırasıyla 11, 22 ve 5 olarak uygulanmıştır. Aktivasyon fonksiyonu olarak Hiperbolik Tanjant (tanh) fonksiyonu kullanılmış olup maksimum iterasyon parametresi 300 olarak ayarlanmıştır.
- Destek Vektör Makineleri için çekirdek parametresi "linear" olarak uygulanmıştır.
- Naive Bayes algoritması için "Multinomial" parametresi uygulanmıştır.
- Lojistik Regresyon algoritması için; multi_class parametresi "multinomial", çözücü fonksiyonu belirlememizi sağlayan solver parametresi "lbfgs", maksimum iterasyon sayısı ise 500 olarak uygulanmıştır.

Uygulanan makine öğrenmesi modellerine ait sonuçlar Tablo 4.1'de gösterilmektedir.

Tablo 4.1: Temel veri seti HO-67/33 modellerin performans değerlendirmesi.

Algoritma	**Doğruluk**	**Kesinlik**	**Duyarlılık**	**F-Ölçütü**	**Auc-Skoru**
Knn	0.7889	0.79	0.79	0.79	**0.8589**
Karar Ağaçları	0.6972	0.70	0.70	0.69	0.7604
Rastgele Orman	**0.8073**	**0.82**	**0.81**	**0.81**	0.8504
DVM	0.6880	0.69	0.69	0.68	0.7880
Logistik Reg.	0.6697	0.66	0.67	0.66	0.7319
YSA	0.6238	0.64	0.62	0.61	0.6849
Naive Bayes	0.4587	0.45	0.46	0.44	0.6187

Tablo 4.1. incelendiğinde en yüksek skorları sağlayan algoritmanın Rastgele Orman algoritması olduğu görülmektedir. Rastgele Orman modeline ait ağaç yapısı çizimi Şekil 4.21'deki gibidir. Net görünmesi açısından maksimum derinlik 3 olacak şekilde budanarak verilmiştir.



Şekil 4.21: Rastgele orman modeline ait ağaç yapısı grafiği.

Modellere ait Karmaşıklık Matrisleri Şekil 4.22'de paylaşılmaktadır.

Knn	Tahmin Edilen Değerler			
	2	3	4	5
Gerçek D.				
2	5	0	0	1
3	2	21	5	0
4	0	3	38	7
5	0	0	5	22

Karar Ağaçları	Tahmin Edilen Değerler			
	2	3	4	5
Gerçek D.				
2	2	4	0	0
3	2	20	6	0
4	0	7	32	9
5	0	0	5	22

Rastgele Orman	Tahmin Edilen Değerler			
	2	3	4	5
Gerçek D.				
2	4	1	1	0
3	4	19	5	0
4	0	2	41	5
5	0	0	3	24

Destek Vektör M.	Tahmin Edilen Değerler			
	2	3	4	5
Gerçek D.				
2	4	1	1	0
3	4	19	5	0
4	0	2	41	5
5	0	0	3	24

YSA	Tahmin Edilen Değerler			
	2	3	4	5
Gerçek D.				
2	1	5	0	0
3	0	12	13	3
4	0	3	35	10
5	0	0	7	20

Lojistik Reg.	Tahmin Edilen Değerler			
	2	3	4	5
Gerçek D.				
2	2	3	0	1
3	2	13	12	1
4	0	5	36	7
5	0	0	5	22

Naive Bayes	Tahmin Edilen Değerler			
	2	3	4	5
Gerçek D.				
2	2	0	1	3
3	3	5	15	5
4	0	9	22	17
5	0	2	4	21

Şekil 4.22: Temel veri seti HO-67/33 modellere ait karmaşıklık matrisleri.

Şekil 4.22'de gösterilmekte olan karmaşıklık matrisleri, test kümesindeki örneklerin ait oldukları karar niteliği sınıflarına dair gerçek ve tahmin değerlerini göstermektedir. Karmaşıklık matrislerinin anlaşılması için örnek olarak Destek Vektör Makinelerine ait karmaşıklık matrisi şu şekilde açıklanabilir. Test kümesinde gerçek değeri "2" olan 6 kayıttan 4 kayıt doğru tahmin edilmiş, 1 kayıt "3" ve 1 kayıt "4" olmak üzere toplam 2 kayıt yanlış tahmin edilmiştir. Benzer şekilde gerçek değeri "3" olan 28 kayıttan 19 kayıt doğru tahmin edilmiş, 4 adet "2" ve 5 adet "4" olmak üzere toplam 9 kayıt yanlış tahmin edilmiştir. Gerçek değeri "4" olan 48 kayıttan 41 kayıt doğru tahmin edilmiş, toplam 7 adet kayıt yanlış tahmin edilmiştir. Gerçek değeri "5" olan 27 kayıttan 24 kayıt doğru tahmin edilmiş, 3 adet kayıt ise yanlış tahmin edilmiştir.

Geliştirilen modellere ait sınıflandırma raporları ile kesinlik, duyarlılık ve F-ölçütü oranları karar niteliğine ait her sınıf değeri için ayrı ayrı görüntülenebilmektedir. Modellere ait sınıflandırma raporları Tablo 4.2'de verilmektedir.

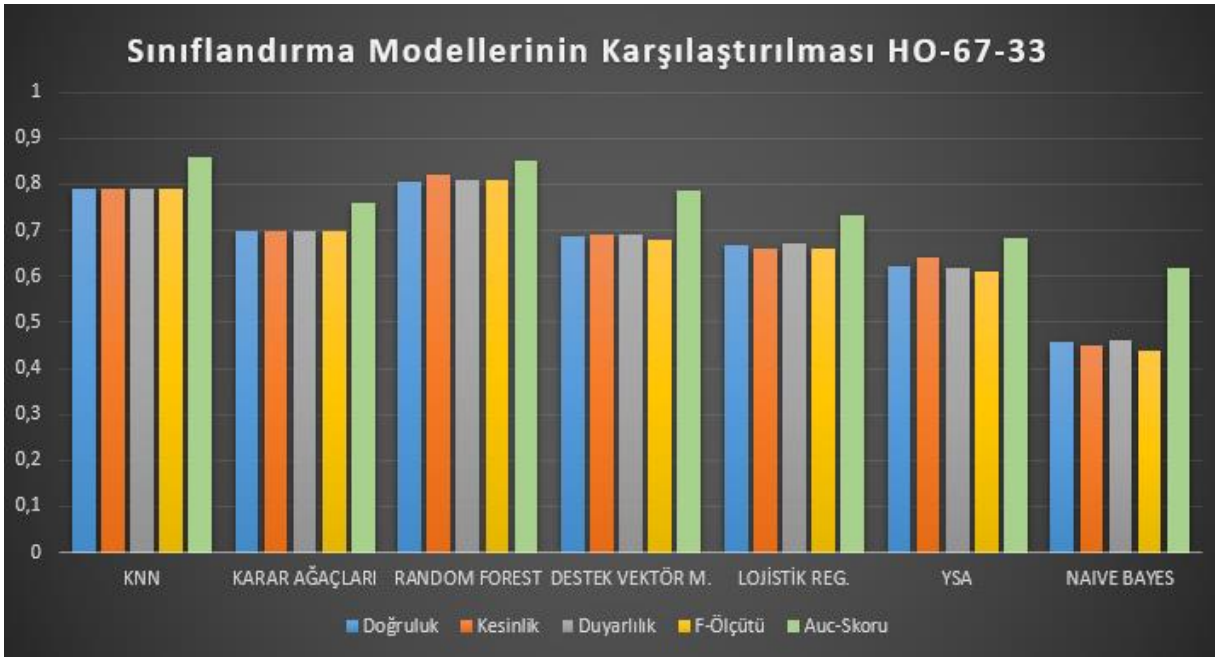
Tablo 4.2: Temel veri seti HO-67/33 modellere ait sınıflandırma raporları.

K-En Yakın Komşu				
Karar Niteliği Sınıfı	Kesinlik	Duyarlılık	F-Ölçütü	Sınıfa ait Örnek Sayısı
2	0.71	0.83	0.77	6
3	0.88	0.75	0.81	28
4	0.79	0.79	0.79	48
5	0.73	0.81	0.77	27
Karar Ağaçları				
Karar Niteliği Sınıfı	Kesinlik	Duyarlılık	F-Ölçütü	Sınıfa ait Örnek Sayısı
2	0.50	0.33	0.40	6
3	0.65	0.71	0.68	28
4	0.74	0.67	0.70	48
5	0.71	0.81	0.76	27
Rastgele Orman				
Karar Niteliği Sınıfı	Kesinlik	Duyarlılık	F-Ölçütü	Sınıfa ait Örnek Sayısı
2	0.50	0.67	0.57	6
3	0.86	0.68	0.76	28
4	0.82	0.85	0.84	48
5	0.83	0.89	0.86	27
Destek Vektör Makineleri				
Karar Niteliği Sınıfı	Kesinlik	Duyarlılık	F-Ölçütü	Sınıfa ait Örnek Sayısı
2	0.44	0.67	0.53	6
3	0.65	0.54	0.59	28
4	0.73	0.67	0.70	48
5	0.73	0.89	0.80	27
Yapay Sinir Ağları				
Karar Niteliği Sınıfı	Kesinlik	Duyarlılık	F-Ölçütü	Sınıfa ait Örnek Sayısı
2	1.00	0.17	0.29	6
3	0.60	0.43	0.50	28
4	0.64	0.73	0.68	48
5	0.61	0.74	0.67	27
Lojistik Regresyon				
Karar Niteliği Sınıfı	Kesinlik	Duyarlılık	F-Ölçütü	Sınıfa ait Örnek Sayısı
2	0.50	0.33	0.40	6
3	0.62	0.46	0.53	28
4	0.68	0.75	0.71	48
5	0.71	0.81	0.76	27
Naive Bayes				
Karar Niteliği Sınıfı	Kesinlik	Duyarlılık	F-Ölçütü	Sınıfa ait Örnek Sayısı
2	0.40	0.33	0.36	6
3	0.31	0.18	0.23	28
4	0.52	0.46	0.49	48
5	0.46	0.78	0.58	27

Tablo 4.2'de yer alan değerler K-En Yakın Komşu algoritması üzerinden şu şekilde açıklanabilir;

Model tarafından hedef niteliğin "3" sınıf değerine ait örneklere yönelik yapılan tahminlerin, %88 kesinlik oranına, %75 duyarlılık oranına ve %81 F-ölçütü oranına sahip olduğu anlaşılmaktadır.

Şekil 4.23'de modellerin başarım oranlarına ait grafik gösterilmektedir.



Şekil 4.23: Temel veri seti HO-67/33 sınıflandırma modellerinin performans karşılaştırma grafiği.

Temel veri seti için Hold-out %67/%33 oranlarında bölme yöntemi ile geliştirilen modellerin sınıflandırma performansları incelendiğinde;

- Rastgele Orman Algoritması %80,73 doğruluk oranı ile en yüksek başarım oranına sahip olmaktadır.
- K-En Yakın Komşu Algoritması %76,14 doğruluk oranı ile Rastgele Orman Algoritmasına en yakın başarım oranına sahip modeldir.
- K-En Yakın Komşu Algoritmasını %69,72 doğruluk oranı ile Karar Ağaçları, %68,80 doğruluk oranı ile Destek Vektör Makineleri, %66,97 doğruluk oranı ile Lojistik Regresyon ve %62,38 doğruluk oranı ile Yapay Sinir Ağları takip etmektedir.

- %45,87 doğruluk oranına sahip olan Naive Bayes algoritması son derece düşük performans göstermişlerdir.
- Kesinlik, Duyarlılık ve F-Ölçütü değerlendirmeleri için sıralamanın değişmediği görülmektedir.
- Auc skorları incelendiğinde, K-En Yakın Komşu algoritması çok küçük bir farkla Rastgele Orman algoritmasından daha yüksek bir değere ulaşmıştır.

Temel veri setinin Hold-out %60/%40 oranlarında eğitim ve test kümelerine ayrılması ile kurulan modellere ait parametre seçimleri aşağıda belirtildiği gibidir.

- Rastgele Orman algoritması için ağaç sayısı parametresi (n_estimators) 10 olarak uygulandığında en iyi sonucu vermiştir.
- K-En Yakın Komşu Algoritması için k=7 ve uzaklık ölçütü olarak "Manhattan" uzaklığı kullanılmıştır.
- Karar Ağaçları Algoritması için maksimum derinlik parametresi (max_depth) 3, bölünme kriteri (criterion) ise "gini" olarak uygulanmıştır.
- Yapay Sinir Ağları modeli giriş katmanı, üç adet gizli katman ve çıkış katmanından oluşmaktadır. Giriş katmanındaki nöron sayısı 11, gizli katmanlardaki nöron sayıları ise sırasıyla 11, 22 ve 5 olarak uygulanmıştır. Aktivasyon fonksiyonu olarak Hiperbolik Tanjant (tanh) fonksiyonu kullanılmış olup maksimum iterasyon parametresi 400 olarak ayarlanmıştır.
- Destek Vektör Makineleri için çekirdek parametresi "linear" olarak uygulanmıştır.
- Naive Bayes algoritması için "Multinomial" parametresi uygulanmıştır.
- Lojistik Regresyon algoritması için; multi_class parametresi "multinomial", çözücü fonksiyonu belirlememizi sağlayan solver parametresi "lbfgs", maksimum iterasyon sayısı ise 500 olarak uygulanmıştır.

Temel veri setinin Hold-out %60/%40 oranlarında bölünmesi ile kurulan modellere ait sonuçlar Tablo 4.3'de gösterilmektedir.

Tablo 4.3: Temel veri seti HO-60/40 modellerin performans değerlendirmesi.

Algoritma	Doğruluk	Kesinlik	Duyarlılık	F-Ölçütü	Roc-Auc
Knn	**0.8030**	**0.81**	**0.80**	**0.80**	**8498**
Karar Ağaçları	0.7575	0.75	0.75	0.75	7737
Rastgele Orman	0.7954	0.79	0.80	0.79	8037
DVM	0.7424	0.74	0.74	0.74	8110
Logistik Reg.	0.6060	0.60	0.61	0.60	7029
YSA	0.5833	0.59	0.58	0.58	6827
Naive Bayes	0.4621	0.46	0.46	0.45	6177

Tablo 4.3. incelendiğinde Hold-out %67/%33 oranlarında bölme yöntemine göre bazı modellerin biraz daha yüksek, bazı modellerin ise biraz daha düşük performans gösterdikleri görülmektedir.

En yüksek skorları sağlayan algoritmanın K-En Yakın Komşu algoritması olduğu görülmektedir. Modellere ait Karmaşıklık Matrisleri Şekil 4.24'de gösterilmektedir.

