T.C.

YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

VERİ ANALİZİ KÜMELEME VE SINIFLANDIRMA YÖNTEMLERİNDE MATEMATİKSEL MODELLEMELER VE İYİLEŞTİRME ÖNERİLERİ

Melike Göksu

YÜKSEK LİSANS TEZİ

Matematik Anabilim Dalı

Matematik Programı

Danışman

Doç. Dr. Filiz KANBAY

Haziran, 2022

T.C.

YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

VERİ ANALİZİ KÜMELEME VE SINIFLANDIRMA YÖNTEMLERİNDE MATEMATİKSEL MODELLEMELER VE İYİLEŞTİRME ÖNERİLERİ

Melike GÖKSU tarafından hazırlanan tez çalışması 22.06.2022 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Matematik Anabilim Dalı, Matematik Programı **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Doç. Dr. Filiz KANBAY
Yıldız Teknik Üniversitesi
Danışman

Jüri Üyeleri

Doç. Dr. Filiz KANBAY, Danışman
Yıldız Teknik Üniversitesi _____

Dr. Öğr. Üyesi Mutlu AKAR, Üye
Yıldız Teknik Üniversitesi _____

Dr. Öğr. Üyesi Mehmet Fatih UÇAR, Üye
İstanbul Kültür Üniversitesi _____

Danışmanım Doç. Dr. Filiz KANBAY sorumluluğunda tarafımca hazırlanan Veri Analizi Kümeleme Ve Sınıflandırma Yöntemlerinde Matematiksel Modellemeler Ve İyileştirme Önerileri başlıklı çalışmada veri toplama ve veri kullanımında gerekli yasal izinleri aldığımı, diğer kaynaklardan aldığım bilgileri ana metin ve referanslarda eksiksiz gösterdiğimi, araştırma verilerine ve sonuçlarına ilişkin çarpıtma ve/veya sahtecilik yapmadığımı, çalışmam süresince bilimsel araştırma ve etik ilkelerine uygun davrandığımı beyan ederim. Beyanımın aksinin ispatı halinde her türlü yasal sonucu kabul ederim.

Melike GÖKSU

İmza

Biricik anneanneme

ve

dedeme

TEŞEKKÜR

Bütün çalışma sürecimde değerli bilgilerini benimle paylaşan, kendisine ne zaman danışsam bana kıymetli zamanını ayırıp sabırla ve büyük bir ilgiyle bana faydalı olabilmek için elinden geleni yapan güler yüzünü ve samimiyetini benden esirgemeyen danışman hocam Doç. Dr. Filiz KANBAY'a, teşekkürlerimi sunarım. Değerli bilgi ve önerileri ile çalışmama destek sağlayan Yıldız Teknik Üniversitesi Bilgisayar Mühendisliği Bölümü öğretim üyesi Prof. Dr. Songül VARLI'ya teşekkür ederim. Ayrıca lisansüstü eğitimim boyunca bilgi ve tecrübeleriyle beni yönlendiren, her zaman destek olup cesaretlendiren hocam Arş. Gör. Eda YILDIZ'a teşekkür ederim. Hayatımın her aşamasında sevgi ve şefkatini üzerimden eksik etmeyen, desteklerini hiçbir zaman esirgemeyen bu hayattaki en büyük şansım olan aileme sonsuz teşekkür ederim.

Melike GÖKSU

İÇİNDEKİLER

SİMGE LİSTESİ

$P(X)$	Bir X olayının olma olasılığı
$P(X\|\ Y)$	Bir Y olayı gerçekleştiğinde X olayının gerçekleşme olasılığı
k	Küme sayısı
m_i	Kümenin merkez noktası
n	Nesne sayısı
K	Veri kümesinin temsili

KISALTMA LİSTESİ

ADABOOST	Adaptive Boosting
CART	Classification and Regression Trees
CHAID	Chi-Squared Automatic Interaction Detector
CRISP-DM	Cross Industry Standard Process for Data Mining
FN	False Negative
FP	False Positive
ID3	Iteratif Dichotomiser 3
K-NN	K Nearest Neighbor Algorithm
MARS	Multivariate Adaptive Regression Splines
NCR	National Cash Register
SLIQ	Supervised Learning in Quest
SPRINT	Scalable Parallelizable Induction of Decision Trees
SPSS	Statistical Package for the Social Sciences
TN	True Negative
TP	True Negavite
WCSS	Within Clusters Sum of Square
XGBOOST	Extreme Gradient Boosting

ŞEKİL LİSTESİ

TABLO LİSTESİ

ÖZET

VERİ ANALİZİ KÜMELEME VE SINIFLANDIRMA YÖNTEMLERİNDE MATEMATİKSEL MODELLEMELER VE İYİLEŞTİRME ÖNERİLERİ

Melike GÖKSU

Matematik Anabilim Dalı

Yüksek Lisans Tezi

Danışman: Doç. Dr. Filiz KANBAY

Bu tez çalışmasının birinci bölümü tezin amacı ve ilgili literatür özetini içermektedir. İkinci bölümü veri madenciliğinin tanımı, kullanım alanları ve veri madenciliği süreçleri ile ilgili temel kavramlara ayrılmıştır. Üçüncü bölümde ise çeşitli kümeleme ve sınıflandırma algoritmalarının matematiksel temelleri açıklanarak model değerlendirme ölçütlerinden bahsedilmiştir. Tezin dördüncü bölümünde 403 adet bilgi içeren bir veri seti ele alınarak ilk üç bölümde konu edilen kümeleme ve sınıflandırma algoritmaları yani en yakın komşu algoritması, k-ortalama değer algoritması, naive bayes algoritması, karar ağacı algoritması, destek vektör makineleri, boosting (XGboost, AdaBoost) ve rastgele orman algoritması uygulanarak ilgili matematiksel modellemeler oluşturulmuştur. İlgili modellemelerin başarısını arttıracak model parametreleri ızgara taraması ile en uygun hale getirilmiştir. Son olarak sonuçların görselliğini sağlamak amacı ile öznitelikler rastgele orman algoritması ile önem sırasına göre sıralanmış ve öznitelik sayıları bu sıralamaya göre azaltılarak veri görselleştirilmesi sağlanmıştır.

Anahtar Kelimeler: Veri madenciliği, makine öğrenmesi, en yakın komşu algoritması, rastgele orman algoritması, destek vektör makineleri.

ABSTRACT

Mathematical Modelling and Improvement Suggestions in Clustering and Classification Methods for Data Analysis

Melike Göksu

The Department of Mathematics

Master's Thesis

Supervisor: Associate Professor Filiz KANBAY

The first chapter of this thesis contains a summary of the relevant literature as well as the purpose of the thesis. The second chapter defines data mining, its application areas, and the fundamental concepts associated with data mining processes. In the third chapter, the mathematical foundations of various clustering and classification algorithms are explained and model evaluation criteria are mentioned. In the fourth chapter of the thesis, a data set which is containing 403 pieces of information was examined. This data set was used to create related mathematical models using the clustering and classification algorithms discussed in the first three chapters such as k nearest-neighbour, k-mean value, naive bayes, decision tree, support vector machines, boosting (XGboost, AdaBoost), and random forest algorithm. The model parameters that will increase the success of the related modeling have been optimized by grid scanning. Finally, in order to provide visualization of the results, the features were ranked in order of importance using the random forest and the number of features was reduced based on this order and data visualization provided.

Keywords: Data Mining, machine learning, k-nearest algoritm, random forest algorithm, support vector machine.