Knn	Tahmin Edilen Değerler			
	2	3	4	5
Gerçek D.				
2	5	1	1	0
3	1	27	6	0
4	0	3	49	6
5	0	0	8	25

Karar Ağaçları	Tahmin Edilen Değerler			
	2	3	4	5
Gerçek D.				
2	2	5	0	0
3	2	26	6	0
4	0	6	47	5
5	0	0	8	25

Rastgele Orman	Tahmin Edilen Değerler			
	2	3	4	5
Gerçek D.				
2	2	4	1	0
3	2	30	2	0
4	0	5	49	4
5	0	0	9	24

Destek Vektör M.	Tahmin Edilen Değerler			
	2	3	4	5
Gerçek D.				
2	4	2	1	0
3	1	25	8	0
4	0	9	40	9
5	0	0	4	29

YSA	Tahmin Edilen Değerler			
	2	3	4	5
Gerçek D.				
2	2	5	0	0
3	1	15	13	5
4	0	5	35	18
5	0	0	8	25

Lojistik Reg.	Tahmin Edilen Değerler			
	2	3	4	5
Gerçek D.				
2	3	2	1	1
3	1	14	18	1
4	0	9	39	10
5	0	1	8	24

Naive Bayes	Tahmin Edilen Değerler			
	2	3	4	5
Gerçek D.				
2	2	0	1	4
3	2	10	17	5
4	0	13	25	20
5	0	3	6	24

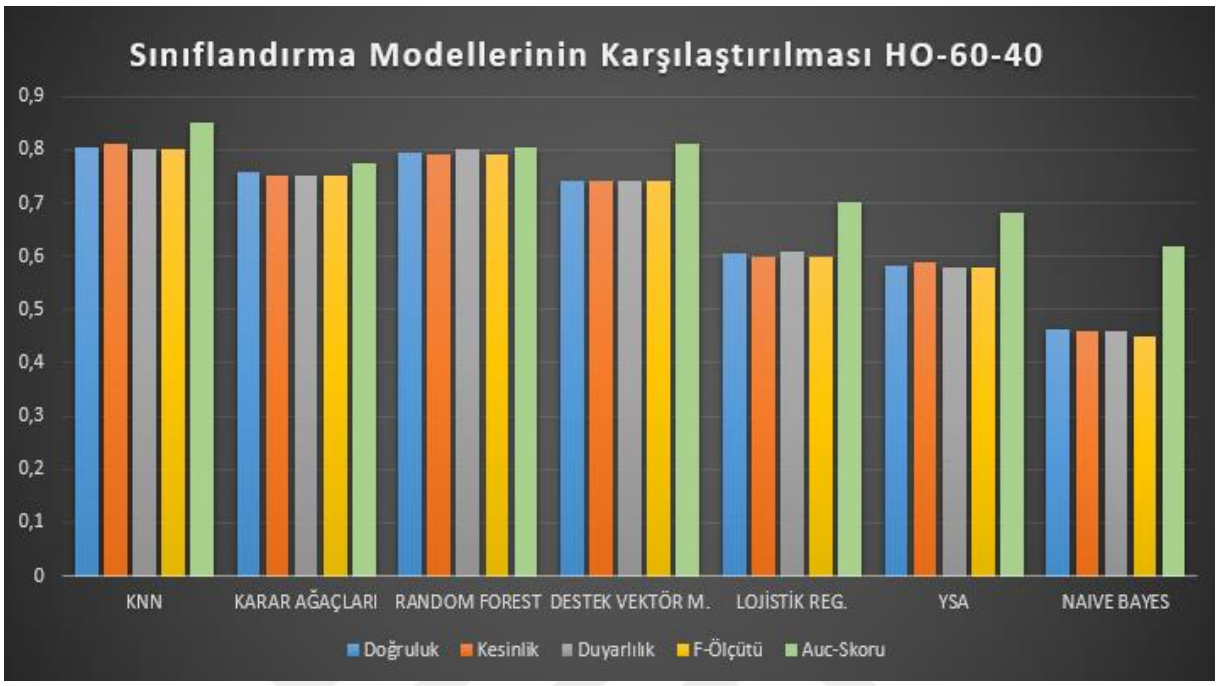
Şekil 4.24: Temel veri seti HO-60/40 modellere ait karmaşıklık matrisleri.

Modellere ait sınıflandırma raporları Tablo 4.4'de gösterilmektedir.

Tablo 4.4: Temel veri seti HO-60/40 modellere ait sınıflandırma raporları.

K-En Yakın Komşu				
Karar Niteliği Sınıfı	Kesinlik	Duyarlılık	F-Ölçütü	Sınıfa ait Örnek Sayısı
2	0.83	0.71	0.77	7
3	0.87	0.79	0.83	34
4	0.77	0.84	0.80	58
5	0.81	0.76	0.78	33
Karar Ağaçları				
Karar Niteliği Sınıfı	Kesinlik	Duyarlılık	F-Ölçütü	Sınıfa ait Örnek Sayısı
2	0.50	0.29	0.36	7
3	0.70	0.76	0.73	34
4	0.75	0.83	0.79	58
5	0.85	0.70	0.77	33
Rastgele Orman				
Karar Niteliği Sınıfı	Kesinlik	Duyarlılık	F-Ölçütü	Sınıfa ait Örnek Sayısı
2	50	29	36	7
3	77	88	82	34
4	80	84	82	58
5	86	73	79	33
Destek Vektör Makineleri				
Karar Niteliği Sınıfı	Kesinlik	Duyarlılık	F-Ölçütü	Sınıfa ait Örnek Sayısı
2	0.80	0.57	0.67	7
3	0.69	0.74	0.71	34
4	0.75	0.69	0.72	58
5	0.76	0.88	0.82	33
Yapay Sinir Ağları				
Karar Niteliği Sınıfı	Kesinlik	Duyarlılık	F-Ölçütü	Sınıfa ait Örnek Sayısı
2	0.67	0.29	0.40	7
3	0.60	0.44	0.51	34
4	0.62	0.60	0.61	58
5	0.52	0.76	0.62	33
Lojistik Regresyon				
Karar Niteliği Sınıfı	Kesinlik	Duyarlılık	F-Ölçütü	Sınıfa ait Örnek Sayısı
2	0.75	0.43	0.55	7
3	0.54	0.41	0.47	34
4	0.59	0.67	0.63	58
5	0.67	0.73	0.70	33
Naive Bayes				
Karar Niteliği Sınıfı	Kesinlik	Duyarlılık	F-Ölçütü	Sınıfa ait Örnek Sayısı
2	0.50	0.29	0.36	7
3	0.38	0.29	0.33	34
4	0.51	0.43	0.47	58
5	0.45	0.73	0.56	33

Şekil 4.25'de modellerin başarım oranlarına ait grafik gösterilmektedir.



Şekil 4.25: Temel veri seti HO-60/40 sınıflandırma modellerinin performans karşılaştırma grafiği.

Normalize edilmemiş veri seti için Hold-out modellerin sınıflandırma performansları incelendiğinde;

- En yüksek başarım oranlarının, verinin %67 eğitim, %33 test kümesi olarak ayrıldığı durumda Rastgele Orman modeli ile elde edildiği gözlemlenmiştir.
- Her iki bölme durumunda da Rastgele Orman ve K-En Yakın Komşu Algoritmaları birbirlerine oldukça yakın yüksek başarım oranlarına sahip olmaktadır.
- Destek Vektör Makineleri ve Karar Ağaçları algoritmaları uygulanan modeller içerisinde ortalama başarım oranlarına sahiplerdir.
- Yapay sinir Ağları, Lojistik Regresyon ve Naive Bayes algoritmalarının Hold-out yöntemi için son derece düşük performans gösterdikleri görülmüştür.

Veri setinin çapraz geçerleme yöntemi ile 5-kat olarak bölünmesi ile elde edilen her bir kat için doğruluk oranı ve tüm katların ortalama doğruluk oranları Tablo 4.5'de gösterilmektedir.

Tablo 4.5: Temel veri seti 5-kat çapraz geçerleme doğruluk oranları.

Algoritma	k-1	k-2	k-3	k-4	k-5	k-ort
Knn	**0.8181**	0.7500	0.6590	**0.8181**	0.6976	**0.7486**
Karar A.	0.6818	0.6818	0.6590	0.6818	0.5813	0.6571
R. Orman	0.7272	**0.8409**	**0.7045**	0.7045	**0.7441**	0.7442
DVM	0.7727	0.7500	**0.7045**	0.6136	0.6279	0.6937
Loj. Reg.	0.5000	0.6818	0.6818	0.5681	0.5348	0.5933
YSA	0.4772	0.6363	0.5909	0.4545	0.5116	0.5341
Nb	0.5000	0.5227	0.5909	0.5454	0.3953	0.5108

Tablo 4.5. incelendiğinde 5-kat çapraz geçerleme ortalama doğruluk oranı en yüksek modelin K-En Yakın Komşu Algoritmasına ait olduğu görülmektedir. Onu sırasıyla Rastgele Orman, Destek Vektör Makineleri, Karar Ağaçları, Lojistik Regresyon, Yapay Sinir Ağları ve Naive Bayes modelleri izlemektedir.

Veri setinin çapraz geçerleme yöntemi ile 10-kat olarak bölünmesi ile elde edilen her bir kat için doğruluk oranı ve tüm katların ortalama doğruluk oranları Tablo 4.6'da gösterilmektedir.

Tablo 4.6: Temel veri seti 10-kat çapraz geçerleme doğruluk oranları.

Algoritma	k-1	k-2	k-3	k-4	k-5	k-6	k-7	k-8	k-9	k-10	k-ort
Knn	**0.8636**	0.7272	0.6818	0.8181	0.7272	0.5454	**0.8181**	0.7272	0.6363	**0.7142**	0.7305
Karar A.	0.6363	0.7727	**0.7272**	0.7727	0.7272	0.5454	0.6818	**0.8181**	0.5909	0.5238	0.6796
R. Orman	0.8181	**0.8181**	**0.7272**	**0.8636**	0.7727	**0.6818**	0.7272	0.7727	**0.7272**	0.6666	**0.7575**
DVM	0.8181	0.7727	**0.7272**	0.8181	**0.8181**	**0.6818**	0.5909	0.7727	0.6363	0.5714	0.7207
Loj. Reg.	0.4545	0.6818	0.7272	0.6818	0.6363	0.6363	0.5454	0.7727	0.5454	0.5714	0.6253
YSA	0.4090	0.5000	0.6818	0.5909	0.6818	0.5000	0.5454	0.4545	0.5909	0.4761	0.5430
Nb	0.4090	0.3636	0.5454	0.5000	0.6363	0.5454	0.5000	0.5909	0.5454	0.3333	0.4969

Tablo 4.6. incelendiğinde 10-kat çapraz geçerleme ortalama doğruluk oranı en yüksek modelin Rastgele Orman Algoritmasına ait olduğu görülmektedir. Onu sırasıyla K-En Yakın Komşu, Destek Vektör Makineleri, Karar Ağaçları, Lojistik Regresyon, Yapay Sinir Ağları ve Naive Bayes modelleri izlemektedir.

Veri setinin çapraz geçerleme yöntemi kullanılarak eğitim ve test kümelerine bölünmesi ile uygulanan modellere bakıldığında;

- Modellerin genelinde doğruluk oranları ortalamaları, Hold-out yöntemi ile kıyaslandığında biraz daha düşüktür.
- En yüksek başarım oranı Rastgele Orman Algoritması ile elde edilmiştir.
- Modeller arasındaki performans farkının Hold-out yöntemine kıyasla düştüğü gözlemlenmiştir.

4.3. NORMALİZE VERİ SETİNE AİT SONUÇLAR

2. Senaryoda veri normalize edilerek analiz edilmiştir. Normalize edilen veri setinin Hold-out %67/%33 oranlarında eğitim ve test kümelerine ayrılması ile kurulan modellere ait parametre seçimleri aşağıda belirtildiği gibidir.

- Rastgele Orman algoritması için ağaç sayısı parametresi (n_estimators) 20 olarak uygulanmıştır.
- K-En Yakın Komşu Algoritması için k=9 ve uzaklık ölçütü olarak “Manhattan” uzaklığı kullanılmıştır.
- Karar Ağaçları Algoritması için maksimum derinlik parametresi (max_depth) 3, bölünme kriteri (criterion) ise “gini” olarak uygulanmıştır.
- Yapay Sinir Ağları modeli giriş katmanı, üç adet gizli katman ve çıkış katmanından oluşmaktadır. Giriş katmanındaki nöron sayısı 11, gizli katmanlardaki nöron sayıları ise sırasıyla 11, 22 ve 6 olarak uygulanmıştır. Aktivasyon fonksiyonu olarak Hiperbolik Tanjant (tanh) fonksiyonu kullanılmış olup maksimum iterasyon parametresi 300 olarak ayarlanmıştır.
- Destek Vektör Makineleri için çekirdek parametresi “linear” olarak uygulanmıştır.

- Lojistik Regresyon algoritması için; multi_class parametresi "multinomial", çözücü fonksiyonu belirlememizi sağlayan solver parametresi "lbfgs", maksimum iterasyon sayısı ise 500 olarak uygulanmıştır.
- Naive Bayes algoritması için "Bernoulli" parametresi uygulanmıştır.

Veri setinin Hold-out %67/%33 oranlarında eğitim ve test kümelerine ayrılması ile kurulan modellere ait sonuçlar Tablo 4.7'de gösterilmektedir.

Tablo 4.7: Normalize Veri seti için hold-out %67/%33 modellerin performans değerlendirmesi.

Algoritma	**Doğruluk**	**Kesinlik**	**Duyarlılık**	**F-Ölçütü**	**Auc-Skoru**
Knn	0.7339	0.74	0.73	0.73	7932
Karar Ağaçları	0.6788	0.68	0.68	0.68	7496
Rastgele Orman	**0.7706**	**0.79**	**0.77**	**0.77**	**8213**
DVM	0.7247	0.73	0.72	0.72	8035
Logistik Reg.	0.7064	0.70	0.71	0.70	7586
YSA	0.7431	0.74	0.74	0.74	7872
Naive Bayes	0.5779	0.54	0.58	0.51	6251

Tablo 4.7. incelendiğinde en yüksek başarı sağlayan algoritmanın Rastgele Orman algoritması olduğu görülmektedir. Yapay Sinir Ağları, K-En Yakın Komşu, Destek Vektör Makineleri ve Lojistik Regresyon algoritmaları birbirlerine oldukça yakın düzeyde başarım oranlarına sahiplerdir. Naive Bayes algoritması diğer modellerle kıyaslandığında oldukça düşük performans göstermiştir. Modellere ait Karmaşıklık Matrisleri Şekil 4.26'da gösterilmektedir.

Knn	Tahmin Edilen Değerler			
	2	3	4	5
Gerçek D.				
2	3	2	1	0
3	0	19	9	0
4	0	4	35	9
5	0	0	4	23

Karar Ağaçları	Tahmin Edilen Değerler			
	2	3	4	5
Gerçek D.				
2	2	4	0	0
3	2	20	6	0
4	0	8	31	9
5	0	0	6	21

Rastgele Orman	Tahmin Edilen Değerler			
	2	3	4	5
Gerçek D.				
2	4	0	2	0
3	4	15	9	0
4	0	2	42	4
5	0	0	4	23

Destek Vektör M.	Tahmin Edilen Değerler			
	2	3	4	5
Gerçek D.				
2	4	1	1	0
3	4	16	8	0
4	0	6	36	6
5	0	0	4	23

YSA	Tahmin Edilen Değerler			
	2	3	4	5
Gerçek D.				
2	2	3	1	0
3	1	21	6	0
4	0	7	34	7
5	0	0	3	24

Lojistik Reg.	Tahmin Edilen Değerler			
	2	3	4	5
Gerçek D.				
2	2	3	1	0
3	1	17	10	0
4	0	6	35	7
5	0	0	4	23

Naive Bayes	Tahmin Edilen Değerler			
	2	3	4	5
Gerçek D.				
2	0	0	6	0
3	0	3	25	0
4	0	3	38	7
5	0	0	5	22

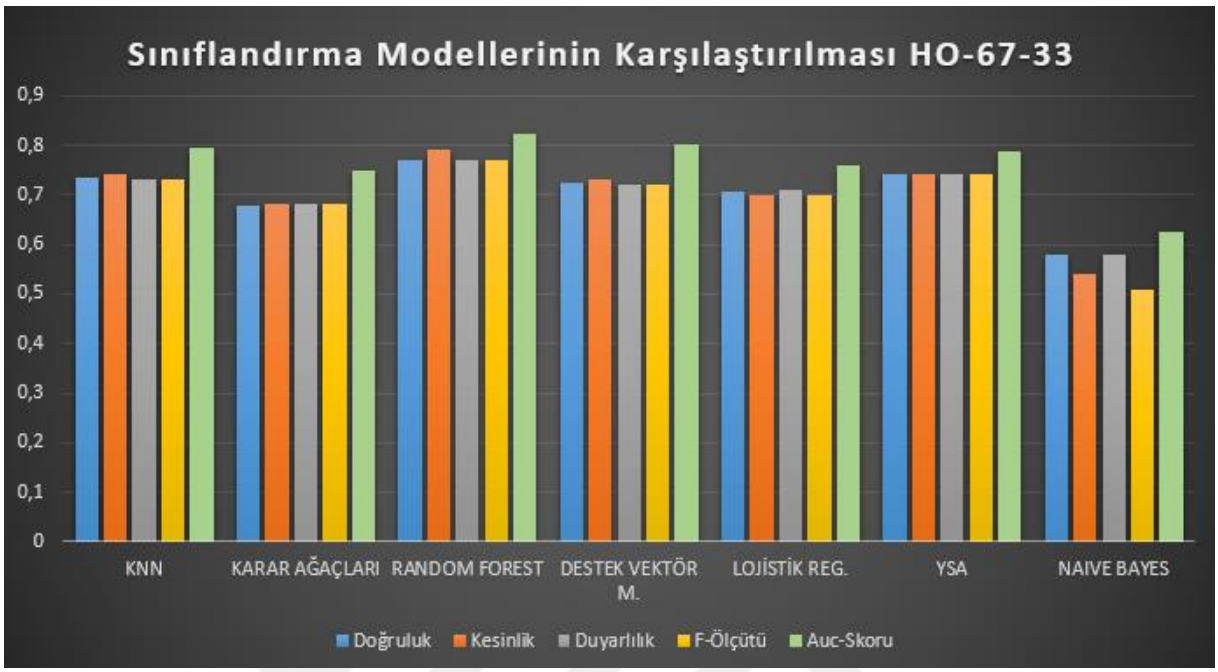
Şekil 4.26: Normalize veri seti için modellere ait karmaşıklık matrisleri (hold-out 67-33).

Modellere ait sınıflandırma raporları Tablo 4.8'de gösterilmektedir.

Tablo 4.8: Normalize veri seti hold-out 67/33 için modellere ait sınıflandırma raporları.

K-En Yakın Komşu				
Karar Niteliği Sınıfı	Kesinlik	Duyarlılık	F-Ölçütü	Sınıfa ait Örnek Sayısı
2	1.00	0.50	0.67	6
3	0.76	0.68	0.72	28
4	0.71	0.73	0.72	48
5	0.72	0.85	0.78	27
Karar Ağaçları				
Karar Niteliği Sınıfı	Kesinlik	Duyarlılık	F-Ölçütü	Sınıfa ait Örnek Sayısı
2	0.50	0.33	0.40	6
3	0.62	0.71	0.67	28
4	0.72	0.65	0.68	48
5	0.79	0.78	0.74	27
Rastgele Orman				
Karar Niteliği Sınıfı	Kesinlik	Duyarlılık	F-Ölçütü	Sınıfa ait Örnek Sayısı
2	0.50	0.67	0.57	6
3	0.88	0.54	0.67	28
4	0.74	0.88	0.80	48
5	0.88	0.85	0.85	27
Destek Vektör Makineleri				
Karar Niteliği Sınıfı	Kesinlik	Duyarlılık	F-Ölçütü	Sınıfa ait Örnek Sayısı
2	0.50	0.67	0.57	6
3	0.70	0.57	0.63	28
4	0.73	0.75	0.74	48
5	0.79	0.85	0.82	27
Yapay Sinir Ağları				
Karar Niteliği Sınıfı	Kesinlik	Duyarlılık	F-Ölçütü	Sınıfa ait Örnek Sayısı
2	0.67	0.33	0.44	6
3	0.68	0.75	0.71	28
4	0.77	0.71	0.74	48
5	0.77	0.89	0.83	27
Lojistik Regresyon				
Karar Niteliği Sınıfı	Kesinlik	Duyarlılık	F-Ölçütü	Sınıfa ait Örnek Sayısı
2	0.67	0.33	0.44	6
3	0.65	0.61	0.63	28
4	0.70	0.73	0.71	48
5	0.77	0.85	0.81	27
Naive Bayes				
Karar Niteliği Sınıfı	Kesinlik	Duyarlılık	F-Ölçütü	Sınıfa ait Örnek Sayısı
2	0.00	0.00	0.00	6
3	0.50	0.11	0.18	28
4	0.51	0.79	0.62	48
5	0.76	0.81	0.79	27

Sınıflandırma modellerinin başarım oranlarına ait grafik Şekil 4.27'de gösterilmektedir.



Şekil 4.27: Normalize veri seti HO-67/33 sınıflandırma modellerinin performans karşılaştırma grafiği.

Normalize veri setinin Hold-out %60/%40 oranlarında ayrılması ile kurulan modellere ait parametre seçimleri aşağıda belirtildiği gibidir.

- Rastgele Orman algoritması için ağaç sayısı parametresi (n_estimators) 10 olarak uygulanmıştır.
- K-En Yakın Komşu Algoritması için k=9 ve uzaklık ölçütü olarak "Manhattan" uzaklığı kullanılmıştır.
- Karar Ağaçları Algoritması için maksimum derinlik parametresi (max_depth) 3, bölünme kriteri (criterion) ise "gini" olarak uygulanmıştır.
- Yapay Sinir Ağları modeli giriş katmanı, üç adet gizli katman ve çıkış katmanından oluşmaktadır. Giriş katmanındaki nöron sayısı 11, gizli katmanlardaki nöron sayıları ise sırasıyla 11, 22 ve 6 olarak uygulanmıştır. Aktivasyon fonksiyonu olarak Hiperbolik Tanjant (tanh) fonksiyonu kullanılmış olup maksimum iterasyon parametresi 300 olarak ayarlanmıştır.
- Destek Vektör Makineleri için çekirdek parametresi "linear" olarak uygulanmıştır.

- Lojistik Regresyon algoritması için; multi_class parametresi "multinomial", çözücü fonksiyonu belirlememizi sağlayan solver parametresi "lbfgs", maksimum iterasyon sayısı ise 500 olarak uygulanmıştır.
- Naive Bayes algoritması için "Bernoulli" parametresi uygulanmıştır.

Veri setinin Hold-out %60/%40 oranlarında eğitim ve test kümelerine ayrılması ile kurulan modellere ait sonuçlar Tablo 4.9'da gösterilmektedir.

Tablo 4.9: Normalize veri seti için hold-out %60/%40 modellerin performans değerlendirmesi.

Algoritma	**Doğruluk**	**Kesinlik**	**Duyarlılık**	**F-Ölçütü**	**Auc-Skoru**
Knn	0.7121	0.72	0.71	0.71	7671
Karar Ağaçları	0.7424	0.74	0.74	0.74	7712
Rastgele Orman	**0.7803**	**0.78**	**0.78**	**0.77**	7870
DVM	0.7348	0.74	0.73	0.73	8057
Logistik Reg.	0.7500	0.75	0.76	0.75	**8097**
YSA	0.7348	0.75	0.73	0.72	7598
Naive Bayes	0.5833	0.53	0.58	0.54	6447

Tablo 4.9. incelendiğinde Auc-skoru haricinde en yüksek başarı sağlayan algoritmanın Rastgele Orman algoritması olduğu görülmektedir. Onu sırasıyla izlemekte olan Lojistik Regresyon, Karar Ağaçları, Yapay Sinir Ağları ve Destek Vektör Makineleri birbirlerine oldukça yakın başarım oranlarına sahip olmuşlardır. Lojistik Regresyon algoritması en yüksek Auc-skoruna sahipt olmaktadır. Naive Bayes algoritması diğer modellerle kıyaslandığında oldukça düşük performans göstermiştir. Modellere ait Karmaşıklık Matrisleri Şekil 4.28'de gösterilmektedir.

Knn	Tahmin Edilen Değerler			
	2	3	4	5
Gerçek D.				
2	3	3	1	0
3	0	20	14	0
4	0	6	44	8
5	0	0	6	27

Karar Ağaçları	Tahmin Edilen Değerler			
	2	3	4	5
Gerçek D.				
2	2	5	0	0
3	3	24	7	0
4	0	4	47	7
5	0	0	8	25

Rastgele Orman	Tahmin Edilen Değerler			
	2	3	4	5
Gerçek D.				
2	2	4	1	0
3	2	26	6	0
4	0	3	52	3
5	0	0	10	23

Destek Vektör M.	Tahmin Edilen Değerler			
	2	3	4	5
Gerçek D.				
2	4	2	1	0
3	4	23	7	0
4	0	9	41	8
5	0	0	4	29

YSA	Tahmin Edilen Değerler			
	2	3	4	5
Gerçek D.				
2	1	5	1	0
3	0	25	9	0
4	0	8	42	8
5	0	0	4	29

Lojistik Reg.	Tahmin Edilen Değerler			
	2	3	4	5
Gerçek D.				
2	4	2	1	0
3	0	24	10	0
4	0	8	43	7
5	0	0	5	28

Naive Bayes	Tahmin Edilen Değerler			
	2	3	4	5
Gerçek D.				
2	0	0	7	0
3	0	7	26	1
4	0	8	38	12
5	0	0	1	32

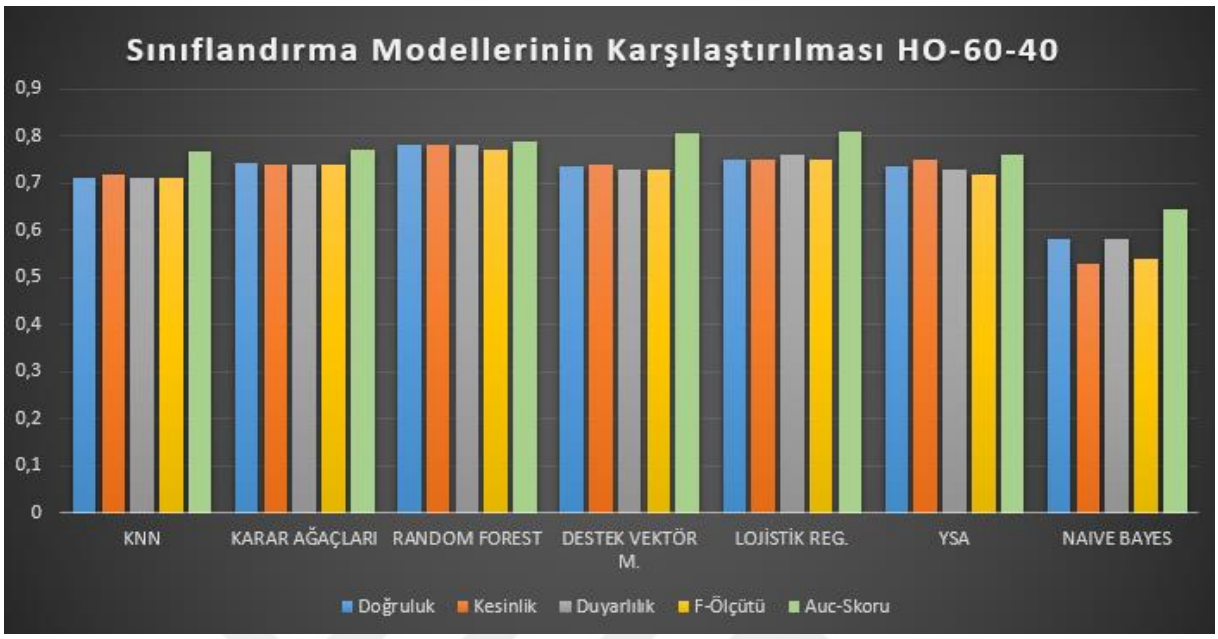
Şekil 4.28: Normalize veri seti için modellere ait karmaşıklık matrisleri (hold-out 60-40).

Algoritmalara ait sınıflandırma raporları Tablo 4.10'da gösterilmektedir.

Tablo 4.10: Normalize veri seti hold-out 60/40 için modellere ait sınıflandırma raporları.

K-En Yakın Komşu				
Karar Niteliği Sınıfı	Kesinlik	Duyarlılık	F-Ölçütü	Sınıfa ait Örnek Sayısı
2	1.00	0.43	0.60	7
3	0.69	0.59	0.63	34
4	0.68	0.76	0.72	58
5	0.77	0.82	0.79	33
Karar Ağaçları				
Karar Niteliği Sınıfı	Kesinlik	Duyarlılık	F-Ölçütü	Sınıfa ait Örnek Sayısı
2	0.40	0.29	0.33	7
3	0.73	0.71	0.72	34
4	0.76	0.81	0.78	58
5	0.78	0.76	0.77	33
Rastgele Orman				
Karar Niteliği Sınıfı	Kesinlik	Duyarlılık	F-Ölçütü	Sınıfa ait Örnek Sayısı
2	0.50	0.29	0.36	7
3	0.79	0.76	0.78	34
4	0.75	0.90	0.82	58
5	0.88	0.70	0.78	33
Destek Vektör Makineleri				
Karar Niteliği Sınıfı	Kesinlik	Duyarlılık	F-Ölçütü	Sınıfa ait Örnek Sayısı
2	0.50	0.57	0.53	7
3	0.68	0.68	0.68	34
4	0.77	0.71	0.74	58
5	0.78	0.88	0.83	33
Yapay Sinir Ağları				
Karar Niteliği Sınıfı	Kesinlik	Duyarlılık	F-Ölçütü	Sınıfa ait Örnek Sayısı
2	1.00	0.14	0.25	7
3	0.66	0.74	0.69	34
4	0.75	0.72	0.74	58
5	0.78	0.88	0.83	33
Lojistik Regresyon				
Karar Niteliği Sınıfı	Kesinlik	Duyarlılık	F-Ölçütü	Sınıfa ait Örnek Sayısı
2	1.00	0.57	0.73	7
3	0.71	0.71	0.71	34
4	0.73	0.74	0.74	58
5	0.80	0.85	0.82	33
Naive Bayes				
Karar Niteliği Sınıfı	Kesinlik	Duyarlılık	F-Ölçütü	Sınıfa ait Örnek Sayısı
2	0.00	0.00	0.00	7
3	0.47	0.21	0.29	34
4	0.53	0.66	0.58	58
5	0.71	0.97	0.82	33

Sınıflandırma modellerinin başarım oranlarına ait grafik Şekil 4.29'da gösterilmektedir.



Şekil 4.29: Normalize veri seti HO-60/40 sınıflandırma modellerinin performans karşılaştırma grafiği.

Normalize veri setinin çapraz geçerleme yöntemi ile 5-kat olarak bölünmesi ile elde edilen her bir kat için doğruluk oranı ve tüm katların ortalama doğruluk oranları Tablo 4.11'de gösterilmektedir.

Tablo 4.11: Normalize veri seti 5-kat çapraz geçerleme doğruluk oranları.

Algoritma	**k-1**	**k-2**	**k-3**	**k-4**	**k-5**	**k-ort**
Knn	0.5681	0.7045	**0.7500**	**0.7045**	**0.7441**	0.6942
Karar A.	0.6818	0.6818	0.6590	0.6818	0.5813	0.6571
R. Orman	0.7272	**0.8409**	0.7045	**0.7045**	**0.7441**	**0.7442**
DVM	0.7045	0.7500	0.7045	0.6136	0.6511	0.6847
Loj. Reg.	0.7045	0.7954	0.7045	0.6590	0.62279	0.6983
YSA	**0.7727**	0.7272	0.7045	**0.7045**	0.6279	0.7073
Nb	0.6363	0.6818	0.5000	0.6363	0.6976	0.6304

Tablo 4.11. incelendiğinde 5-kat çapraz geçerleme ortalama doğruluk oranı en yüksek modelin Rastgele Orman Algoritmasına ait olduğu görülmektedir. Onu sırasıyla, Yapay Sinir Ağları, Lojistik Regresyon, K-En Yakın Komşu, Destek Vektör Makineleri, Karar Ağaçları ve Naive Bayes modelleri izlemektedir.

Veri setinin çapraz geçerleme yöntemi ile 10-kat olarak bölünmesi ile elde edilen her bir kat için doğruluk oranı ve tüm katların ortalama doğruluk oranları Tablo 4.12'de gösterilmektedir.

Tablo 4.12: Normalize veri seti 10-kat çapraz geçerleme doğruluk oranları.

Algoritma	k-1	k-2	k-3	k-4	k-5	k-6	k-7	k-8	k-9	k-10	k-ort
Knn	0.5909	0.5909	**0.7727**	0.6818	**0.8636**	0.6363	0.6363	0.7727	0.5909	0.7142	0.6850
Karar A.	0.6363	0.7727	0.7272	0.7727	0.7272	0.5454	0.5909	**0.8181**	0.5909	0.5238	0.6705
R. Orman	**0.8181**	0.8181	0.7272	**0.8636**	0.7727	0.6818	**0.7272**	0.7727	0.7272	0.6666	**0.7575**
DVM	0.7727	0.7727	0.7272	0.8181	0.7727	**0.7272**	0.5454	0.7272	**0.7727**	0.5714	0.7207
Loj. Reg.	0.7272	0.7727	0.7272	**0.8636**	0.7727	0.6818	0.5909	0.7727	0.7272	0.5714	0.7207
YSA	**0.8181**	**0.8636**	0.6363	0.8181	0.7727	0.5454	0.6818	0.7727	0.6363	0.6666	0.7212
Nb	0.7727	0.5909	0.6363	0.8181	0.5454	0.5000	0.5909	0.6363	0.5909	**0.7619**	0.6443

Tablo 4.12. incelendiğinde 10-kat çapraz geçerleme ortalama doğruluk oranı en yüksek modelin Rastgele Orman Algoritmasına ait olduğu görülmektedir. Onu sırasıyla, Yapay Sinir Ağları, Lojistik Regresyon, Destek Vektör Makineleri, K-En Yakın Komşu, Karar Ağaçları ve Naive Bayes modelleri izlemektedir.