1
GİRİŞ

1.1 Literatür Özeti

Veri, geçmişten günümüze önemini hiç yitirmeden ulaşmış bir unsur olup; elde tutulması, korunması tarihin farklı zamanlarında çeşitli yöntemler ile sağlanmıştır. Sümerlerin kullandığı taş tabletlerden Mısırlıların kullandığı papirüse, oradan kağıda ve günümüzde dijital ortama aktarılan verinin muhafaza edilmesi için o çağın mevcut teknolojisinden yararlanılarak farklı sistemler geliştirilmiştir. Bu ihtiyaçlar doğrultusunda veri madenciliğinin temelleri, 1763 yılında Thomas Bayes'in Bayes Teoremi ve Adrien Marie Legendre ve Carl Gauss'un değişkenler arasındaki ilişkinin belirlenmesini sağlayan regresyon analizi modelinin temeli olan en küçük kareler yönteminin açıklanması ile atılmıştır [1]. İstatistikteki bu gelişmeleri takiben Alan Turing, 1936 yılında daha sonra günümüz bilgisayarlarının geliştirilmesinde kullanılacak hesaplamaları işleme yeteneğine sahip olacak "Evrensel Makine" fikrini ortaya koymuştur [2]. 1943 yılında ise Warren McCullah ve Walter Pitts bir sinir ağının ilk modelini oluşturarak nöronların girdi almak, girdileri işlemek ve çıktı üretmek olmak üzere 3 ana görevinin olduğundan bahsetmişlerdir [3]. Ardından 1952'de Arthur Samuel "makine öğrenimi" tabirini ilk defa kullanmıştır [4]. 1960'larda ve 1970'lerde, veri kümelerinin boyutu büyümüş ve terabaytlarca verinin depolanmasına ihtiyaç duyulmuştur. İlk derin öğrenme ağı 1965 yılında aralarında Ivakhnenko ve Lapa'nın bulunduğu araştırmacılar tarafından oluşturulmuştur [5]. John Henry Holland 1975'te yayınladığı Doğal ve Yapay Sistemlerde Adaptasyon isimli kitabında genetik algoritmaların ana hatlarını çizmiştir [6]. Bu algoritmalar, 1980'lerin sonlarında geliştirilen bildiğimiz şekliyle veri madenciliğinin başlangıcının temelini oluşturmaktadır.

1990'ların başında, veri madenciliği, veri hazırlama (depolama, veri temizleme, ön işleme verileri vb.) ve sonuçların analizini, görselleştirilmesini içeren çeşitli süreçlerle ilgili olan geniş bir veri yelpazesiyle gizli bilgilerin keşfi olarak yani Veritabanlarında Bilgi Keşfi olarak daha geniş bir süreç içinde bir alt süreç olarak kabul edildi[7]. 2000'li

yıllarda veri bilimi ve bilgisayar bilimi birleşerek istatistiksel ve algoritmik modelleme çalışmaları yapılmaya başlandı ve yeni algoritmalar geliştirilerek günümüze kadar ulaştı. Örneğin günümüzde en sık kullanılan algoritmalardan olan k -En Yakın Komşu (kNN) algoritması, ilk olarak 1950'li yıllarda kendinden söz ettirmesine rağmen büyük veri setleri için işlem maliyeti zaman aldığı için bilgi işleme hızı artana kadar yeterince etkinleşememiştir [8].

En çok kullanılan sınıflandırma algoritmalarından biri olan rastgele orman algoritması Bagging yöntemi ve Random Subspace yöntemlerinin birleşimi ile oluşmuş olup 2001 yılında Leo Breiman tarafından geliştirilmiştir [9].

Mantıksal temeli 18.yy'da T. Bayes tarafından atılmış olan Naive Bayes, bir diğer sınıflandırma algoritmasıdır.

1931'de Frank P. Ramsey bir karar metodolojisini geliştiren ilk kişi olmuştur. 1954 yılında John von Neumann ve Oscar Morgenstern tarafından metodoloji geliştirilmiş ve resmileştirilmiştir. Algoritma, 1960'lı yıllarda Neumann ve Wald'ın katkıları ile artık Karar Teorisi olarak bilinmeye başlandı [10].

Bir istatistiksel sınıflandırma meta-algoritması olan AdaBoost (Adaptive Boosting), Yoav Freund ve Robert Schapire tarafından oluşturulmuştur [11,12]. XGBoost ise (Extreme Gradient Boosting) 2016'da çalışılmıştır [13].

Destek vektör makineleri algoritmasının temelleri Vladimir N. Vapnik ve Alexey Ya tarafından 1963 yılında atılmış sonrasında 1995 yılında Corinna Cortes ve Vapnik tarafından önerilmiştir ve yayımlanmıştır [14, 15]. Veri madenciliğinin sadeliği nedeniyle en yaygın kullanılan algoritmalardan biri olan K -Ortalama Algoritması 1967 senesinde J.B. MacQueen tarafından geliştirilmiştir [16].

1.2 Tezin Amacı

Veri madenciliği kümeleme ve sınıflandırma yöntemlerinde matematiksel modellemeler yaparak bu modellemeye ait en uygun parametrelerin belirlenmesi ile model doğruluk ölçütlerinde iyileştirme sağlanması amaçlanmaktadır.

1.2 Hipotez

Veri madenciliği kümeleme ve sınıflandırma yöntemleri için kullanılan k- en yakın komşu, naive bayes, rastgele orman, destek vektör makineleri, karar ağacı, boosting, k-

ortalamalar algoritmalarına ait model parametrelerinin model doğruluk parametreleri ile ilişkisi incelenerek model doğruluk ölçütlerini arttıracak yeni matematiksel modeller araştırılacaktır.

2
VERİ MADENCİLİĞİ SÜRECİ

Veri madenciliği süreçleri konusunda 1996 yılında, SPSS ve NCR'ı temsil eden bir analist grubu yoğun çalışmalar yapmış ve bu çalışmanın sonucu olarak CRISP- DM ismini verdikleri bir veri madenciliği süreci geliştirmişlerdir. Bir aşamanın sonucu, diğer bir aşamanın girdisi olduğundan veri madenciliği sürecinde her aşama dikkatle izlenmelidir. Veri madenciliği sürecini açıklayan CRISP-DM, 6 aşamadan oluşmaktadır. Bu sebeple veri madenciliği, altı alt başlıkta incelenebilir [17].

2.1 Problemin Tanımlanması

Veri madenciliği sürecinin önemli aşamalarından olan bu bölüm, beklenti ve amaçların ortaya konulduğu kısımdır. Hedefler ve ihtiyaçlar bu bölümde veri madenciliği tanımına dönüştürülür ve sonrasında da değerlendirilir. Böylece problem için bir ön plan oluşturulur. Lakin bu bölüm verinin analizini içermez.

2.2 Verinin Anlaşılması

Veriyi anlama aşaması olan ikinci aşama, ilk olarak verilerin toplanma süreci, daha sonra kalite sorunlarının değerlendirilmesi, verilerin tanımlanması ve ilk bilgilerin ortaya konulmasını hedefler. Böylece yeni bilgiler için hipotezler ortaya koymayı sağlar.

2.3 Verinin Modellenmeye Hazır Hale Getirilmesi

2.3.1 Veri Temizleme

Bu aşamada eksik veriler doldurulur veya kaldırılır, gürültülü veriler düzeltilir, aykırı veriler temizlenir ve uyuşmazlıklar çözümlenir. Kayıp verilerin yaratacağı problemleri ortadan kaldırmak için kayıp verinin bulunduğu kayıt veri kümesinden çıkartılabilir,

kayıp veriler teker teker doldurulabilir, kayıp verilere sabit bir değer atanabilir, kayıp veriler yerine ortalama değer girilebilir veya eksik olmayan veriler kullanılarak Regresyon, Karar Ağaçları, Maksimum beklenti, Doğrusal İnterpolasyon, Jackknife gibi yöntemler ile kayıp veriler tahmin edilebilir.

Gürültülü verilerin temizlenmesi veya düzgünleştirilmesi için ise paketleme, regresyon, kümeleme, temel bileşenler analizi kullanılabilir.

2.3.2 Veri Bütünleştirme

Veri birleştirme, farklı kaynaklardan (veritabanları, veri küpleri, metin dosyaları vb.) alınan veriler arasında uyum oluşturarak bu verilerin bir tek veri kümesi altında birleştirilmesi işlemidir**.**

2.3.3 Veri Dönüştürme

Bazen ele alınan verinin, veri madenciliği sürecine doğrudan katılması mümkün olmayabilir. Bu durumda modellemede kullanılacak algoritmanın seçimi ve algoritmanın işleyişine uygun veri dönüşümü gerekir. Değişkenlerin birimlerinden arındırılması, gürültü etkilerinin azaltılması, kategorik değişkenlerin nümerik değişkenlere dönüştürülmesi gibi durumlar söz konusu olduğunda Min- Max Normalleştirmesi veya Z- Skor Standartlaştırma dönüşüm yöntemlerinden biri seçilerek bu söz konusu değişkenlerin normalleştirilmesi veya standartlaştırılması işlemi yapılır.