4.4. TEMEL VERİ SETİ VE NORMALİZE VERİ SETİNE AİT SONUÇLARIN KARŞILAŞTIRILMASI

Temel veri seti ve normalize veri setine uygulanan makine öğrenmesi modellerinin performans değerlendirmeleri karşılaştırılmıştır (Tablo 4.13).

Tablo 4.13: Tüm modellerin karşılaştırmalı performans değerlendirmesi.

Accuracy /Veri seti	Normalize Edilmemiş (Temel) Veri Seti				Normalize Veri Seti			
	HO (67-33)	**HO 60-40**	**k-5 fold**	**k-10 fold**	**HO (67-33)**	**HO 60-40**	**k-5 fold**	**k-10 fold**
KNN	0.7889	**0.8030**	**0.7486**	0.7305	0.7339	0.7121	0.6942	0.6850
KARAR AĞAÇLARI	0.6972	0.7575	0.6571	0.6796	0.6788	0.7424	0.6571	0.6705
RASTGELE ORMAN	**0.8073**	0.7954	0.7442	**0.7575**	**0.7706**	**0.7803**	**0.7442**	**0.7575**
DESTEK VEKTÖR MAKİNELERİ	0.6880	0.7424	0.6937	0.7207	0.7247	0.7348	0.6847	0.7207
LOJİSTİK REGRESYON	0.6697	0.6060	0.5933	0.6253	0.7064	0.7500	0.6983	0.7207
YSA	0.6238	0.5833	0.5341	0.5430	0.7431	0.7348	0.7073	0.7212
NAIVE BAYES	0.4587	0.4621	0.5108	0.4969	0.5779	0.5833	0.6304	0.6443

Çalışmada geliştirilen tüm makine öğrenmesi modelleri karşılaştırıldığında; en başarılı modelin, Rastgele Orman algoritmasının temel veri seti üzerinde ve hold-out 67/33 bölme oranı ile uygulanması sonucunda elde edildiği görülmektedir.

4.5. MODELİN UYGULAMAYA GEÇİRİLMESİ

En yüksek başarım oranlarına ulaşan makine öğrenmesi modeli kullanılarak öğrencilerin akademik başarı tahminine yönelik, yerel sunucuda çalışan basit ara yüze sahip bir sistem geliştirilmiştir. Yazılımın tasarımı için yararlanılan programlar ve kütüphaneler Tablo 4.14'de belirtilmektedir.

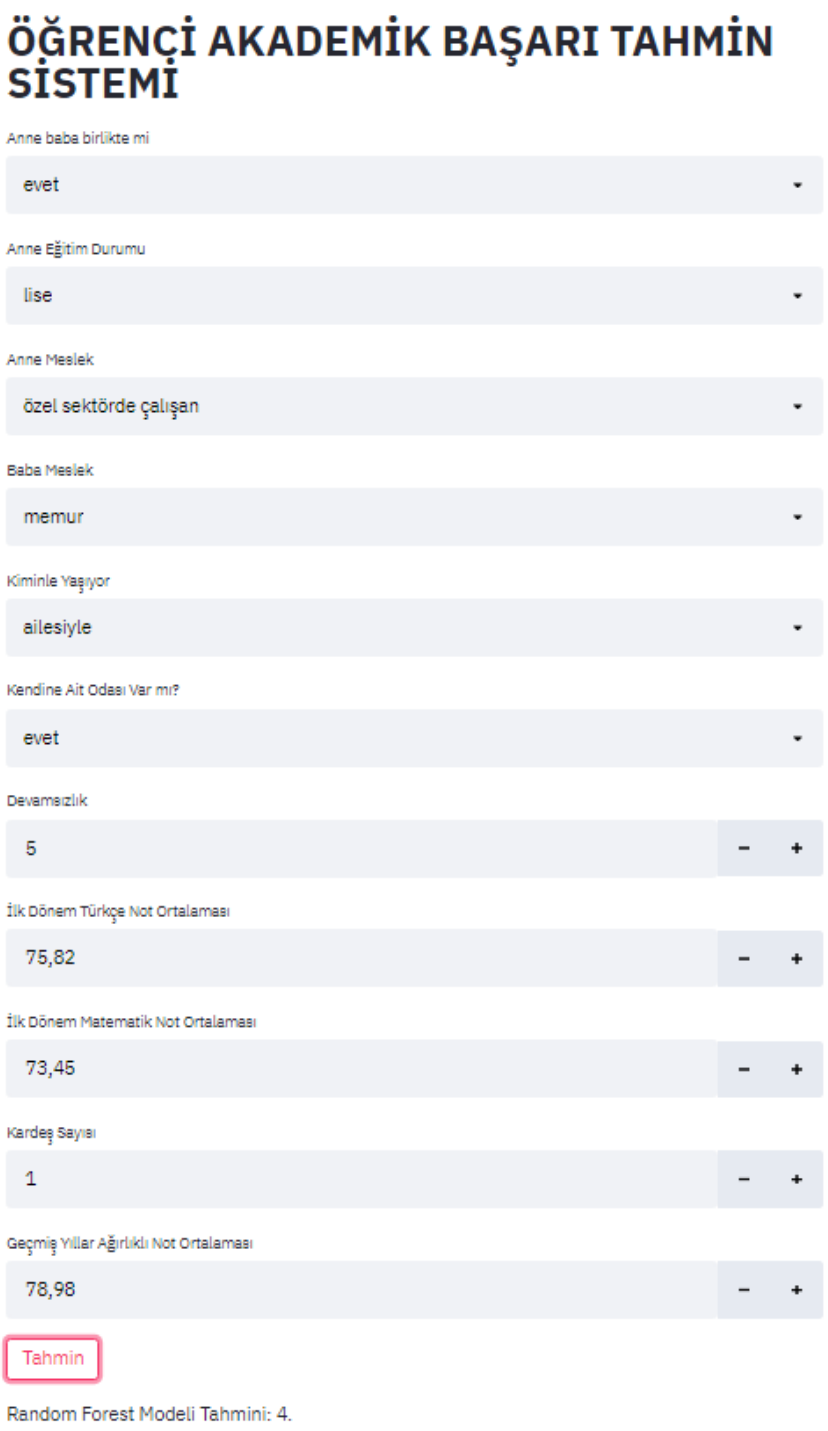
Tablo 4.14: Geliştirilen yazılımda kullanılan programlar ve kütüphaneler.

Program/Kütüphane Adı	Sürüm Bilgisi	Kullanım Amacı	Kaynak
Streamlit	0.74.1	Geliştirme ortamı - Python kütüphanesi	Teixeira ve diğ., 2021
Heroku	-	Bulut tabanlı web platform servisi	Hanjura, 2014
Python	3.6.5	Programlama dili	Van Rossum ve Drake, 2009
Numpy	1.14.3	Bilimsel hesaplamalar için kullanılan kütüphane	Harris ve diğ., 2020
Pandas	0.23.0	Veri seçme ve dönüştürme işlemleri	McKinney., 2010
Pickle	0.7.5	Modelin kaydedilmesi ve kayıtlı modelin kullanılması	Van Rossum ve diğ., 2020

Sistem aşağıda belirtilen uygulama adımları takip edilerek oluşturulmuştur.

- Rastgele orman modeli, "Pickle" fonksiyonu kullanılarak kaydedilmiştir.
- "Streamlit" çatısı (framework) altında sınıflandırıcının eğitiminde kullanılan özniteliklerin kullanıcı tarafından girilmesini sağlayacak bir ara yüz oluşturulmuştur.
- Ara yüz aracılığıyla girilen değerlerin kayıtlı makine öğrenmesi modeli ile uyumlu hale getirilmesi için gerekli veri dönüşümlerini sağlayan düzenlemeler yapılmıştır.
- Butona basıldığında, eğitilmiş model tarafından sene sonu ağırlıklı not ortalamasının tahmin edilerek ekrana yazdırılmasını sağlayan düzenlemeler yapılmıştır.
- Streamlit çatısı altında çalışan yazılım modeli Heroku web platformu bağlantısı yapılarak web ortamına aktarılmıştır. https://model-tahmin.herokuapp.com adresinden modele ulaşılabilmektedir.

Geliştirilen yazılıma ait örnek ekran görüntüsü Şekil 4.30'da paylaşılmaktadır.



Şekil 4.30: Geliştirilen yazılıma ait örnek ekran görüntüsü.

Şekil 4.30'da görüldüğü üzere; kullanıcı tarafından öğrenciye ait gerekli bağımsız değişkenler sistemin ara yüzü kullanılarak girilebilmektedir. "Tahmin" butonuna basıldığında önceden eğitilmiş olan rastgele orman modeli çalışarak öğrencinin sene sonu ağırlıklı not ortalaması

tahmini ekrana yazdırılmaktadır. Sistem tarafından üretilen çıktı değeri Tablo 3.3'de belirtilen MEB'in 5'lik sistem not dönüşüm çizelgesindeki not değerlerini ifade etmektedir.

5. TARTIŞMA VE SONUÇ

Akademik başarıyı etkileyen çok fazla etken bulunmaktadır. Bu etkenlerin bazılarına (öğrencinin psikolojik durumu, aile içi durum vb.) güvenilir veri olarak erişmek zor olabilmektedir. Aynı zamanda başarı tahmininde kullanılan öznitelikler bireyler üzerinde farklı etkiler yaratabilmektedir. Bu gibi nedenlerden dolayı akademik başarı tahmini zor bir işlem olarak görünmektedir (Aydemir, 2017). Bu çalışmada akademik başarının tahmin edilmesi amacıyla demografik, sosyoekonomik ve akademik geçmişe ait öznitelikler kullanılmıştır.

Literatür incelendiğinde akademik başarı tahminine yönelik çok sayıda çalışma olup ortaokul seviyesine dair çalışmaların az sayıda olduğu görülmektedir (Bölüm 2.4). Bu tez çalışması ortaokul öğrencilerinin akademik başarı tahminine yönelik olması nedeniyle bu alandaki literatüre katkı sağlayacağı düşünülmektedir.

Bu tez çalışmasının amacı sınıflandırmaya dayalı makine öğrenmesi yöntemleri kullanılarak ortaokul öğrencilerinin akademik başarılarının önceden tahmin edilmesi ve başarıya en fazla etki eden faktörlerin tespit edilmesidir. Çalışma kapsamında 7 algoritma ve 5 değerlendirme ölçütü kullanılarak sınıflandırma modelleri oluşturulmuştur. Kullanılan algoritmalar Karar Ağaçları Algoritması, Rastgele Orman Algoritması, K-En Yakın Komşu Algoritması, Destek Vektör Makineleri, Naive Bayes Algoritması, Lojistik Regresyon Algoritması ve Yapay Sinir Ağları olup önceki çalışmalar ile uyumludur (Tablo 2.6). Hedef değişken 0-5 arası kategorik değerlere sahip olmakla birlikte kullanılan veri setinde benzersiz sınıf değerleri 2-5 aralığında 4 seviyelidir.

Cortez ve Silva (2008), Şen ve diğ. (2012), Guo ve diğ. (2015), Abbasoğlu (2020) çalışmalarında örneklem olarak ortaöğretim öğrencilerine ait veri kullanmış ve hedef değişkenleri 5 seviyelidir. Bu tez çalışmasında da literatüre benzer şekilde 5 seviyeli hedef değişken kullanılarak ortaöğretim öğrencilerine ait veri analiz edilmiştir.

Bu tez çalışmasında geliştirilen sınıflandırma modelleri arasında en başarılı sonucu, temel veri seti ile yapılan analizde %80.73 doğruluk oranı ile Rastgele Orman algoritması vermiştir. Kesinlik, duyarlılık, F-ölçütü ve Roc-auc skoru değerleri de algoritmanın başarısını doğrular niteliktedir ve modelin güvenilirliğini artırmaktadır. Abbasoğlu (2020), çalışmasında Lojistik Regresyon algoritması %64.13 doğruluk oranı ile en başarılı sonucu elde etmiştir. Sınıflandırma

başarısının ortalama seviyede olduğu görülmekte olup bu durumun, veri setinde geçmiş akademik başarıya dair değişkenlerin kullanılmamasından kaynaklandığı düşünülmektedir. Başka bir çalışmada (Cortez ve Silva, 2008), en yüksek model başarımı %78,5 doğruluk oranı ile Naive Bayes algoritması tarafından sağlanmıştır. En başarılı model bu tez çalışması ile farklılık göstermekte olup model başarımı açısından yakın sonuçlar elde edilmiştir. Şen ve diğ. (2012), çalışmalarında en yüksek başarım oranları Karar Ağaçları modeli ile sağlanmıştır ve bu tez çalışması ile farklılık göstermektedir. %94,5 doğruluk oranına ulaşılmış olup bu tez çalışmasından daha yüksek bir değer elde edilmiştir. Bu durumun, veri setindeki bağımsız değişkenlerin ağırlıklı olarak not bilgilerinden oluşmasından kaynaklandığı düşünülmektedir. Bir diğer çalışmada (Guo ve diğ., 2015), en başarılı model %77,2 doğruluk oranına sahip derin öğrenme modeli olmuştur. Model başarım oranı mevcut tez çalışması ile benzerlik göstermektedir.

Sınıflandırma modellerinde hedef niteliğe ait sınıf değerleri sayısı arttıkça model performansları doğal olarak düşmektedir. Bu çalışmada 4 seviyeli bağımlı değişkenin tahmini yapılmış olup kabul edilebilir seviyede başarılı sonuçlar elde edildiği düşünülmektedir. Tahmin başarısının yüksek olmasında, öğrencilerin Türkçe, Matematik derslerine ait notları ile önceki yıllar not ortalamalarının model eğitiminde kullanılmasının büyük etkisi olduğu belirlenmiştir.

Çalışmanın diğer odak noktası olan başarıya en fazla etki eden faktörlerin belirlenmesi kapsamında aşağıda belirtilen bilgilere ulaşılmıştır:

- Akademik geçmişe ait değişkenler olan geçmiş yıllar not ortalaması, Türkçe dersi 1. dönem ortalaması ve Matematik dersi 1. dönem ortalaması başarıyı en fazla etkileyen öznitelikler olarak tespit edilmiştir. Aydın (2007), Cortez ve Silva (2008), Ramaswami ve Bhaskaran (2009), Şen ve diğ. (2012), Göker (2012), Acharya ve Sinha (2014), Devasia ve diğ. (2016), Tomasevic ve diğ. (2020) çalışmalarında bu tez çalışması ile benzer şekilde, geçmiş akademik başarıya ait değişkenlerin başarıya en fazla etki eden öznitelikler olduğunu belirtmişlerdir. Bu nedenle akademik geçmişin başarı üzerinde yüksek belirleyiciliği olduğu söylenebilir.
- Devamsızlık durumu başarıyı etkileyen bir diğer önemli faktör olarak belirlenmiştir. Bu durum literatür (Göker, 2012; Özdemir, 2016; Uzel ve diğ., 2018; Abbasoğlu, 2020) ile uyumludur.

- Demografik ve sosyoekonomik özelliklerin akademik başarıya çok fazla etkisi olmamakla birlikte tahmin başarısına katkısı olduğu görülmüştür. Öğrencinin kiminle yaşadığı, anne babasının birlikte olup olmadığı, babasının mesleği, annesinin mesleği, annesinin eğitim durumu, kendisine ait odasının olup olmadığı ve kardeş sayısı öznitelikleri başarıya en fazla etki eden demografik ve sosyoekonomik özellikler olarak bulunmuştur. Çalışma bu yönüyle literatürdeki çalışmalar ile benzerlik göstermektedir (Cortez ve Silva, 2008; Göker, 2012; Depren ve diğ., 2017; Abbasoğlu, 2020).
- Benzer çalışmalarda (Göker, 2012; Acharya ve Sinha, 2014; Devasia ve diğ., 2016; Gök, 2017; Uzel ve diğ., 2018) yüksek önem derecesine sahip olarak bulunan baba eğitim durumu, gelir durumu, burs alma durumu ve cinsiyet değişkenlerinin bu çalışmada yapılan özellik seçimi analizi sonuçlarına göre bağımlı değişkene etkisinin düşük olduğu tespit edilmiştir.
- Literatürde (Özdemir, 2016; Gök, 2017; Depren ve diğ., 2017; Filiz, 2019; Gomes ve Jelihovschi, 2019) hedef nitelik üzerinde önemli etkisi olduğu belirtilen motivasyon, ailenin ders konusundaki desteği ve kurs alma durumu vb. öznitelikleri bu çalışmadaki veri setinde bulunmamaktadır.