2.3.4 Veri İndirgeme

Çok büyük veri kümeleri ile yapılan veri madenciliği uygulamalarında analiz işlemi uzun sürebileceğinden bu gibi durumlarda verinin esas özelliklerini kaybetmeden veri sayısı ya da değişken sayısı azaltılabilir. Böylece elde edilen daha küçük hacimli veri kümesinde veri madenciliği teknikleri uygulanarak daha etkili sonuçlar elde edilebilir. Bu amaçla verinin boyutunu indirgeme, sıkıştırma veya ayrıklaştırma gibi yöntemler kullanılabilir.

2.4 Modelleme

Bu adım, çeşitli modelleme tekniklerinin ve algoritmaların seçimi, uygun parametrelerin belirlenmesi, test edilmesi, yeni model geliştirilmesi ve tahmin işlemlerini içermektedir.

Veri madenciliği, her problem için farklı yöntemler sunmaktadır. Bir veri madenciliği probleminde uygun olan teknik veya tekniklerin bulunabilmesi işlemi, birçok teknik oluşturarak bunların içinden en uygun olanları seçmek şeklinde düşünülebilir. Veri Madenciliği problemleri için kullanılacak olan modelleri genelde tanımlayıcı (descriptive) ve tahmin edici (predictive) olmak üzere iki farklı başlık altında ele alabiliriz.

Tahmin edici modellerde amaç, sınıfları (değerleri) belirli olan veriler üzerinde bir model oluşturarak, bu model yardımıyla sınıfları (değerleri) bilinmeyen veri kümelerinin sınıflarının (değerlerinin) tahmin edilmesini sağlamaktır. Sınıflama (classification), regresyon ve kestirim (prediction) madenciliği tahmin edici tekniklerin başlıcalarıdır.

Tanımlayıcı modellerin amacı ise mevcut verileri tanımlayarak modelin doğru karar vermesini sağlamaktır. Kümeleme (clustering), birliktelik kuralı (association rule) ve ardışıl örüntü (sequential pattern) madenciliği tanımlayıcı tekniklerden bazılarıdır.

2.5 Modelin Değerlendirilmesi

Model bu aşamada, hedefler göz önüne alınarak uygulamaya koyulmadan önce son kez her yönden değerlendirilir. Gerekli ise iyileştirilmeler yapılır. Modelin, proje hedeflerinin gerçeğine ne derecede uygun olduğu sorgulanır.

2.6 Modelin Uygulanması

Sürecin son aşaması olan model uygulaması, kurulan ve geçerliliği kabul edilen modelin kullanıldığı adımdır.

3

MAKİNE ÖĞRENMESİ KÜMELEME SINIFLANDIRMA ALGORİTMALARI VE MODEL DEĞERLENDİRME ÖLÇÜTLERİ

Makine öğrenme algoritmaları öğrenme biçimi bakımından genel olarak iki ana başlıkta incelenir. Bu öğrenme biçimlerinden ilki olan denetimli öğrenme, bir modelin etiketli bir veri kümesi üzerinde eğitildiği, yani hem giriş hem de çıkış parametrelerine sahip bir öğrenme türüdür. Veri seti uzman kişi tarafından etiketlenmiştir. Buradaki işleyiş uzmanı tarafından etiketlendirilmiş bir veri seti üzerinde bir model kurup, bu model aracılığı ile sınıf etiketine sahip olmayan bir verinin etiketlendirilmesi üzerinedir. Problemi çözmek için eğitim verisi adı verilen örnekler kullanılır. Algoritma bu örnekleri kullanarak bir tahmin modeli üretir ve yeni girdi verildiğinde bu modeli kullanarak sınıf etiketini belirlemeye yardımcı olur. Mevcut bilginin kategorik olması, iyi tanımlı olması önemlidir.

Denetimli makine öğrenimi yöntemi, bilinen veriler için yaklaşık bir tahmin sonucu oluşturmamız gerektiğinde kullanışlı olup sınıflandırma ve regresyon problemleri için uygulanır. Sınıflandırma algoritmaları, verinin çıktıları çeşitli sınıflardan meydana geldiğinde; regresyon algoritmaları ise, verinin çıktıları sayısal değerlere sahip olduğunda kullanılır.

Başlıca denetimli öğrenme algoritmaları aşağıda belirtilmiştir.

- K- En Yakın Komşu Algoritması
- Naive Bayes
- Karar Ağaçları
- Regresyon
- Rastgele Orman
- Gradyan Güçlendirilmiş Ağaçlar

Diğer bir öğrenme biçimi olan denetimsiz öğrenme algoritmaları veri kümesindeki etiketsiz verileri işleyerek, bu verilerin benzer özelliklerini tespit ederek bunları belirleyip kümelemeyi amaçlar. Denetimsiz öğrenme, verilerdeki bilinmeyen her türlü özniteliği inceler ve verilerin kendi içindeki ilişkilerine bakar. Verilere herhangi bir etiket verilmediğinden denetimsiz öğrenme algoritmaları, denetimli öğrenme algoritmalarına göre az karışık işlem görevleri içeridiğinden büyük veri kümelerinde kullanılmak üzere daha elverişlidir.

Bu öğrenme algoritmaları, sadece etiketsiz veri yığınını alarak, verileri ortak özellikleri kullanarak gruplayıp kümeler. Her yeni veri kümesi için daha önce tespit ettiği ortak özellikleri bu verilerde test ederek karar verir.

3.1 Kümeleme ve Sınıflandırma Algoritmaları

3.1.1 K- En Yakın Komşu Algoritması

3.1.1.1 Klasik K- En Yakın Komşu Algoritması

KNN modellemesi, ele alınan veri için yakınlık ilişkisi kullanarak sınıflandırma yapan denetimli öğrneme algoritmalarından biridir. Diğer denetimli öğrenme algoritmalarına benzemeyen sınıflandırma ve regresyon için kullanılan basit bir algoritmadır, eğitim aşamasına sahip değildir.

K en yakın komşu algoritmasının çıktısı verinin komşularının çoğunluk oyuyla oluşturulmuş sınıf üyeliğidir. Veri, böylece komşularının çoğunluk oyuyla ilgili sınıfa atanır.

Öncelikle verilen bir noktaya en yakın komşulukların sayısı bir k parametresi olarak belirlenir. Eğer k sayısı küçük olarak alınırsa tahmin bölgesi sınırlandırılmış olur. Böylece sınıflandırıcı daha az hassas hale getirilir. Burada amaç verilen bir noktaya en yakın k adet komşulukları belirlemek olduğu için söz konusu olan nokta ile diğer tüm noktalar arasındaki uzaklıklar hesaplanır. Kullanılan uzaklıklar Öklid, Manhattan, Chebyshev, Hamming mesafesi ve Kosinüs benzerliği olarak sıralanabilir. Hesaplanan bu uzaklıklar küçükten büyüğe sıralanır ve en küçük uzaklığa sahip olan k adeti seçilerek en çok tekrar eden sınıf etiketi belirlenir. Belirlenenmiş olan sınıf etiketi, gözlem değerinin sınıf etiketi olmuş olur.

Algoritmanın etkinliğini arttırmak için veriler yeniden ölçeklendirilebilir, farklı mesafe ölçümleri kullanılabilir veya PCA gibi boyut azaltma teknikleri uygulanabilir.