Bu çalışma İstanbul ilinde bulunan tek bir ortaokulda eğitim görmekte olan öğrencilere ait veri seti ile gerçekleştirilmiştir. Farklı örneklemler üzerinde uygulandığı takdirde model başarım oranları ve belirleyici nitelikler değişkenlik gösterebilir. Örneklemin genişletilmesi durumunda daha fazla ve daha çeşitli veriye ulaşılacağı için daha başarılı sonuçlar elde edilebileceği düşünülmektedir.

Çalışmanın yüksek seviyeli hedef değişken tahmini için yüksek sayılabilecek model başarım oranlarına sahip olduğu düşünülmektedir. Ortaokul öğrencileri için akademik başarı tahminine yönelik bir sistem geliştirilmesi yönüyle özgün bir çalışma olduğu ve literatüre katkı sağlayabileceği düşünülmektedir. Çalışmada tasarlanan sistemin daha fazla ve çeşitli veri ile modellenerek geliştirildiği takdirde;

- Karar verici konumdaki kurumlar açısından ortaöğretim stratejilerinin belirlenmesinde yardımcı olabilir.
- Ortaokul kademesindeki eğitim yöneticilerine ve öğretmenlere akademik başarı konusunda erken tedbirlerin alınması için yol gösterici olabilir.

- Öğrenci ve velileri tarafından kullanılarak öğrencilerin yakın gelecekteki akademik başarıları hakkında bir fikir edinilebilir.

Günümüzde veri madenciliği ve makine öğrenmesi yöntemleri ile gerçekleştirilen akademik başarı tahmininin, eğitimin planlanması ve önleyici eğitim açısından önemli olduğu ve geliştirilmesi gerektiği düşünülmektedir. Bu tez çalışmasında olduğu gibi geliştirilecek olan sistemler sayesinde akademik başarı yönünden risk grubunda olan öğrencilerin tespit edilmesi ve önleyici tedbirlerin alınması sağlanabilir. Gelecek çalışmalar için daha geniş kapsamlı veri setleri ve farklı yapay zeka yöntemleri kullanılarak geliştirilecek modeller ile başarılı sonuçlar elde edilebilir. Makine öğrenmesi yöntemleri ile geliştirilecek sistemlerin eğitime olumlu katkı sağlayacağı düşünülmektedir.

KAYNAKLAR

Abbasoğlu, B., 2020, Ortaokul Öğrencilerinin Akademik Başarılarının Eğitsel Veri Madenciliği Yöntemleri ile Tahmini, *Veri Bilimi Dergisi*, 3 (1), 1-10.

Acharya, A. ve Sinha, D., 2014, Early Prediction of Students Performance Using Machine Learning Techniques, *International Journal of Computer Applications*, 107 (1).

Adejo, O.W. ve Connolly, T., 2018, Predicting Student Academic Performance Using Multi-Model Heterogeneous Ensemble Approach, *Journal of Applied Research in Higher Education*.

Adekitan, A.I. ve Salau, O., 2020, Toward an Improved Learning Process: The Relevance of Ethnicity to Data Mining Prediction of Students' Performance, *SN Applied Sciences*, 2 (1), 8.

Agarwal, S., Pandey, G.N. ve Tiwari, M.D., 2012, Data Mining in Education: Data Classification and Decision Tree Approach, *International Journal of e-Education, e-Business, e-Management and e-Learning*, 2 (2), 140.

Agarwal, S., 2013, Data mining: Data Mining Concepts and Techniques, *In 2013 International Conference on Machine Intelligence and Research Advancement*, 203-207, IEEE.

Akhtar, Z. ve Niazi, H.K., 2011, The Relationship Between Socio-Economic Status and Learning Achievement of Students at Secondary Level, *International Journal of Academic Research*, 3 (2).

Akın, Y.K., 2008, *Veri Madenciliğinde Kümeleme Algoritmaları ve Kümeleme Analizi*, Doktora Tezi, Marmara Üniversitesi.

Akpınar, H., 2017, *Data: Veri Madenciliği Veri Analizi*, 2. baskı, Papatya Yayıncılık Eğitim, İstanbul, ISBN: 978-605-4220-81-6.

Alpaydın, E., 2014, *Introduction to Machine Learning*, MIT Press, ISBN: 978-0-262-02818-9.

Anaconda, 2020, Anaconda Software Distribution, Anaconda Inc., https://docs.anaconda.com/, [Ziyaret Tarihi: 23.11.2020].

Anyoha, R., 2017, The History of Artificial Intelligence, http://sitn.hms.harvard.edu/flash/2017/history-artificial-intelligence/, [Ziyaret Tarihi: 23.11.2020].

Asif, R., Merceron, A., Ali, S.A. ve diğ., 2017, Analyzing Undergraduate Students' Performance Using Educational Data Mining, *Computers & Education*, 113, 177-194.

Aydemir, B., 2017, *Veri Madenciliği Yöntemleri Kullanarak Meslek Yüksek Okulu Öğrencilerinin Akademik Başarı Tahmini*, Yüksek Lisans Tezi, Pamukkale Üniversitesi Fen Bilimleri Enstitüsü.

Aydın, S., 2007, *Veri Madenciliği ve Anadolu Üniversitesi Uzaktan Eğitim Sisteminde Bir Uygulama*, Doktora Tezi, Anadolu Üniversitesi Sosyal Bilimler Enstitüsü.

Ayhan, S. ve Erdoğmuş, Ş., 2014, Destek Vektör Makineleriyle Sınıflandırma Problemlerinin Çözümü için Çekirdek Fonksiyonu Seçimi, *Eskişehir Osmangazi Üniversitesi İktisadi ve İdari Bilimler Dergisi*, 9 (1), 175-201.

Azar, A.T., Elshazly, H.I., Hassanien, A.E. ve diğ., 2014, A Random Forest Classifier for Lymph Diseases, *Computer Methods and Programs in Biomedicine*, 113 (2), 465-473.

Azevedo, A.I.R.L., Santos, M.F., 2008, KDD, SEMMA and CRISP-DM: A parallel overview, http://recipp.ipp.pt/bitstream/10400.22/136/3/KDD-CRISP-SEMMA.pdf, [Ziyaret Tarihi: 11.01.2021].

Bayes, M. ve Price, M., 1763, An Essay towards Solving a Problem in the Doctrine of Chances, By the Late Rev. Mr. Bayes, F.R.S. Communicated by Mr. Price, in a Letter to John Canton, A.M.F.R.S., *Royal Society of London*, İngiltere.

Belgiu, M. ve Dragut, L., 2016, Random Forest in Remote Sensing: A Review of Applications and Future Directions, *ISPRS Journal of Photogrammetry and Remote Sensing*, 114, 24-31.

Bilen, Ö., Hotaman, D., Aşkın, Ö.E. ve diğ., 2014, LYS Başarılarına Göre Okul Performanslarının Eğitsel Veri Madenciliği Teknikleriyle İncelenmesi: 2011 İstanbul Örneği, *Eğitim ve Bilim*, 39 (172).

Boser, B.E., Guyon, I.M. ve Vapnik, V.N., 1992, A Training Algorithm for Optimal Margin Classifiers, *In Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, 144-152.

Bowles, M., 2015, *Machine Learning in Python : Essential Techniques for Predictive Analysis*, John Wiley & Sons, ProQuest Ebook Central, https://ebookcentral.proquest.com, ISBN: 978-1-1189-6176-6.

Budak, H., 2018, Özellik Seçim Yöntemleri ve Yeni Bir Yaklaşım, *Journal of Natural & Applied Sciences*.

Buttrey, S.E. ve Whitaker, L.R., 2017, *A Data Scientist's Guide to Acquiring, Cleaning, and Managing Data in R*, John Wiley & Sons, ProQuest Ebook Central, https://ebookcentral.proquest.com, ISBN: 978-1-1190-8006-0.

Breiman, L., 1999, 1 Random Forests--Random Features.

Breiman, L., 2001, *Random Forests, Machine Learning*, 45 (1), 5–32.

Breiman, L., Friedman, J., Stone, C.J. ve diğ., 1984, *Classification and Regression Trees*, Taylor & Francis, ISBN: 978-0-412-04841-8.

Chakrabarti, S., Cox, E., Frank, E. ve diğ., 2008, *Data Mining: Know It All*, Morgan Kaufmann Publishers is an Imprint of Elsevier, Burlington, ABD, ISBN: 978-0-12-374629-0.

Chang, C.C. ve Lin, C.J., 2011, LIBSVM: A Library for Support Vector Machines, *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2 (3), 1-27.

Chmielewski, M.R. ve Grzymala-Busse, J.W., 1996, Global Discretization of Continuous Attributes as Preprocessing for Machine Learning, *International Journal of Approximate Reasoning*, 15 (4), 319-331.

Cortes, C. ve Vapnik, V., 1995. Support-Vector Networks, *Machine Learning*, 20 (3), 273–297.

Cortez, P. ve Silva, A.M.G., 2008, Using Data Mining to Predict Secondary School Student Performance.

Çayıroğlu, İ., 2015, İleri Algoritma Analizi-5: Yapay Sinir Ağları, *Karabük Üniversitesi Mühendislik Fakültesi*.

Çetin, N. ve Mahir, N., 2006, Genel Matematik Dersindeki Öğrenci Başarısı ile Öss Başarısı Arasındaki İlişki, *İnönü Üniversitesi Eğitim Fakültesi Dergisi*, 7 (11), 37-46.

Chen, M.S., Han, J., Yu, P.S., 1996, Data Mining: An Overview from a Database Perspective, *IEEE Transactions on Knowledge and Data Engineering*, 8 (6), 866-883.

Çırak G., 2012, *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.

Cover, T.M. ve Hart, P.E., 1967, Nearest Neighbor Pattern Classification, *IEEE Transactions On İnformation Theory*, 13 (1), 21-27.

Dangeti, P., 2017, *Statistics for Machine Learning*, Packt Publishing Ltd.

Delen, D., 2010, A Comparative Analysis of Machine Learning Techniques for Student Retention Management, *Decision Support Systems*, 49 (4), 498-506.

Devasia, T., Vinushree, T.P. ve Hegde, V., 2016, Prediction of Students Performance Using Educational Data Mining, *International Conference on Data Mining and Advanced Computing (SAPIENCE)*, 91-95, IEEE.

Depren, S.K., Aşkın, Ö.E. ve Öz, E., 2017, Identifying the Classification Performances of Educational Data Mining Methods: A Case Study for TIMSS, *Educational Sciences: Theory & Practice*, 17 (5).

Dökeroğlu, T., Malik, Z.M.M. ve Shadi, A.S., 2018, Gözetimsiz Makine Öğrenme Teknikleri ile Miktara Dayalı Negatif Birliktelik Kural Madenciliği, *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, 6 (4), 1119-1138.

Emre, İ.E., 2017, *Veri Madenciliği ile Çocukluk Çağındaki Akut Romatizmal Ateşin Kalp Hastalığına Etkilerinin Analizi*, Yüksek Lisans Tezi, İstanbul Üniversitesi.

Erişlik, K., 2020, *Girişim Şirketlerinin Finansal Başarısızlıklarının Yapay Sinir Ağları ve Lojistik Regresyon Analizi ile Tahmin Edilmesi*, Yüksek Lisans Tezi, İstanbul Ticaret Üniversitesi.

Ertürk, S., 1994, *Eğitimde Program Geliştirme*, Meteksan Yayınları, Ankara.

Fayyad, U., Piatetsky-Shapiro, G. ve Smyth, P., 1996a, From Data Mining to Knowledge Discovery in Databases, *AI Magazine*, 17 (3), 37–54.

Fayyad, U., Piatetsky-Shapiro, G. ve Smyth, P., 1996b, The KDD Process for Extracting Useful Knowledge from Volumes of Data, *Communications of the ACM*, 39 (11), 27–34.

Fawcett, T., 2006, An introduction to ROC Analysis, *Pattern Recognition Letters*, 27 (8), 861-874.

Filiz, E., 2019, *Makine Öğrenmesi Yöntemleri ve Eğitim Verisi Üzerine Bir Uygulama: Uluslararası Matematik ve Fen Eğilimleri Araştırması 2015 Türkiye Örneği*, Doktora Tezi, Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü.

Flach, P., 2012, *Machine Learning: The Art and Science of Algorithms That Make Sense of Data*, Cambridge University Press, ABD, ISBN: 978-1-107-09639-4.

Fogarty, J., Baker, R.S. ve Hudson, S.E., 2005, Case Studies in the Use of ROC Curve Analysis for Sensor-Based Estimates in Human Computer Interaction, *Proceedings of Graphics Interface*, 129-136.

García, S., Luengo, J. ve Herrera, F., 2015, *Data Preprocessing in Data Mining*, Cham, Switzerland: Springer International Publishing, 195-243.

García, S., Ramírez-Gallego, S., Luengo, J. ve diğ., 2016, Big Data Preprocessing: Methods and Prospects, *Big Data Analytics*, 1 (1), 1-22.

Gardner, M.W., Dorling, S.R., 1998, Artificial Neural Networks (The Multilayer Perceptron)-A Review of Applications in the Atmospheric Sciences, *Atmospheric Environment*, 32 (14), 2627-2636.

Gelbal, S., 2010, Sekizinci Sınıf Öğrencilerinin Sosyoekonomik Özelliklerinin Türkçe Başarısı Üzerinde Etkisi, *Eğitim ve Bilim*, 33 (150).

Gomes, C.M.A. ve Jelihovschi, E., 2020, Presenting the Regression Tree Method and Its Application in a Large-Scale Educational Dataset, *International Journal of Research & Method in Education*, 43 (2), 201-221.

Gorunescu, F., 2011, *Data Mining - Concepts, Models and Techniques*, 1. baskı, Springer-Verlag Berlin Heidelberg, (Intelligent Systems Reference Library), ISBN: 978-3-642-19720-8.

Gök, M., 2017, Makine Öğrenmesi Yöntemleri ile Akademik Başarının Tahmin Edilmesi, *Gazi Üniversitesi Fen Bilimleri Dergisi Part C: Tasarım ve Teknoloji*, 5 (3), 139-148.

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

Grubbs, F.E., 1969, Procedures for Detecting Outlying Observations in Samples, *Technometrics*, 11 (1), 1-21.

Guo, B., Zhang, R., Xu, G. ve diğ., 2015, Predicting Students Performance in Educational Data Mining, *International Symposium on Educational Technology (ISET)*, 125-128, IEEE.

Hackeling, G., 2014, *Mastering Machine Learning with Scikit-Learn*, ProQuest Ebook Central, https://ebookcentral.proquest.com, ISBN: 978-1-7839-8837-2.

Hall, M.A., 1999, *Correlation-based Feature Selection for Machine Learning*, Doktora Tezi, The University of Waikato, Yeni Zelanda.

Han, J., Kamber, M. ve Pei, J., 2012, *Data Mining: Concepts and Techniques*, 3. baskı, Morgan Kaufman Publishers, ABD, ISBN: 978-0-12-381479-1.

Han, J. ve Kamber, M., 2006, *Data Mining: Concepts and Techniques (The Morgan Kaufmann Series in Data Management Systems)*, 2. baskı, Morgan Kaufmann Publishers, ISBN: 978-1-55860-901-3.

Hanjura, A., 2014, *Heroku Cloud Application Development*, Packt Publishing Ltd, Birmingham, İngiltere, ISBN: 978-1-78355-097-5.

Harrington, P., 2012, *Machine Learning in Action*, Manning Publications Co., ABD, ISBN: 978-1-61729-018-3.

Harris, C.R., Millman, K.J., van der Walt, S.J. ve diğ., 2020, Array Programming with NumPy, *Nature*, 585 (7825), 357-362.