K- en yakın komşu algoritmasında yaygın olarak kullanılan bazı uzaklık ölçütleri aşağıdaki şekildedir:

a) Öklid Mesafesi

Uzaklık hesaplamada en çok kullanılan ölçü, Öklid uzaklığıdır. Öklid uzaklığı fonksiyonu, iki nokta arasında doğrusal uzaklık olup herhangi iki nokta

$P = (x_1, x_2, \dots, x_n)$ ve $Q = (y_1, y_2, \dots, y_n)$ olmak üzere denklem 3.1'de sunulmuştur [18]:

$$dist_{euclidean} = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{3.1}$$

b) Manhattan Mesafesi

Manhattan uzaklığı, n boyutlu iki nokta arasındaki farkın mutlak değerlerinin toplamıdır. Herhangi iki nokta, $P = (x_1, x_2, \dots, x_n)$ ve $Q = (y_1, y_2, \dots, y_n) \in R^n$ olacak şekilde P ve Q arasındaki Manhattan uzaklığı denklem 3.2'de sunulmuştur [18]:

$$dist_{manhattan} = \sum_{i=1}^{n}|x_i - y_i| \tag{3.2}$$

c) Minkowski Mesafesi

Minkowski mesafesinin Öklid ve Manhatten mesafesinin genelleştirilmiş şekli olduğunu söylenebilir. Minkowski uzaklığı, Öklid uzayında tanımlı bir dizidir. n boyutlu uzayda herhangi iki nokta $P = (x_1, x_2, \dots, x_n)$ ve $Q = (y_1, y_2, \dots, y_n) \in R^n$ olacak şekilde Minkowski uzaklığı denklem 3.3'te belirtiliği şeklildedir [18].

$$dist_{minkowski} = \left(\sum_{i=1}^{n}|x_i - y_i|^p\right)^{1/p} \tag{3.3}$$

d) Hamming Mesafesi

Çoğunlukla metin işleme işleri gibi kategorik değişkenler söz konusu olduğunda, kullanılan bir mesafedir. Veri kümesinde sayısal ve kategorik değişkenler bir arada kullanıldığında ise 0 ile 1 arasındaki sayısal değişkenlerin standardizasyonunu

gerçekleştirir. Dolayısıyla, Hamming mesafesi, ikili diziler olarak da adlandırılan iki ikili vektör arasındaki mesafeyi hesaplar.

e) Chebyschev Mesafesi

Minkowski uzaklığının, $n \to \infty$ için hesaplanan özel durumu olan Chebyschev uzaklığı (maksimum değer uzaklığı), iki nokta arasındaki farkların mutlak değerlerinin maksimumu olarak ifade edilir. $P = (x_1, x_2, \dots, x_n)$ ve $Q = (y_1, y_2, \dots, y_n)$ herhangi iki nokta olmak üzere, P ve Q arasındaki Chebyschev uzaklığı,

$$dist_{chebyschev} = \lim_{p\to\infty} (\sum_{i=1}^{n} |x_i - y_i|^p)^{1/p} \tag{3.4}$$

olarak ifade edilir [19].

f) Dilca Mesafesi

Kategorik öznitelik değerleri arasındaki mesafeyi ölçümlemek amacı ile kullanılan Dilca metriği çift aşamalı bir ölçüttür [20]. Bu ölçüt kullanılırken ilk olarak simetrik belirsizlik katsayısı yöntemi kullanmak suretiyle öznitelik seçimi yapılır ve eş-oluşum tablosu oluşturulur. Ardından bu tablo üzerinde koşullu olasılık ve Öklid uzaklığına dayalı hesaplamalar gerçekleştirilir ve uzaklık ölçümlendirilir. Bu mesafe X ve Y öznitelikleri için

$$SU = 2 \text{ x } \left[\frac{HG}{H(Y)+H(X)}\right] \tag{3.5}$$

olarak verilen simetrik belirsizlik katsayısı kullanılarak hesaplanır [21,22].

Özniteliklerin seçimi sonrası uzaklık

$$d(x_i, x_j) = \sqrt{\sum_{Y\in baglam(X)} \sum_{y_k\in Y} (P(x_i|y_k) - P(x_j|y_k))^2} \tag{3.6}$$

ile hesaplanır [20].

3.1.1.2 Bulanık K-Ortalama Algoritması (Fuzzy- KNN)

Bulanık-kNN algoritması, Bulanık küme metodlarının k-NN algoritması ile birleştirilmesiyle gelişen bir algoritma olup klasik K-NN algoritmasına göre daha küçük hata oranına sahiptir. Bu nedenle bulanık-kNN algoritması, standart kNN algoritması üzerinde bir gelişme olarak kabul edilebilir.

Bulanık k-NN algoritmasını klasik k-NN algoritmasından ayıran en önemli husus ait olduğu sınıfı bilinmeyen herhangi bir y giriş vektörünü belirli bir sınıfa dahil etmek yerine bu vektöre sınıf üyeliği atıyor olmasıdır. Bu sebeple üyelik değerlerinin, sınıflamada bir güven ölçümü yerine geçtiği söylenebilir. Bu da algoritmanın avantajlarından biridir.Bu yöntem uygulanırken, eğitim seti ile sınıf üyeliklerini önceden belirlenir. Daha sonra, test kümesinin her bir örneği için k-NN uygulanır. Bulanık-kNN algoritması için biçimsel gösterim aşağıdaki şekilde özetlenebilir:

TR bir eğitim veri kümesi ve TS bir test kümesi olsun, bunlar sırasıyla n ve t sayıda örneklerden oluşsun. Her örnek x_i şeklinde bir vektördür (x_{i1}, x_{i2} ,…, x_{im}), burada $x_{i,j}$ i-inci örneğinin j-inci özelliğinin değerini ifade etmektedir. TR'nin her örneği bilinen bir ω sınıfına aitken, TS için sınıf bilinmemektedir.

Bulanık k-NN'nin iki farklı aşaması vardır: İlk aşama, TR'nin k_{memb} en yakın komşularını, $leave - one - out$ şemasını takip etmeye devam ederek kendisine karşı hesaplar.Bunu yapmak için, x_{train} ile tüm TR örnekleri arasındaki mesafeleri hesaplayarak en yakın k_{memb} örneklerini arar. Komşuları hesapladıktan sonra, Denklem 3.7'de gösterildiği üzere sınıf üyeliğini oluşturur. Böylece TR , özgün sınıf etiketi yerine bir sınıf üyeliği vektörüne sahip olmuş olur [23].

$$u_j(x) = \begin{cases} 0.51 + \left({}^{n_j}/_{k_{memb}}\right).0.49\,, & i = j \\ \left({}^{n_j}/_{k_{memb}}\right).0.49\,, & i \neq j \end{cases} \tag{3.7}$$

İkinci aşama, ilk aşama gibi en yakın komşuları hesaplar, ancak bu durumda, TS'nin her bir örneği için TR'deki en yakın k'yi hesaplar [23].

$$u_i(x) = \frac{\sum_{j=1}^{K} u_{ij}(1/|x-x_j|^{2/(m-1)}}{\sum_{j=1}^{K}(1/|x-x_j|^{2/(m-1)}} \quad (3.8)$$

3.1.2 Rastgele Orman Algoritmasi

Topluluk öğrenmesi (ensemble learning) metodlarından biri olan rastgele orman algoritması, aynı zamanda bir denetimli öğrenme algoritması olup sıklıkla tercih edilen bir algoritmadır. Bu yöntem, çeşitli rastgele karar ağaçlarını bir araya getirip onların ortalamalarını alarak tahminlerini birleştirmektedir [24]. Kurulan bu ormanlar "bagging" yöntemi ile eğitilir. Farklı veri setleri üzerinde eğitim gerçekleştirildiği için karar ağaçlarının en büyük sorunu olan ezbere öğrenme (overfitting) azalır. Rastgele orman modelinin bir diğer kullanım amacı öznitelikler arasındaki önem sırasını belirlemesidir. Bir özniteliğin önemi yüksek ise bağımlı değişkenin varyansının açıklanmasına sağladığı katkı o kadar fazladır.

Rastgele orman algoritması çalışırken öncelikle algoritma, veri kümesi içerisinden rastgele örnekler seçer. Seçilen her örnek için algoritma bir karar ağacı oluşturur ve tahmin sonucunu alır. Eğer problem bir sınıflandırma problemi ise mod; regresyon problemi ise ortalamayı kullanarak sonucu tahmin eder [25]. Son olarak algoritma son tahmin için en çok oylanan tahmin sonucunu seçecektir.

3.1.3 Naive Bayes Algoritmasi

Naive Bayes sınıflandırma algoritması, koşullu olasılık özelliklerine göre temellenmiş bir sınıflandırma yöntemidir. Bu yöntemde etiketli veriler için koşullu olasılık fonksiyonu kullanılarak etiketsiz verilerin sınıflarının belirlenmesi amaçlanır [26]. Bu algoritma nümerik değerli veriler için kullanılabildiği gibi, kategorik veriler üzerinde de kullanılabilir [27].