Hastie, T., Tibshirani, R. ve Friedman, J., 2009, *The Elements of Statistical Learning: Data Mining, Inference and Prediction*, Springer Science & Business Media, ISBN: 978-0-387-84857-0.

Hosmer Jr, D.W., Lemeshow, S. ve Sturdivant, R.X., 2013, *Applied Logistic Regression*, John Wiley & Sons, ISBN: 978-0-47058-247-3.

Huang, J. ve Ling, C.X., 2005, Using AUC and Accuracy in Evaluating Learning Algorithms, *IEEE Transactions on knowledge and Data Engineering*, 17 (3), 299-310.

Hunter, J.D., 2007, Matplotlib: A 2D Graphics Environment, *Computing in Science & Engineering*, 9 (3), 90-95.

Jalota, C. ve Agrawal, R., 2019, Analysis of Educational Data Mining Using Classification, *International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, 243-247, IEEE.

John, G.H., Kohavi, R. ve Pfleger, K., 1994, Irrelevant Features and the Subset Selection Problem, *In Machine Learning Proceedings*, 121-129.

Jin, R., Breitbart, Y. ve Muoh, C., 2007, Data Discretization Unification, *Seventh IEEE International Conference on Data Mining (ICDM), Omaha Nebreska, IEEE Conference Publications*, 183-192.

Kaelbling, L.P., Littman, M.L. ve Moore, A.W., 1996, Reinforcement Learning: A Survey, *Journal Of Artificial Intelligence Research*, 4, 237-285.

Kantardzic, M., 2011, *Data Mining Concepts, Models, Methods and Algorithms*, John Wiley & Sons Inc., New Jersey, ISBN: 978-0-470-89045-5.

Karabulut, E. ve Alpar, R., 2011, *Lojistik Regresyon, Uygulamalı Çok Değişkenli İstatistiksel Yöntemler*, Detay Yayıncılık Ankara, ISBN: 978-605-5437-42-8.

Kartal, E., 2015, *Sınıflandırmaya Dayalı Makine Öğrenmesi Teknikleri ve Kardiyolojik Risk Değerlendirmesine İlişkin Bir Uygulama*, Doktora Tezi, İstanbul Üniversitesi.

Kass, G.V., 1980, An exploratory Technique for Investigating Large Quantities of Categorical Data, *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 29 (2), 119-127.

Kaur, P., Singh, M. ve Josan, G.S., 2015, Classification and Prediction Based Data Mining Algorithms to Predict Slow Learners in Education Sector, *Procedia Computer Science*, 57, 500-508.

Kim, J.H., 2009, Estimating Classification Error Rate: Repeated Cross-Validation, Repeated Holdout and Bootstrap, *Computational Statistics & Data Analysis*, 53 (11), 3735-3745.

Kleinbaum, D.G. ve Klein, M., 2010, *Logistic Regression: A Self-Learning Text*, Springer Science & Business Media, ISBN: 978-1-4419-1741-6.

Koçoğlu, F.Ö., 2017, *Müşteri Kayıp Analizi Probleminin Çözümünde Analitik Yaklaşımlar*, Doktora Tezi, İstanbul Üniversitesi.

Kodratoff, Y. ve Michalski, R.S., 1990, *Machine Learning: An Artificial Intelligence Approach*, 3. baskı, Morgan Kaufmann Publishers, ABD, ISBN: 1-55860-19-8.

Kohavi, R., 1995, A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection, *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, Stanford, CA, 1137–1145.

Kotsiantis, S.B., 2007, Supervised Machine Learning: A Review of Classification Techniques, *Informatica*, 31, 249-268.

Kotsiantis, S.B., Zaharakis, I. ve Pintelas, P., 2007, Supervised Machine Learning: A Review of Classification Techniques, *Emerging Artificial Intelligence Applications in Computer Engineering*, 160 (1), 3-24.

Kovacic, Z., 2010, Early Prediction of Student Success: Mining Students' Enrolment Data, *Proceedings of Informing Science & IT Education Conference*.

Köktürk, F., 2012, *K-En Yakın Komşuluk, Yapay Sinir Ağları ve Karar Ağaçları Yöntemlerinin Sınıflandırma Başarılarının Karşılaştırılması*, Doktora Tezi, Bülent Ecevit Üniversitesi.

Langley, P., 1996, *Elements of Machine Learning*, Morgan Kaufmann Publishers, ABD, ISBN: 1-55860-301-8.

Laradji, I.H., Mueller, A. ve Qian., J., 2020, *Multi-layer Perceptron*, https://github.com/scikit-learn/scikit-learn/blob/0fb307bf3/sklearn/neural_network/_multilayer_perceptron.py#L705, [Ziyaret Tarihi: 25.10.2020].

Lau, E.T., Sun, L. ve Yang, Q., 2019, Modelling, Prediction and Classification of Student Academic Performance Using Artificial Neural Networks, *SN Applied Sciences*.

Linoff, G.S., Berry, M.J.A., 2011, *Data Mining Techniques : For Marketing, Sales, and Customer Relationship Management*, John Wiley & Sons, New York, ProQuest Ebook Central, ISBN: 978-1-1180-8750-3.

Loh, W.Y. ve Shih, Y.S., 1997, Split Selection Methods for Classification Trees, *Statistica Sinica*, 815-840.

Louppe, G., Holt, B., Arnoud, J. ve diğ., 2020, *Forest of Trees-Based Ensemble Methods*, https://github.com/scikit-learn/scikit-learn/blob/0fb307bf3/sklearn/ensemble/_forest.py#L883, [Ziyaret Tarihi: 25.10.2020].

Louppe, G., Prettenhofer, P., Holt., B. ve diğ., 2020, *Tree-Based Methods*, https://github.com/scikit-learn/scikit-learn/blob/0fb307bf3/sklearn/tree/_classes.py#L597, [Ziyaret Tarihi: 25.10.2020].

MacQueen, J., 1967, Some Methods for Classification and Analysis of Multivariate Observations, *In Proceedings of the Fifth Berkeley Symposium On Mathematical Statistics and Probability* 1 (14), 281-297.

McKinney, W., 2010, Data Structures for Statistical Computing in Python, *Proceedings of the 9th Python in Science Conference*, 445, 51-56.

MEB, 2004, *Millî Eğitim Bakanlığı Orta Öğretim Kurumları Sınıf Geçme ve Sınav Yönetmeliği*, T.C. Resmi Gazete, Sayı: 25664.

Mitchell, T.M., 1997, *Machine Learning*, 1. baskı, McGraw-Hill Science/Engineering/Math, ISBN: 0-07-042807-7.

Michel, V., Pedregosa, F., Aides, A. ve diğ., 2020, *Naive Bayes Algorithms*, https://github.com/scikit-learn/scikit-learn/blob/0fb307bf3/sklearn/naive_bayes.py#L669, [Ziyaret Tarihi: 25.10.2020].

Microsoft Corporation, 2018, Microsoft Excel, https://office.microsoft.com/excel, [Ziyaret Tarihi: 25.10.2020].

Minaei-Bidgoli, B., Kashy, D.A., Kortemeyer, G. ve diğ., 2003, Predicting Student Performance: An Application of Data Mining Methods with an Educational Web-Based System, *In 33rd Annual Frontiers in Education*, 1, T2A-13, IEEE.

Mohri, M., Rostamizadeh, A. ve Talwalkar, A., 2012, *Foundations of Machine Learning*, The MIT Press, ISBN: 0-262-01825-X.

Müller, A.C. ve Guido, S., 2016, *Introduction to Machine Learning with Python: A Guide for Data Scientists*, O'Reilly Media, Inc, ABD, ISBN: 978-1-449-36941-5.

Nandy, A. ve Biswas, M., 2017, *Reinforcement Learning: With Open AI, TensorFlow and Keras Using Python*, Apress.

Narsky, I. ve Porter, F.C., 2013, *Statistical Analysis Techniques in Particle Physics : Fits, Density Estimation and Supervised Learning*, ProQuest Ebook Central, https://ebookcentral.proquest.com, ISBN: 978-3-5276-7731-3.

Naser, S.A., Zaqout, I., Ghosh, M.A. ve diğ., 2015, Predicting Student Performance Using Artificial Neural Network: in the Faculty of Engineering and Information Technology, *International Journal of Hybrid Information Technology*, 8 (2), 221-228.

Nilsson, N.J., 1998, *Artificial Intelligence: A New Synthesis*, Morgan Kaufmann Publishers, ABD, ISBN: 1-55860-467-7.

Nobari, A.G., 2020, *Short Term Load Forecasting by Using Artificial Neural Networks*, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi.

Nordman, A., 2011, Data Mining Lecture 6: Evaluating the Performance of a Model, http://staffwww.itn.liu.se/~aidvi/courses/06/dm/lectures/lec6.pdf, [Ziyaret Tarihi: 13.09.2020].

Noriega, L., 2005, Multilayer Perceptron Tutorial, *School of Computing, Staffordshire University*.

Ogor, E.N., 2007, Student Academic Performance Monitoring and Evaluation Using Data Mining Techniques, *Electronics, Robotics and Automotive Mechanics Conference (CERMA)*, 354-359, IEEE.

Olson, D.L., 2017, Recency Frequency and Monetary Model, *Descriptive Data Mining*, 43-59, Springer, Singapore.

Olson, D.L. ve Delen, D., 2008, *Advanced Data Mining Techniques*, Springer Verlag, Berlin Heidelberg, ISBN: 978-3-540-76917-0.

Olson, D.L. ve Wu, D., 2017, *Predictive Data Mining Models*, Springer, Singapore, ISBN: 978-981-13-9664-9.

Özdemir, Ş., 2016, *Eğitimde Veri Madenciliği ve Öğrenci Akademik Başarı Öngörüsüne İlişkin Bir Uygulama*, Doktora Tezi, İstanbul Üniversitesi.

Özkan, Y., 2013, *Veri Madenciliği Yöntemleri*, 2. baskı, Papatya Yayıncılık Eğitim, ISBN: 0-201-74128-8.

Özkan, Y. ve Selçukcan Erol, Ç., 2017, Biyoenformatik DNA Mikrodizi Veri Madenciliği, 2. baskı, Papatya Yayıncılık Eğitim, ISBN: 978-605-4220-89-2.

Öztemel, E., 2012*, Yapay Sinir Ağları*, Papatya Yayıncılık Eğitim, İstanbul, ISBN: 975-978-6797-39-6.

Pal, M., 2005, Random Forest Classifier for Remote Sensing Classification, *International Journal of Remote Sensing*, 26 (1), 217-222.

Paper, D., 2018, *Data Science Fundamentals for Python and MongoDB*, Apress Media, ISBN: 978-1-4842-3597-3.

Patel, A.A., 2019, *Hands-On Unsupervised Learning Using Python: How to Build Applied Machine Learning Solutions from Unlabeled Data*, O'Reilly Media.

Patterson, D.W., 1990, *Introduction to Artificial Intelligence and Expert Systems*, Prentice-hall of India.

Pedregosa, F., Varoquaux, G., Gramfort, A. ve diğ., 2011, Scikit-learn: Machine learning in Python, *The Journal of Machine Learning Research*, 12, 2825-2830.

Pettigrew, E.J., 2009, *A Study of the Impact of Socioeconomic Status on Student Achievement in a Rural East Tennessee School System*, Electronic Theses and Dissertations, 1844, East Tennessee State University.

Polyzou, A. ve Karypis, G., 2019, Feature Extraction for Next-Term Prediction of Poor Student Performance, *IEEE Transactions on Learning Technologies*, 12 (2), 237-248.

Pyle, D., 1999, *Data Preparation for Data Mining*, Morgan Kaufmann Publishers, ABD, ISBN: 1-55860-529-0.

Rahm, E. ve Do, H.H., 2000, Data Cleaning: Problems and Current Approaches, *IEEE Data Eng. Bull*., 23(4), 3-13.

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

Quinlan, J.R., 1993, *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers, San Mateo, ABD.

Quinlan, J.R., 1996, Bagging, boosting, and C4.5, *In Aaai/iaai*, 1, 725-730.

Ramaswami, M. ve Bhaskaran, R. 2009, A Study on Feature Selection Techniques in Educational Data Mining, *Journal of Computing*, 1 (1).

Rogers, S. ve Girolami, M., 2011, *A First Course in Machine Learning*, CRC Press, ISBN: 978-1-4398-2414-6.

Roiger, R. ve Geatz, M., 2003, *Data Mining: A Tutorial Based Primer*, Addison Wesley, ABD, ISBN: 0- 201-74128-8.

Roiger, R.J., 2016, *Data Mining : A Tutorial-Based Primer*, 2.baskı, ProQuest Ebook Central https://ebookcentral.proquest.com, ISBN: 978-1-4987-6398-1.

Romero, C. ve Ventura, S., 2007, Educational Data Mining: A Survey from 1995 to 2005, *Expert Systems with Applications*, 33 (1), 135-146.

Rosenblatt, F., 1958, The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain, *Psychoanalytic Review*, 65, 386-408.

Rumelhart, D.E., Hinton, D.E. ve Williams, R.J., 1986, Learning Representation by Backpropagating Errors, *Nature*, 323 (9), 533-536.

Rumelhart, D.E. ve McClelland, J.L., 1986, Parallel Distributed Processing, Explorations in the Microstructure of Cognition, *Foundations*, MIT Press Cambridge, MA, 1.

Saharidis, G.K.D., Androulakis, I.P. ve Ierapetritou, M.G., 2011, Model Building Using Bi-Level Optimization, *Journal of Global Optimization*, 49 (1), DOI: 10.1007/s10898-010-9533-9, 49–67.

Salal, Y.K., Abdullaev, S.M. ve Kumar, M., 2019, Educational Data Mining: Student Performance Prediction in Academic, *IJ of Engineering and Advanced Tech*, 8 (4C), 54-59.

SAS Institute Inc, 2017, Introduction to SEMMA, https://documentation.sas.com/?docsetId=emref&docsetTarget=n061bzurmej4j3n1jnj8bbjjm1a2.htm&docsetVersion=14.3&locale=en, [Ziyaret Tarihi: 13.01.2021].

Shalev-Shwartz, S. ve Ben-David, S., 2014, *Understanding Machine Learning: From Theory to Algorithms*, Cambridge University Press.

Shannon, C.E., 1948, A Mathematical Theory of Communication, *The Bell System Technical Journal*, 27 (3), 379-423.

Shearer, C., 2000, The CRISP-DM Model: The New Blueprint for Data Mining, *Journal Of Data Warehousing*, 5 (4), 13-22.

Sirin, S.R., 2005, Socioeconomic Status and Academic Achievement: A Meta-Analytic Review of Research, *Review of Educational Research*, 75 (3), 417-453.

Sokolova, M. ve Lapalme, G., 2009, A Systematic Analysis of Performance Measures for Classification Tasks, *Information Processing & Management*, 45 (4), 427–437.

Sutton, R.S. ve Barto, A.G., 1998, *Introduction to Reinforcement Learning*, 135, MIT press, Cambridge, İngiltere.

Şen, B., Uçar, E. ve Delen, D., 2012, Predicting and Analyzing Secondary Education Placement-Test Scores: A Data Mining Approach, *Expert Systems with Applications*, 39 (10), 9468-9476.

Şengür, D., 2013, *Öğrencilerin Akademik Başarılarının Veri Madenciliği Metotları ile Tahmini*, Yüksek Lisans Tezi, Fırat Üniversitesi, Eğitim Bilimleri Enstitüsü, Bilgisayar ve Öğretim Teknolojileri Eğitimi, Elazığ.

Teixeira, T., Treuille, A., Conkling, T. ve diğ., 2021, Streamlit, 0.74.1., https://github.com/streamlit/streamlit, [Ziyaret Tarihi: 13.01.2021].

Tomasevic, N., Gvozdenovic, N. ve Vranes, S., 2020, An Overview and Comparison of Supervised Data Mining Techniques for Student Exam Performance Prediction, *Computers & Education*, 143, 103676.

Tomul, E. ve Savasci, H.S., 2012, Socioeconomic Determinants of Academic Achievement, *Educational Assessment, Evaluation and Accountability*, 24 (3), 175-187.

Turing, A.M., 1948, Intelligent Machinery.

Ulutürk Akman, S., 2004, Tüketicilerin Fiyat Bilinci Üzerinde Etkili Olan Faktörlere İlişkin Bir İnceleme, *Maliye Araştırma Merkezi Konferansları*, (46), 129.