Temelinde Bayes teoremi yatan ve büyük veri setleri için kullanışlı olan istatistik tabanlı Naive Bayes sınıflandırma algoritmasının uygulanabilmesi için tüm sınıfların istatistiksel olarak birbirinden bağımsız olduğu ve aynı zamanda tüm sınıfların hedef sınıfa eşit derecede katkıda bulunduğu kabul edilir. Algoritmanın adını aldığı teorem olan Naive Bayes Teoremi, olasılıkları hesaplamak için kullanılan ve rastgele seçilen iki

olayın koşullu ve marjinal olasılıklarını ilişkilendiren bir teorem olup iki olayın kesişim olasılıklarının marjinal olasılık değerine bölünmesi anlamına gelen koşullu olasılık teoremine dayanmaktadır [28]. Bu teorem maksimum olabilirlik ilkesini temel alarak mevcut olasılıkların doğruluk oranını hesaplamak için kullanılır.

n örnekten ve m öznitelikten oluşan bir veri setinin k farklı sınıfı $C_1, C_2, \dots, C_k$ olmak üzere sınıf etiketine sahip olmayan bir $X = [x_1, x_2, \dots, x_m]$ örneğinin C_i sınıfına ait olma olasılığı $P(C_i|\,\mathrm{X})$ ile gösterilsin. X örneğinin ait olduğu sınıfı belirlemek amacı ile $P(C_i|\,\mathrm{X})$ olasılığının $i = 1,2, \dots, k$ için maksimum olduğu i değeri kullanılır ve

$$P(C_i|\,\mathrm{X}) = \frac{P(X|\,C_i)\,P(C_i)}{P(X)} \tag{3.9}$$

şeklinde hesaplanır [29].

$$\mathrm{P}(C_i) = \frac{C_i \text{ sınıfına ait örnek sayısı}}{n} \tag{3.10}$$

şeklinde ifade edilir.

Ayrıca, $P(X)$ değeri her sınıflandırma için aynı değeri alacağından göz ardı edilebilirdir. X örneğinin C_i sınıfına ait örnekler arasında bulunma olasılığı denklem 3.11'de ifade edilmiştir [30].

$$P(X|\,C_i) = \prod_{j=1}^{m} P(x_j|\,C_i) \tag{3.11}$$

Bu tanımlara dayanarak Bayes Sınıflandırıcısı;

$$max_i P(C_i|\mathrm{X}) = P(C_k|\mathrm{X}) \tag{3.12}$$

olmak üzere X örneği için i nin maksimum olduğu C_k sınıfını seçer.

İşlem kolaylığı amacıyla paydadaki $P(X)$ değeri her sınıflandırma için aynı olduğundan ihmal edilirse;

$$g_i(x) = P(X|\,C_i)P(\,C_i) \tag{3.13}$$

şeklinde fonksiyon ifade edilebilir [30].

α_i: Herhangi bir X örneğinin C_i sınıfına seçilmesi, $\lambda_{ik} = \alpha_i$ nin gerçekleşmesi ile oluşan kayıp olarak tanımlansın. α_i gerçekleşmesi sonucu beklenen risk fonksiyonu denklem 3.14 ve 3.15'teki şekilde tanımlanır [29,30].

$$R(\alpha_i|x) = \sum_{j=1}^{K} \lambda_{ij}\ P(C_j|\mathrm{x}) \tag{3.14}$$

$$= g_i(x) = -R(\alpha_i|x) \tag{3.15}$$

$$g_t(x) = max_i\ g_i(x) \tag{3.16}$$

olmak üzere X örneği C_t sınıfına ait olarak belirlenir [29,30].

En büyük olasılık tahmini yaparken kullandığımız $g_i(x)$ ifadesindeki

$$P(X|\ C_i\ ) = \prod_{j=1}^{m} P(x_j|\ C_i) \tag{3.17}$$

eşitliği logaritma fonksiyonu kullanılarak toplamsal hale getirilirse denklem 3.18 elde edilir.

$$logP(X|\ C_i\ ) = \sum_{j=1}^{m} log\ P(x_j|\ C_i) \tag{3.18}$$

Bu durumda;

$$g_i(x) = logP(X|\ C_i\ ) + logP(\ C_i\ ) \tag{3.19}$$

şeklinde tanımlanabilir [30].

Bayes kuramı, dağılıma bağlı bir sınıflandırma olduğu için problemin türüne göre çeşitli sınıflandırmalar kullanılabilir. Eğer iki sınıf mevcut ise Bernoulli, daha fazla sınıf mevcut ise Katlı Terimli Dağılım, Gauss Dağılım veya Normal Dağılım modelleme için kullanılabilecek dağılımlardır [30].

Normal dağılım fonskiyonunun matematiksel ifadesi σ_i varyans, μ_i ortalama olmak üzere;

$$P(X) = \frac{1}{\sigma_i\sqrt{2\pi}} e^{-\frac{(x-\mu_i)^2}{2{\sigma_i}^2}} \quad , -\infty < x\ < \infty \tag{3.20}$$

şeklindedir.

3.20 eşitliğinin her iki tarafının logaritması alınırsa;

$$g_i(x) = -\frac{1}{2} log2\pi - log\sigma_i - \frac{(x-\mu_i)^2}{2{\sigma_i}^2} + logP(\ C_i\ ) \tag{3.21}$$

ifadesi elde edilir [31].

Burada $P(X|\ C_i\ )$ ifadesinin sıfır olma durumunu ihmal edebilmek için,

$$P(x_j|\ C_i) = \frac{T_{c_{ix_j}}}{\sum_{j=1}^{m} T_{c_{ix_j}}} \tag{3.22}$$

olmak üzere;

$$P(X|\ C_i) = \frac{T_{c_i}X+1}{\sum_{j=1}^{m}(T_{c_{ix_j}}+1)} = \frac{T_{c_i}X+1}{(\sum_{j=1}^{m} T_{c_i x})+m} \tag{3.23}$$

yazılır [31].

Multinominal modeller için ise bu düzgünleştirme; X örneğine ait x_j özniteliğinin bir C_i kümesinde olma ihtimali $P(x_j|\ C_i)$ ile gösterilsin. Bu ihtimalin sıfir olma durumunu göz ardı edebilmek için

$$\bar{P}(x_j|\ C_i) = \frac{N_{x_j}+\alpha}{N_X+\alpha m} \tag{3.24}$$

şeklinde uygulanır. Burada x örneğine ait x_j özniteliğinin bir C_i kümesinde bulunma sayısı N_{x_j} ,

X örneğine ait tüm özniteliklerin C_i kümesinde bulunma sayısı

$$N_x = \sum_{j=1}^{n} N_{x_j} \tag{3.25}$$

dir [31].

Bu ifadede eğer $\alpha = 1$ alınırsa bu düzgünleştirme işlemi Laplace Düzgünleştirme adını alır.

3.1.4 Karar Ağaçları

Karar ağacı algoritmasının amacı, bir veri kümesine birtakım kurallar uygulaması sonucu grubun tüm elemanlarını aynı sınıf etiketine sahip olana dek küçük gruplara tekrar tekrar bölmektir. Bu algoritma basit karar verme adımları kullanarak büyük veri setlerini küçük parçalara böler ve geliştirir [32].

Anlaşılması ve yorumlanması basit bir algoritma olup, hem sayısal hem de kategorik verilerin işlenmesine olanak sağlayan bir algoritma olduğundan sıklıkla tercih edilen bir algoritmadır. Aynı zamanda hata tahmini (error estimation) ve önem testi (Significance testing – Chi Square Testing) gibi istatistiksel testler yardımıyla modelin doğrulanması

mümkün olup rahatlıkla görselleştirilebilirdir. Bu avantajların yanı sıra aşırı öğrenme (over- fitting) gerçekleşebilir. Bu sorunun çözümü için budama işlemi denilen az sayıda öğe bulunduran yaprakların ve düğümlerin ağaçtan çıkartılması işlemi gerçekleştirilebilir.

Karar ağacı oluşturmak için kullanılan çeşitli algoritmaların en yaygın olanları CHAID, CART, ID3 , Exhaustive CHAID, C4.5, MARS (Multivariate Adaptive Regression Splines), QUEST (Quick, Unbiased, Efficient Statistical Tree), C5.0, SLIQ (Supervised Learning in Quest), SPRINT (Scalable Parallelizable Induction of Decision Trees) dır [32].

3.1.5 Boosting

Boosting algoritması, yavaş öğrenmeye dayalı, sıralı bir yöntemdir. Hatadan öğrenmeyi amaçlayan bu algoritma düşük hassasiyete sahip modelleri birleştirerek güçlü bir tahmin edici model oluşturmayı hedefler. Bu yöntem, her yinelemede elde edilen modelleri, belli kurallar ile birleştirerek daha güçlü bir model elde etmeyi amaçlar [33, 34].