Uzel, V.N., Turgut, S.S. ve Özel, S.A., 2018, Prediction of Students' Academic Success Using Data Mining Methods, *Innovations in Intelligent Systems and Applications Conference (ASYU)*, 1-5, IEEE.

VanderPlas, J., 2016, *Python Data Science Handbook: Essential Tools for Working with Data*, O'Reilly Media, Inc., ISBN: 978-1-491-91205-8.

Vanderplas, J., Pedregosa, F., Gramfort, A. ve diğ., 2020, *Nearest Neighbor Classification*, https://github.com/scikit-learn/scikit-learn/blob/0fb307bf3/sklearn/neighbors/_classification.py#L26, [Ziyaret Tarihi: 25.10.2020].

Van Rossum, G. ve Drake, F.L., 2009, Python 3 Reference Manual, Scotts Valley, CA: CreateSpace.

Van Rossum, G., Peters, T., Vassolotti, A. ve diğ., 2020, Create portable serialized representations of Python objects, https://github.com/python/cpython/blob/master/Lib/pickle.py, [Ziyaret Tarihi: 18.12.2020].

Vapnik, V.N., 1999, An Overview of Statistical Learning Theory, *IEEE Transactions On Neural Networks*, 10 (5), 988-999.

Varoquaux, G., Pedregosa, F., Gramfort, A. ve diğ., 2020, *Logistic Regression*, https://github.com/scikit-learn/scikit-learn/blob/0fb307bf3/sklearn/linear_model/_logistic.py#L1011, [Ziyaret Tarihi: 25.10.2020].

Waskom, M. ve diğ., 2017. mwaskom/seaborn: v0.8.1, Zenodo, https://doi.org/10.5281/zenodo.883859, [Ziyaret Tarihi: 25.10.2020].

Yegnanarayana, B., 2009, *Artificial Neural Networks*, PHI Learning Pvt. Ltd.

Youssef, M., Mohammed, S. ve Wafaa, B.F., 2019, A Predictive Approach Based on Efficient Feature Selection and Learning Algorithms' Competition: Case Of Learners' Dropout in MOOCs, *Education and Information Technologies*, 24 (6), 3591-3618.

Zafari, A., Kianmehr, M.H ve Abdolahzadeh, R., 2013, Modeling the Effect of Extrusion Parameters on Density of Biomass Pellet Using Artificial Neural Network, *International Journal of Recycling of Organic Waste in Agriculture*, 2 (1), 9.

Zhang, H., Weng, T.W., Chen, P.Y. ve diğ., 2018, Efficient Neural Network Robustness Certification with General Activation Functions, *In Advances in Neural Information Processing Systems*, 4939-4948.

Zhang, S., Zhang, C. ve Yang, Q., 2003, Data Preparation for Data Mining, *Applied Artificial Intelligence*, 17 (5-6), 375-381.

Zhu, X. ve Wu, X., 2004, Class Noise vs. Attribute Noise: A Quantitative Study, *Artificial Intelligence Review*, 22 (3), 177-210.

Zweig, M.H. ve Campbell, G., 1993, Receiver-Operating Characteristic (ROC) Plots: A Fundamental Evaluation Tool in Clinical Medicine, *Clinical Chemistry*, 39 (4), 561-577.

EKLER

EK 1. MEB İzin Dilekçesi.

T.C.
İSTANBUL VALİLİĞİ
İl Millî Eğitim Müdürlüğü

Sayı : 59090411-44-E.23945912
Konu : Anket Araştırma İzni

03.12.2019

İSTANBUL ÜNİVERSİTESİ REKTÖRLÜĞÜ'NE

İlgi: a) 11.11.2019 tarihli ve 261 sayılı yazınız.
b) Valilik Makamının 02.12.2019 tarihli ve 23773071 sayılı oluru.

Üniversiteniz Eğitim Bilimleri Enstitüsü yüksek lisans öğrencisi Suat ŞAHİN'in **"Makine Öğrenmesi Yöntemleri İle Ortaokul Öğrencisi Başarılarının Tespiti Ve Bir Uygulama"** konulu araştırma çalışması hakkındaki ilgi (a) yazınız ilgi (b) valilik onayı ile uygun görülmüştür.

Bilgilerinizi ve araştırmacının söz konusu talebi; bilimsel amaç dışında kullanmaması, **uygulama sırasında bir örneği müdürlüğümüzde muhafaza edilen mühürlü ve imzalı veri toplama araçlarının kurumlarımıza araştırmacı tarafından ulaştırılarak uygulanılması,** katılımcıların gönüllülük esasına göre seçilmesi, araştırma sonuç raporunun müdürlüğümüzden izin alınmadan kamuoyuyla paylaşılmaması koşuluyla, gerekli duyurunun araştırmacı tarafından yapılması, okul idarecilerinin denetim, gözetim ve sorumluluğunda, eğitim-öğretimi aksatmayacak şekilde ilgi (b) Valilik Onayı doğrultusunda uygulanması ve işlem bittikten sonra 2 (iki) hafta içinde sonuçtan Müdürlüğümüz Strateji Geliştirme Bölümüne rapor halinde bilgi verilmesini arz ederim.

Timur TUĞRAL
İl Milli Eğitim Müdürü a.
Şube Müdürü

EK:
1- Valilik Onayı
2- Ölçekler

Milli Eğitim Müdürlüğü Binbirdirek M. İmran Öktem Cad.
No:1 Eski Adliye Binası Sultanahmet Fatih/İstanbul
E-Posta: sgb34@meb.gov.tr

Bilgi İçin Aydın. BALTA VHKİ
Tel: (0 212) 384 34 00- 3628

Bu evrak güvenli elektronik imza ile imzalanmıştır. https://evraksorgu.meb.gov.tr adresinden 12f2-fa27-31a8-b2c3-5578 kodu ile teyit edilebilir.