3.1.5.1 XGBoost

Orijinal hali Freidman tarafından geliştirilen XGBoost (Extreme Gradient Boosting) temeli gradient boosting ve karar ağacı algoritmaları olan bir tekniktir. Fazlasıyla yüksek bir tahmin etme gücüne sahip olan bu algoritma genel performansı iyileştirir, aşırı uyum ya da aşırı öğrenmeyi azaltmayı amaçlar. Parametre almadan çalışan bu algoritma kendi içerisinde çapraz doğrulama yaparak modelin doğruluğunun maksimum olmasını sağlar [35].

3.1.5.2 AdaBoost

Fazlasıyla kullanılan boosting yöntemlerinden biri olan AdaBoost (Adaptive Boosting) Yanlış tahminlenmiş gözlem değerlerini yeniden ağırlıklandırmak suretiyle hatayı en aza indirmeyi amaçlar. Burada her bir döngüde sınıflandırıcının ağırlıkları yeniden ayarlanarak veri noktalarının, yeni gözlemleri doğru tahmin edecek şekilde düzenlenmesi sağlanır.

3.1.6 Destek Vektör Makineleri

Denetimli bir algoritma olan Destek vektör makineleri (SVM), sınıflandırma ve regresyon problemleri için kullanılabilmektedir. Bu algoritma, her bir veriyi, n; sahip olduğumuz özelliklerin sayısı olmak üzere, n-boyutlu boşluğa bir nokta olarak işaretler ve belirli bir koordinatın değeri olan her özelliği sınıfı ile birlikte ele alır. Bu algoritmada sınıflandırıcı doğrusal ve doğrusal olmayan sınıflandırma yapabilmektedir. Verilerin türüne bağlı olarak çekirdek fonksiyonlarda kullanılabileceğinden algoritma hem doğrusal hem de işlemlerini gerçekleştirilebilmektedir. Destek vektör makineleri, basit destek vektör makineleri ve çekirdek destek vektör makineleri olmak üzere ikiye ayrılabilir. Eğer veri kümesi ayrıştırılabilir durumda ise tek bir düzlem ile sınıflandırılma sağlanabilir bunun mümkün olmadığı durumlarda çekirdek fonskiyonlar kullanılarak farklı türde ayrıştırılmalar yapılabilmektedir [36].

3.1.7 K- Ortalamalar Algoritması

3.1.7.1 Klasik K- Ortalamalar Algoritması

Bu algoritmanın temel işleyişi n adet etiketsiz veriden oluşan bir veri kümesini, parametre değeri olarak verilen k sayıda özgün kümeye ayırmaktır. Bu ayırma işleminde amaç küme içi benzerliğinin maksimum ve kümeler arası benzerliklerinin minimum olmasıdır. Uygulanabilirliğinin basit olması ve büyük veri kümelerinde hızlı çalışabilmesi sebebiyle sıklıkla tercih edilimesine rağmen başlangıçta k sayısını belirleme zorunluluğu ve gürültülü verilere karşı duyarlı olması gibi dezavantajlara da sahiptir [37].

Rastgele k merkez noktası seçilerek veri kümesindeki elemanlar kendisine en yakın merkez noktanın ait olduğu kümeye atanır, bu atama sonrasında değerlendirme x, C_i kümesinde bulunan bir veri; m_i, C_i kümesinin merkez noktası olmak üzere verilerin bulundukları kümenin merkez noktalarına olan uzaklıklarının karelerinin toplamı olan

$$SSE = \sum_{i=1}^{k} \sum_{x \in C_i} dis^2(m_i, x) \tag{3.26}$$

SSE adı verilen karesel hata kriteri ile hesaplanır [30,38]. Algoritma en düşük SSE değerine sahip kümeleme ile en iyi sonucu bulmayı amaçlar.

3.1.7.2 K- Ortalamalar Algoritmasının Matematiksel İfadesi

n örnekten ve m öznitelikten oluşan sınıf etiketine sahip olmayan bir veri kümesini k farklı sınıfa ayırmak için aşağıdaki adımlar izlenir:

1. k adet nesne seçilir, seçilen bu k nesne küme merkezlerini temsil eder. $M_1, M_2, \dots, M_k$ Örnek orta nokta şu şekilde hesaplanır [39].

$$M_k = \frac{1}{n_k} \sum_{i=1}^{n_k} x_{ik} \tag{3.27}$$

2. Küme içi değişmeler karesel hata formülü kullanılarak denklem 3.28'deki şekilde hesaplanır [40].

$$e_k^2 = \sum_{i=1}^{n_k} (x_{ik} - M_k)^2 \tag{3.28}$$

k kümesini içeren bütün kümeler uzayı için kare-hata, küme içindeki değişimlerin toplamı olup denklem 3.29'da ifade edilmiştir.

$$E_k^2 = \sum_{k=1}^{K} e_k^2 \tag{3.29}$$

3. Merkezin dışında kalan örnekler mesafelerine göre en yakın kümeye atanır.
4. Yapılmış olan atama işleminin sonucunda tekrar k adet küme için merkezler hesaplanır.
5. Kararlı hale gelinene kadar yani küme merkezlerinde değişiklik olmayıncaya kadar 2. ve 3. adımlar tekrarlanır [41].

3.1.7.3 Çekirdek Ağırlıklı K-Ortalama Algoritması

Üzerinde işlem yapılan uzay bakımından k- ortalamalar yönteminden farklıdır. Bu algoritma günümüzde çeşitlenen verilerden çıkarım yapmanın kolaylaşması ve hesaplamaların minimize edilmesi nedeniyle k- ortalamalar metodunun geliştirilmiş lineer olmayan versiyonu olarak tanımlanabilir.

Sınıflar $C_1, C_2, \dots, C_k$ şeklinde gösterilmek üzere bazı standart fonksiyonları;

- Polinom Çekirdek $K(x_i, x_j) = (x_i . x_j + c)^d$
- Gaussian Çekirdek $K(x_i, x_j) = exp\left(-\|x_i - x_j\|^2 / 2\sigma^2\right)$
- Sigmoid Çekirdek $K(x_i, x_j) = tanh(\, c(\, x_i . x_j) + \theta)$
- Doğrusal Çekirdek $K(x_i, x_j) = x_i . x_j$

şeklinde tanımlı olan çekirdek fonksiyonlarından doğrusal olmayan bir çekirdek fonksiyonu kullanılarak elde edilmiş $H = \left[K_{ij}\right]_{nxn}$ matrisi ile ifade edilir; burada K, Veri kümesinin temsili; n, Veri setinin eleman sayısıdır ve

$K: VxV \rightarrow \mathbb{R}$ şeklinde tanımlı bir fonksiyon olup $x_i, x_j \epsilon V$ olmak üzere

$$\phi(x_i).\phi(x_j) = K(x, y) \tag{3.30}$$

ile ifade edilir [42].

V veri setini k adet sınıfa ayırmak için her kümenin ortalaması

$$m_i = \sum_{X_p \epsilon C_i} \frac{\phi(x_p)}{|C_i|} \tag{3.31}$$

ile gösterilmek üzere, minimize edilmek istenen amaç fonksiyonu

$$J = \sum_{i=1}^{k} \sum_{X_j \epsilon C_i} \left\|\phi(x_j) - m_i\right\|^2 \tag{3.32}$$

şeklinde olup,

$$\|\phi(a) - \phi(b)\|^2 = \phi(a)\phi(a) - 2\phi(a)\phi(b) + \phi(b)\phi(b) \tag{3.33}$$

$$= K(a, a) - 2K(a, b) + K(b, b\,) \tag{3.34}$$

dir [42].

Bu ifadeye göre,

$$\left\|\phi(x_j) - m_i\right\|^2 = \left\|\phi(x_j) - \sum_{X_p \epsilon C_i} \frac{\phi(x_p)}{|C_i|}\right\|^2 \quad (3.35)$$

$$= K(x_j, x_j) - \frac{2}{|C_i|}\sum_{x_p \epsilon C_i} K(x_p, x_i) + \frac{1}{|C_i|^2}\sum_{X_p \epsilon C_i}\sum_{X_s \epsilon C_i} K(x_p, x_s) \quad (3.36)$$

Bu durumda işlem algoritması, $\left\|\phi(x_j) - m_i\right\|^2$ ifadesini hesaplayıp ilgili x_j elemanlarını V veri kümesindeki en yakın merkeze ayırmaktır. İşlem, ilgili kümeleme güncellemesi sonucunda tekrar m_i hesabı ile yinelenir [42,43, 44].