EK 2. Model Geliştirme – Python Kodları.

```
# GEREKLİ KÜTÜPHANELERİN YÜKLENMESİ ###########
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder,OneHotEncoder
from sklearn.preprocessing import StandardScaler,MinMaxScaler
from sklearn.metrics import mean_squared_error as mse
from sklearn.metrics import mean_absolute_error as mae
from sklearn.neighbors import KNeighborsClassifier
from sklearn.neighbors import KNeighborsRegressor

# VERİ SETİNİN DATAFRAME'E AKTARILMASI VE VERİ SETİ HAKKINDA ÖZET BİLGİLER
df=pd.read_excel('tez-veri-birlesik-clasf.xlsx')
print('Eksik Değerler Listesi:\n',df.isnull().sum())
print(df.info())
aciklama=df.describe()
print(aciklama)
index=[-1,-2,-6,-7,-8]
renk=['#4682b4', '#ff1493','gray','#3cb371','orange']

for i in range(0,5):
    sns.boxplot(x=df.iloc[:,index[i]],color=renk[i])
    plt.show()

#AYKIRI DEĞER TESPİTİ
q1=df['SON ORT'].quantile(0.25)
q3=df['SON ORT'].quantile(0.75)
IQR=q3-q1
alt_sinir=q1-1.5*IQR
ust_sinir=q3+1.5*IQR
alt_uc_degerler=(df['SON ORT']<(alt_sinir))
ust_uc_degerler=(df['SON ORT']>ust_sinir)
print("alt sınır değerler: ", alt_sinir)
print("aykırı değer sayısı: ", df['SON ORT'][alt_uc_degerler].shape[0])
print("aykırı değerler: \n", df['SON ORT'][alt_uc_degerler])
df['SON ORT'][alt_uc_degerler]=45.39

sns.boxplot(x=df.iloc[:,-1],color='#4682b4')
plt.show()

#100'LÜK SİSTEMDEKİ NOTLARIN 5'LİK SİSTEME ÇEVRİLMESİ
for j in range(0,328):
    if(df['SON ORT'][j]<45):
        df['SON ORT'][j]=1
    elif(df['SON ORT'][j]>=45 and df['SON ORT'][j]<55):
        df['SON ORT'][j]=2
    elif(df['SON ORT'][j]>=55 and df['SON ORT'][j]<70):
        df['SON ORT'][j]=3
    elif(df['SON ORT'][j]>=70 and df['SON ORT'][j]<85):
        df['SON ORT'][j]=4
    else:
        df['SON ORT'][j]=5

liste=[x for x in df.columns ]
```

```
print(liste)

sns.countplot(x=df['BABA MESL'])
plt.xticks(rotation=90)
plt.show()

#HEDEF NİTELİĞİN DAĞILIM GRAFİĞİ ################
print(df.iloc[:,-1].value_counts())
ax=sns.countplot(x=df.iloc[:,-1],data=df, )
for p in ax.patches:
    ax.annotate('{:.2f}'.format(p.get_height()), (p.get_x()+0.15,
p.get_height()+1))
plt.figure(figsize=(12,8))
plt.show()

sns.distplot(df['SON ORT'])
plt.show()


#ÖZNİTELİKLERİN DAĞILIM GRAFİKLERİ #########
for i in range(0,25):
    ax=sns.countplot(x=df.iloc[:,i],data=df)
    for p in ax.patches:
        ax.annotate('{:.2f}'.format(p.get_height()), (p.get_x()+0.15,
p.get_height()+1))
    plt.xticks(rotation=90)
    plt.show()


#EKSİK VERİLERİN TAMAMLANMASI ##############################
from sklearn.impute import SimpleImputer,KNNImputer
from sklearn.experimental import enable_iterative_imputer
from sklearn.impute import IterativeImputer

imp= SimpleImputer(missing_values=np.nan,strategy='mean')
imp2= SimpleImputer(missing_values=np.nan,strategy='most_frequent')

df.iloc[:,10:11]=imp.fit_transform(df.iloc[:,10:11])
df.iloc[:,17:18]=imp.fit_transform(df.iloc[:,17:18])
df.iloc[:,10:11]=df.iloc[:,10:11].round()
df.iloc[:,17:18]=df.iloc[:,17:18].round()
df.iloc[:,18:19]=imp.fit_transform(df.iloc[:,18:19])
df.iloc[:,19:20]=imp.fit_transform(df.iloc[:,19:20])
df.iloc[:,23:24]=imp.fit_transform(df.iloc[:,23:24])
df.iloc[:,24:25]=imp.fit_transform(df.iloc[:,24:25])


df.iloc[:,0:10]=imp2.fit_transform(df.iloc[:,0:10])
df.iloc[:,11:17]=imp2.fit_transform(df.iloc[:,11:17])
df.iloc[:,20:23]=imp2.fit_transform(df.iloc[:,20:23])
 ###########################################################

cinsiyet=df.iloc[:,0:1].values
annesag=df.iloc[:,1:2].values
babasag=df.iloc[:,2:3].values
abbirlikte=df.iloc[:,3:4].values
anneegt=df.iloc[:,4:5].values
babaegt=df.iloc[:,5:6].values
annehastalik=df.iloc[:,6:7].values
```

```
babahastalik=df.iloc[:,7:8].values
annemeslek=df.iloc[:,8:9].values
babameslek=df.iloc[:,9:10].values
kardessayisi=df.iloc[:,10:11].values
gelir=df.iloc[:,11:12].values
kiminleyasiyor=df.iloc[:,12:13].values
odavarmi=df.iloc[:,13:14].values
evkirami=df.iloc[:,14:15].values
isinma=df.iloc[:,15:16].values
sureklihastalik=df.iloc[:,16:17].values
devamsizlik=df.iloc[:,17:18].values
turkce=df.iloc[:,18:19].values
matematik=df.iloc[:,19:20].values
surekliilac=df.iloc[:,20:21].values
ameliyat=df.iloc[:,21:22].values
burslumu=df.iloc[:,22:23].values
gecmis_ort=df.iloc[:,23:24].values
son_ort=df.iloc[:,24:25].values

#KATEGORİK VERİLERİN SAYISAL HALE DÖNÜŞTÜRÜLMESİ ##############
le=LabelEncoder()
cinsiyet=le.fit_transform(cinsiyet)
annesag=le.fit_transform(annesag)
babasag=le.fit_transform(babasag)
abbirlikte=le.fit_transform(abbirlikte)
anneegt=le.fit_transform(anneegt)
babaegt=le.fit_transform(babaegt)
annehastalik=le.fit_transform(annehastalik)
babahastalik=le.fit_transform(babahastalik)
annemeslek=le.fit_transform(annemeslek)
babameslek=le.fit_transform(babameslek)
gelir=le.fit_transform(gelir)
kiminleyasiyor=le.fit_transform(kiminleyasiyor)
odavarmi=le.fit_transform(odavarmi)
evkirami=le.fit_transform(evkirami)
isinma=le.fit_transform(isinma)
sureklihastalik=le.fit_transform(sureklihastalik)
surekliilac=le.fit_transform(surekliilac)
ameliyat=le.fit_transform(ameliyat)
burslumu=le.fit_transform(burslumu)
#########################################################

#SÜTUNLARI TEKRAR DATAFRAME'E CEVİRME ###############
d1=pd.DataFrame(data=cinsiyet,columns=['cinsiyet'])
d2=pd.DataFrame(data=annesag,columns=['annesag'])
d3=pd.DataFrame(data=babasag,columns=['babasag'])
d4=pd.DataFrame(data=abbirlikte,columns=['abbirlikte'])
d5=pd.DataFrame(data=anneegt,columns=['anneegt'])
d6=pd.DataFrame(data=babaegt,columns=['babaegt'])
d7=pd.DataFrame(data=annehastalik,columns=['annehastalik'])
d8=pd.DataFrame(data=babahastalik,columns=['babahastalik'])
d9=pd.DataFrame(data=annemeslek,columns=['annemeslek'])
d10=pd.DataFrame(data=babameslek,columns=['babameslek'])
d11=pd.DataFrame(data=gelir,columns=['gelir'])
d12=pd.DataFrame(data=kiminleyasiyor,columns=['kiminleyasiyor'])
d13=pd.DataFrame(data=odavarmi,columns=['odavarmi'])
d14=pd.DataFrame(data=evkirami,columns=['evkirami'])
d15=pd.DataFrame(data=isinma,columns=['isinma'])
d16=pd.DataFrame(data=sureklihastalik,columns=['sureklihastalik'])
```

```
d17=pd.DataFrame(data=devamsizlik,columns=['devamsizlik'])
d18=pd.DataFrame(data=turkce,columns=['turkce'])
d19=pd.DataFrame(data=matematik,columns=['matematik'])
d20=pd.DataFrame(data=surekliilac,columns=['surekliilac'])
d21=pd.DataFrame(data=ameliyat,columns=['ameliyat'])
d22=pd.DataFrame(data=burslumu,columns=['burslumu'])
d23=pd.DataFrame(data=kardessayisi,columns=['kardessayisi'])
d24=pd.DataFrame(data=gecmis_ort,columns=['gecmis_ort'])
d25=pd.DataFrame(data=son_ort,columns=['son_ort'])
############################################################

#DATAFRAME'LERİ BİRLEŞTİRME ##############################
df1=pd.concat([d1,d2] , axis=1)
df2=pd.concat([df1,d3] , axis=1)
df3=pd.concat([df2,d4] , axis=1)
df4=pd.concat([df3,d5] , axis=1)
df5=pd.concat([df4,d6] , axis=1)
df6=pd.concat([df5,d7] , axis=1)
df7=pd.concat([df6,d8] , axis=1)
df8=pd.concat([df7,d9] , axis=1)
df9=pd.concat([df8,d10] , axis=1)
df10=pd.concat([df9,d11] , axis=1)
df11=pd.concat([df10,d12] , axis=1)
df12=pd.concat([df11,d13] , axis=1)
df13=pd.concat([df12,d14] , axis=1)
df14=pd.concat([df13,d15] , axis=1)
df15=pd.concat([df14,d16] , axis=1)
df16=pd.concat([df15,d17] , axis=1)
df17=pd.concat([df16,d18] , axis=1)
df18=pd.concat([df17,d19] , axis=1)
df19=pd.concat([df18,d20] , axis=1)
df20=pd.concat([df19,d21] , axis=1)
df21=pd.concat([df20,d22] , axis=1)
df22=pd.concat([df21,d23] , axis=1)
df23=pd.concat([df22,d24] , axis=1)
df24=pd.concat([df23,d25] , axis=1)
############################################################

x=df24.iloc[:,0:24]
y=df24.iloc[:,24:25]

y=y.astype('int')
liste=[x for x in df24.columns ]
print(liste)

# ÖZELLİK SEÇİMİ ##########################################
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2
from sklearn.feature_selection import f_regression
from sklearn.feature_selection import mutual_info_regression
from sklearn.feature_selection import mutual_info_classif
from sklearn.feature_selection import f_classif

bestf=SelectKBest(score_func=chi2,k=11)
fit=bestf.fit(x,y)
dfscores=pd.DataFrame(fit.scores_)
dfcolumns=pd.DataFrame(x.columns)
ozellik_skoru=pd.concat([dfcolumns,dfscores],axis=1)
ozellik_skoru.columns=['Specs','Score']
```

```
print(ozellik_skoru.nlargest(11,'Score'))
############################################################

import warnings
warnings.filterwarnings('ignore')

from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.metrics import f1_score
from sklearn.metrics import r2_score
from sklearn.metrics import confusion_matrix


x=df24.iloc[:,[3,4,8,9,11,12,16,17,18,22,23]]
y=df24.iloc[:,24:25]
y=y.astype('float')


#ISI HARİTASI İLE ÖZNİTELİKLER ARASI KORELASYONUN GÖRÜNTÜLENMESİ
corr=df24.iloc[:,[16,17,18,22,23,24]].corr()
plt.figure(figsize=(8,6))
sns.heatmap(corr,annot=True,cmap=plt.cm.Reds)
print(corr)
plt.show()
#########################################
"""m=[x for x in range(1,len(y)+1)]
plt.plot(m,y)
plt.show()
"""
sns.distplot(y,kde=False)
plt.show()

x_train,x_test,y_train,y_test=train_test_split(x,y,
test_size=0.33,random_state=0)

#YSA İLE TAHMİN
from sklearn.neural_network import MLPClassifier
model=MLPClassifier(hidden_layer_sizes=(11,22,5 ), alpha=1e-05,
activation='tanh', solver='adam', random_state=42, max_iter=300,
learning_rate='adaptive')
model.fit(x_train,y_train)
tahminysa=model.predict(x_test)
oran7=accuracy_score(y_test,tahminysa)
print('Ysa Tahmin Doğruluk Oranı:',oran7)

cm=confusion_matrix(y_test, tahminysa)
print('Yapay Sinir Ağları Karmaşıklık Matrisi:\n',cm)

#KARAR AĞAÇLARI SINFLANDIRMA ALGORİTMASI İLE TAHMİN
from sklearn.tree import DecisionTreeClassifier
""" #PARAMETRE SEÇİMİ
for i in range(1,30):
    dtc=DecisionTreeClassifier(max_depth=i,min_samples_leaf=1)
    dtc.fit(x_train,y_train)
    tahmindtc=dtc.predict(x_test)
    oran1=accuracy_score(y_test,tahmindtc)
    print(i,'parametresi için Karar Ağaçları Doğruluk Oranı:',oran1)
"""
dtc=DecisionTreeClassifier(max_depth=3)
```

```
dtc.fit(x_train,y_train)
tahmindtc=dtc.predict(x_test)
oran1=accuracy_score(y_test,tahmindtc)
print('Karar Ağaçları Tahmin Doruluk Oranı:',oran1)
cm=confusion_matrix(y_test, tahmindtc)
print('Karar Ağaçları Karmaşıklık Matrisi:\n',cm)

##########################################

#RANDOM FOREST ALGORİTMASI İLE TAHMİN
SEED=1
from sklearn.ensemble import RandomForestClassifier
""" #PARAMETRE SEÇİMİ
for i in range(10,200,10):
    rfc=RandomForestClassifier(n_estimators=i, min_samples_leaf=1,
random_state=42)
    rfc.fit(x_train,y_train)
    tahminrfc=rfc.predict(x_test)
    oran2=accuracy_score(y_test,tahminrfc, normalize=True)
    print(i,'parametresi için Random Forest Doğruluk Oranı:',oran2)
"""
rfc=RandomForestClassifier(n_estimators=20, min_samples_leaf=1,
random_state=42)
rfc.fit(x_train,y_train)
tahminrfc=rfc.predict(x_test)
oran2=accuracy_score(y_test,tahminrfc, normalize=True)

print('Random Forest Doğruluk Oranı:',oran2)
cm=confusion_matrix(y_test, tahminrfc)
print('Random Forest Karmaşıklık Matrisi:\n',cm)

##########################################

#AĞAÇ YAPISI GÖRSELLEŞTİRME ####################

print('random forest karar ağacı sayısı: ',len(rfc.estimators_))
from sklearn import tree
fig, axes = plt.subplots(nrows = 1,ncols = 1,figsize = (4,4), dpi=800)
dot_data=tree.plot_tree(rfc.estimators_[0],

feature_names=x.columns,class_names=['2','3','4','5'],
                        filled=True)
fig.savefig('agac_yapi.png')

##########################################

#DESTEK VEKTÖR MAKİNESİ İLE TAHMİN
from sklearn.svm import SVC
svc_cls=SVC(kernel='linear')
svc_cls.fit(x_train,y_train)
tahminsvc=svc_cls.predict(x_test)
oran3=accuracy_score(y_test,tahminsvc)
print('Destek Vektör Regresyon Doğruluk Oranı:',oran3)
cm=confusion_matrix(y_test, tahminsvc)
print('Destek Vektör Makinesi Karmaşıklık Matrisi:\n',cm)

##########################################A

#KNN SINIFLANDIRMA ALGORİTMASI İLE TAHMİN
```

```
from sklearn.neighbors import KNeighborsClassifier
""" #PARAMETRE SEÇİMİ
for i in range(1,10):
    knn=KNeighborsClassifier(n_neighbors=i, metric='manhattan')
    knn.fit(x_train,y_train)
    tahminknn=knn.predict(x_test)
    oran4=accuracy_score(y_test,tahminknn)
    print(i,'parametresi için Knn Doğruluk Oranı:',oran4)
"""
knn=KNeighborsClassifier(n_neighbors=7, metric='manhattan')
knn.fit(x_train,y_train)
tahminknn=knn.predict(x_test)
oran4=accuracy_score(y_test,tahminknn)
print('Knn Doğruluk Oranı:',oran4)

cm=confusion_matrix(y_test, tahminknn)
print('Knn Karmaşıklık Matrisi:\n',cm)

from sklearn.preprocessing import label_binarize
from sklearn.metrics import roc_curve, auc
from sklearn.multiclass import OneVsRestClassifier
from sklearn.model_selection import cross_val_predict

##########################################

#NAIVE BAYES SINFLANDIRMA ALGORİTMASI İLE TAHMİN
from sklearn.naive_bayes import MultinomialNB
nb= MultinomialNB()
nb.fit(x_train,y_train)
tahminnb=nb.predict(x_test)

oran5=accuracy_score(y_test,tahminnb)
print('Naive Bayes Doğruluk Oranı:',oran5)
cm=confusion_matrix(y_test, tahminnb)
print('Naive Bayes Karmaşıklık Matrisi:\n',cm)

##########################################

#LOJİSTİK REGRESYON ALGORİTMASI İLE TAHMİN
from sklearn.linear_model import LogisticRegression
logr= LogisticRegression(random_state=None, solver='lbfgs', max_iter=500,
multi_class='multinomial')
logr.fit(x_train,y_train)
tahminlogr=logr.predict(x_test)

oran6=accuracy_score(y_test,tahminlogr)
print('Lojistik Regresyon Doğruluk Oranı:',oran6)
cm=confusion_matrix(y_test, tahminlogr)
print('Lojistik Regresyon Karmaşıklık Matrisi:\n',cm)

#########################################################

#K-KAT ÇAPRAZ GEÇERLEME ##########################
kf=10
from sklearn.model_selection import cross_val_score
basari=cross_val_score(estimator=knn,X=x_train,y=y_train,cv=kf,scoring='acc
uracy')
print('Knn k-fold değeri:\n',basari)
print('Knn k-fold ortalama değer:\n',basari.mean())
```

```
basari=cross_val_score(estimator=dtc,X=x_train,y=y_train,cv=kf,scoring='acc
uracy')
print('Karar Ağaçları k-fold değeri:\n',basari)
print('Karar Ağaçları k-fold ortalama değer:\n',basari.mean())

basari=cross_val_score(estimator=rfc,X=x_train,y=y_train,cv=kf,scoring='acc
uracy')
print('Random Forest k-fold değeri:\n',basari)
print('Random Forest k-fold ortalama değer:\n',basari.mean())

basari=cross_val_score(estimator=svc_cls,X=x_train,y=y_train,cv=kf,scoring=
'accuracy')
print('DEstek Vektör Makinesi k-fold değeri:\n',basari)
print('DEstek Vektör Makinesi k-fold ortalama değer:\n',basari.mean())

basari=cross_val_score(estimator=nb,X=x_train,y=y_train,cv=kf,scoring='accu
racy')
print('Naive Bayes k-fold değeri:\n',basari)
print('Naive Bayes k-fold ortalama değer:\n',basari.mean())

basari=cross_val_score(estimator=logr,X=x_train,y=y_train,cv=kf,scoring='ac
curacy')
print('Lojistik Regresyon k-fold değeri:\n',basari)
print('Lojistik Regresyon k-fold ortalama değer:\n',basari.mean())

basari=cross_val_score(estimator=model,X=x_train,y=y_train,cv=kf,scoring='a
ccuracy')
print('Yapay Sinir Ağları k-fold değeri:\n',basari)
print('Yapay Sinir Ağları k-fold ortalama değer:\n',basari.mean())

##########################################################
from sklearn.metrics import classification_report
#SINIFLANDIRMA RAPORLARI ############################
cp_knn=classification_report(y_test,tahminknn)
cp_dtc=classification_report(y_test,tahmindtc)
cp_rfc=classification_report(y_test,tahminrfc)
cp_svc=classification_report(y_test,tahminsvc)
cp_logr=classification_report(y_test,tahminlogr)
cp_nb=classification_report(y_test,tahminnb)
cp_ysa=classification_report(y_test,tahminysa)

print('knn:\n',cp_knn)
print('dtc:\n',cp_dtc)
print('rfc:\n',cp_rfc)
print('svc:\n',cp_svc)
print('logr:\n',cp_logr)
print('nb:\n',cp_nb)
print('ysa:\n',cp_ysa)

##########################################################
from sklearn.preprocessing import LabelBinarizer
from sklearn.metrics import roc_auc_score

#ROC-AUC SKORU HESAPLAMAK İÇİN GEREKLİ FONKSİYON
def multiclass_roc_auc_score(y_test, tahminknn, average="macro"):
    lb = LabelBinarizer()
    lb.fit(y_test)
    y_test = lb.transform(y_test)
```

```
    tahminknn = lb.transform(tahminknn)
    return roc_auc_score(y_test, tahminknn, average=average)

#MODELLERE AİT ROC-AUC SKORU HESAPLAMASI
roc_skoru=multiclass_roc_auc_score(y_test, tahminknn)
print('knn roc değeri:',roc_skoru)

roc_skoru=multiclass_roc_auc_score(y_test, tahmindtc)
print('dtc roc değeri:',roc_skoru)

roc_skoru=multiclass_roc_auc_score(y_test, tahminrfc)
print('rfc roc değeri:',roc_skoru)

roc_skoru=multiclass_roc_auc_score(y_test, tahminsvc)
print('svc roc değeri:',roc_skoru)

roc_skoru=multiclass_roc_auc_score(y_test, tahminysa)
print('ysa roc değeri:',roc_skoru)

roc_skoru=multiclass_roc_auc_score(y_test, tahminlogr)
print('logr roc değeri:',roc_skoru)

roc_skoru=multiclass_roc_auc_score(y_test, tahminnb)
print('nb roc değeri:',roc_skoru)
#RASTGELE ORMAN MODELİNİN KAYDEDİLMESİ###
import pickle
dosya='tez_rfc_model.pkl'
pickle.dump(tahminrfc,open(dosya,'wb'))
```

EK 3. Modelin Uygulamaya Geçirilmesi – Python Kodları.

```
import streamlit as st
import pandas as pd
import numpy as np
import joblib

def main():
    st.title("ÖĞRENCİ AKADEMİK BAŞARI TAHMİN SİSTEMİ")

if __name__ == '__main__':
    main()


abbirlikte_dict = {'evet':0,'hayır':1}
anneegt_dict = {'okuma-yazma
yok':3,'ilkokul':0,'ortaokul':4,'lise':2,'lisans':1}
annemeslek_dict = {'özel sektörde işçi':6,'özel sektörde
çalışan':7,'serbest
meslek':4,'memur':3,'emekli':1,'esnaf':2,'diğer':0,'çalışmıyor':5}
babameslek_dict = {'özel sektörde işçi':11,'özel sektörde
çalışan':12,'serbest meslek':9,'memur':8,'içişleri
bak.':5,'MSB':0,'emekli':3,'esnaf':4,'kamu kurumunda işçi':6,'kamu
kurumunda sözleşmeli':7,'bağkur mensubu':1,'diğer':2,'çalışmıyor':10}
kiminleyasiyor_dict = {'ailesiyle':0,'annesi ile':1,'babası
ile':2,'yurt':3}
odavarmi_dict = {'evet':0,'hayır':1}



abbirlikte = st.selectbox("Anne baba birlikte
mi",tuple(abbirlikte_dict.keys()))
anneegt = st.selectbox("Anne Eğitim Durumu", tuple(anneegt_dict.keys()))
annemeslek = st.selectbox("Anne Meslek", tuple(annemeslek_dict.keys()))
babameslek = st.selectbox("Baba Meslek",tuple(babameslek_dict.keys()))
kiminleyasiyor = st.selectbox("Kiminle
Yaşıyor",tuple(kiminleyasiyor_dict.keys()))
odavarmi = st.selectbox("Kendine Ait Odası Var
mı?",tuple(odavarmi_dict.keys()))
devamsizlik = st.number_input("Devamsızlık",0,20)
turkce = st.number_input("İlk Dönem Türkçe Not Ortalaması",0.00,100.00)
matematik = st.number_input("İlk Dönem Matematik Not
Ortalaması",0.00,100.00)
kardessayisi = st.number_input("Kardeş Sayısı",0,20)
gecmis_ort=st.number_input("Geçmiş Yıllar Ağırlıklı Not
Ortalaması",0.00,100.00)

abbirlikte=abbirlikte_dict.get(abbirlikte)
anneegt=anneegt_dict.get(anneegt)
annemeslek=annemeslek_dict.get(annemeslek)
babameslek=babameslek_dict.get(babameslek)
kiminleyasiyor=kiminleyasiyor_dict.get(kiminleyasiyor)
odavarmi=odavarmi_dict.get(odavarmi)

st.write(abbirlikte,' ',anneegt,' ',annemeslek,' ',babameslek,'
',kiminleyasiyor,' ',odavarmi)
```

```
res = pd.DataFrame(data =

{'abbirlikte':[abbirlikte],'anneegt':[anneegt],'annemeslek':[annemeslek],

'babameslek':[babameslek],'kiminleyasiyor':[kiminleyasiyor],'odavarmi':[oda
varmi],

'kardessayisi':[kardessayisi],'devamsizlik':[devamsizlik],'turkce':[turkce]
,
          'matematik':[matematik],
          'gecmis_ort':[gecmis_ort]})

import pickle
with open('tez_rfc_model_2.pkl', 'rb') as f:
    rfc = pickle.load(f)

prediction = rfc.predict(res)
prediction = str(prediction).strip('[]')

if st.button('Tahmin'):
    st.write("Random Forest Modeli Tahmini: ",prediction)
```

ÖZGEÇMİŞ

Kişisel Bilgiler	
Adı Soyadı	Suat ŞAHİN
Doğum Yeri	
Doğum Tarihi	
Uyruğu	☑ T.C. ❒ Diğer:
Telefon	
E-Posta Adresi	
Web Adresi	

Eğitim Bilgileri	
Lisans	
Üniversite	Çanakkale 18 Mart Üniversitesi
Fakülte	Eğitim Fakültesi
Bölümü	Bilgisayar Ve Öğretim Teknolojileri Eğitimi
Mezuniyet Yılı	Tarih girmek için tıklayın veya dokunun.

Yüksek Lisans	
Üniversite	İstanbul Üniversitesi
Enstitü Adı	Fen Bilimleri Enstitüsü
Anabilim Dalı	Enformatik Anabilim Dalı
Programı	Enformatik

Makale ve Bildiriler