3.2 Model Değerlendirme Ölçütleri

Sınıflandırma yöntemleri ile elde edilen modelin başarısını ölçmek ve değerlendirmek amacıyla çeşitli ölçüm yöntemlerine başvurulur. Bu ölçütlerden başlıcaları hata matrisi, doğruluk, duyarlılık kesinlik, f1 skoru, kohen'in kappa skoru şeklindedir.

3.2.1 Hata Matrisi (Confusion Matrix)

Bir sınıflandırma modeli aracılığı ile doğru veya yanlış tahminlenmiş örnek sayısını özetleyen matrise, hata matrisi denir [45]. Amaç hata matrisin esas köşegeni üzerindeki sayıların yüksek, esas köşegen dışındaki eleman sayısının yani yanlış tahminlenmiş verilerin sayısının az olmasıdır. İkili sınıflandırıcı için örnek bir Hata matrisi aşağıdaki şekilde ifade edilir.

Tablo 3.1 Hata matrisinin temsili

		VAR OLAN DURUM	
		POZİTİF DURUMLAR	NEGATİF DURUMLAR
TAHMİN	POZİTİF	TP	FP
	NEGATİF	FN	TN

Gerçek pozitif (True Positive – TP) : Olumlu olarak tahmin edilmiş ve tahminin doğru olduğu durumların sayısıdır.

Gerçek negatif (True Negative – TN) : Olumsuz olarak tahmin edilmiş ve tahminin doğru olduğu durumların sayısıdır.

Yanlış pozitif (False Positive – FP) : Olumlu olarak tahmin edimiş ve tahminin yanlış olduğu durumların sayısıdır.

Yanlış negatif (False Negative – FN): Olumsuz olarak tahmin edilmiş ve tahminin yanlış olduğu durumların sayısıdır.

Hata matrisi baz alınarak hesaplanan bazı oranlar doğruluk, kesinlik, duyarlılık, F1 skoru ve Kappa skorudur.

3.2.2 Doğruluk (Accuracy)

Oluşturmuş olan modelde doğru olarak yapılan tahminlerin tüm tahminlere oranı ile hesaplanır [46,47].

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{3.37}$$

3.2.3 Kesinlik (Precision)

Pozitif olarak tahmin edilen bir durumdaki başarıyı gösteren bu ölçüt doğru pozitif tahminlerin tüm pozitif tahminlere oranı ile hesaplanır. Pozitif Tahmin Edici Değer (Positive Predictive Value) olarak da bilinir [47].

$$Precision = \frac{TP}{TP+FP} \tag{3.38}$$

3.2.4 Hassasiyet (Recall)

Pozitif durumların ne kadar başarılı tahmin edildiğini gösterir. Sonuçların ne kadar eksiksiz olduğu sorusuna cevap verir [47].

$$Recall = \frac{TP}{TP+FN} \tag{3.39}$$

3.2.5 F1 Skor

Sınıflandırıcının ne kadar iyi performans gösterdiğini ölçmek ve sınıflandırıcıları karşılaştırmak amacıyla recall ve precision oranlarının harmonik ortalaması alınarak hesaplanır [48].

$$F - Skor = \frac{2.precision.recall}{precision+recall} \tag{3.40}$$

3.2.6 ROC Eğrisi ve Eğri Altında Kalan Alan

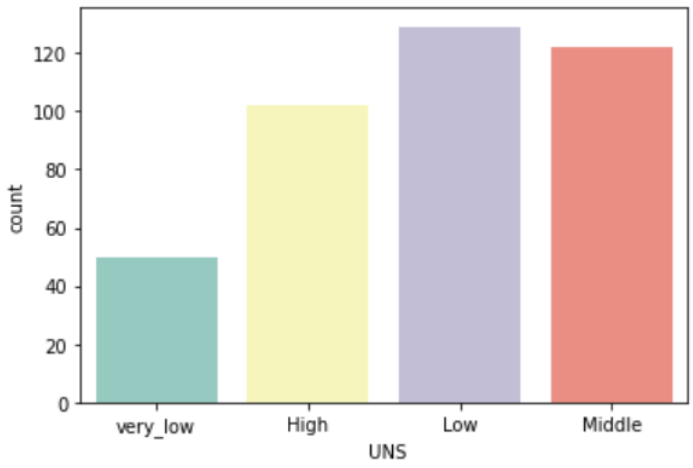
ROC eğrisi, yanlış pozitif oranı x ekseni, doğru pozitif oranı y ekseni ile gösterilen bir düzlemde sınıflandırıcının gerçek pozitif oranı ile yanlış pozitif oranı arasındaki dengeyi gösteren grafiksel bir ölçüttür [45]. İki sınıf arasında bir parametrenin ne denli ayırt edici bir ölçü olduğunu belirlemek için çizilen bu eğrinin altında kalan alanı temsil eden AUC (Area Under Curve) kullanılır.

3.2.7 Kappa Skoru

Kappa skoru olarak adlandırlan başarı ölçütü kategorik veriler arasındaki uyumu ölçen istatistiksel bir yöntemdir [49]. -1 ile 1 arasında değer alabilir.

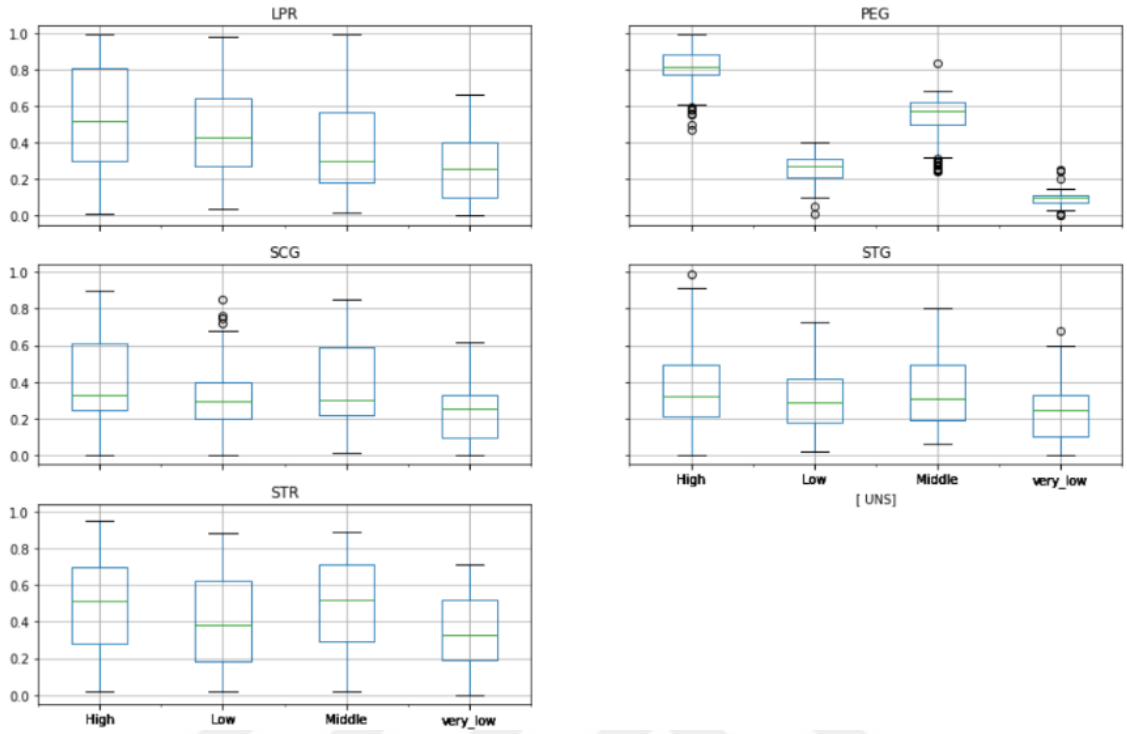
4
MATEMATİKSEL MODELLEMELER ÜZERİNE BİR UYGULAMA

Bu çalışmada https://archive.ics.uci.edu/ml/index.php internet sitesinden H. Tolga Kahraman'a ait 403 adet bilgi içeren ve kayıp verisi bulunmayan "User Knowledge Modeling Data Set" adlı veri seti alınarak; verinin incelenmesi, modellerin kurulması, sonuçların görselleştirilmesi için Jupyter Notebook üzerinde Numpy, Pandas, Matplotlib, Seaborn, Sklearn kütüphaneleri kullanılmıştır [50,51]. Veri seti; STG, SCG, STR, LPR, PEG isimli beş nümerik değerli özniteliğe ve UNS isimli kategorik değerli sınıf etiketine sahiptir. STG; Hedef nesne materyalleri için çalışma süresinin derecesi, SCG; Hedef nesne materyalleri için kullanıcının tekrarlama sayısı, STR; Hedef nesne ile ilgili nesneler için kullanıcının çalışma süresi derecesi, LPR; Hedef nesne ile ilgili nesneler için kullanıcının sınav performansı, PEG; Kullanıcının hedef nesneler için sınav performansı ve UNS; Kullanıcıların bilgi düzeyini göstermektedir [52]. Söz konusu veri setinin sınıf dağılımı aşağıdaki şekilde görüldüğü gibi dengeli bir dağılıma sahiptir.



Şekil 4.1 Veri setinin sınıf dağılım grafiği

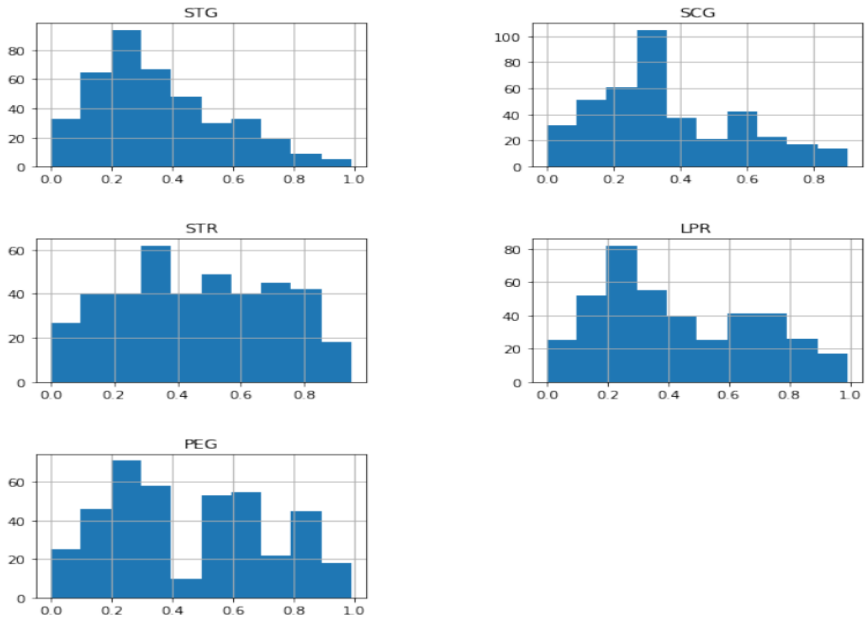
Beş özniteliğin sınıf etiketlerine göre dağılımlarının görsel olarak gözlemlenmesi amacıyla kutu grafiği şekil 4.2'de sunulmuştur.



Şekil 4.2 Veri setinin kutu grafiği ile görselleştirilmesi

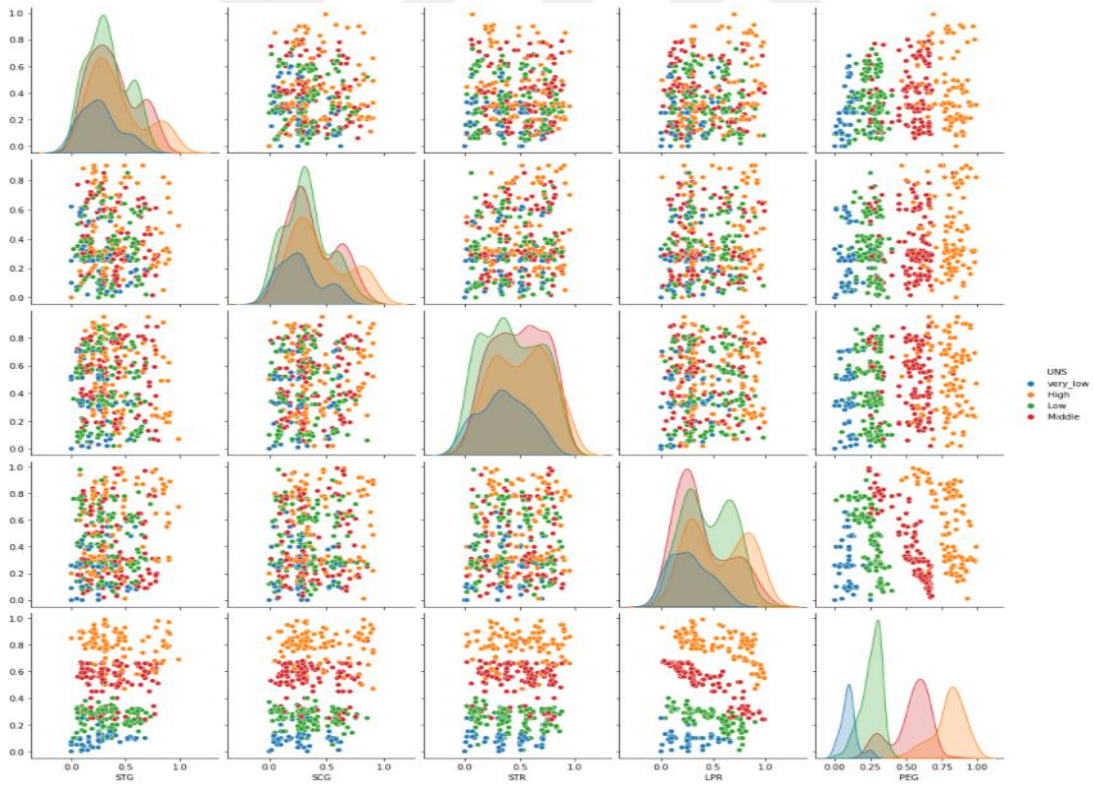
Şekil 4.2 'de görüldüğü üzere sınıf dağılımının PEG özniteliğinde diğer özniteliklere göre daha belirgin olduğu görülmektedir.

Özniteliklerin nümerik değerlerinin dağılımı şekil 4.3'teki gibi görselleştirilebilir:



Şekil 4.3 Veri setinin nümerik değerli özniteliklerinin dağılım grafiği

Özniteliklerin sınıf dağılımı ise aşağıdaki gibidir:



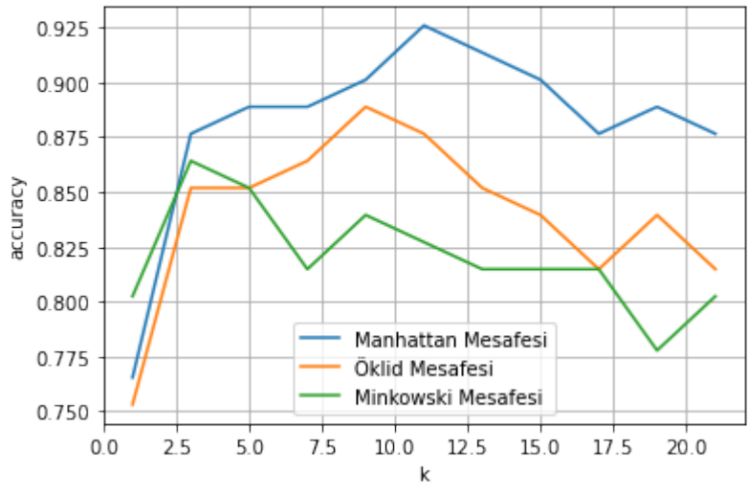
Şekil 4.4 Veri setinin özniteliklerinin sınıf dağılım grafiği

Ele alınan veri setinin modellenmesi amacı ile bir önceki bölümde bahsedilen En yakın komşu algoritması, K-ortalama değer algoritması, Naive Bayes algoritması, Karar ağacı algoritması, Destek vektör makineleri, boosting ve Rastgele orman modelleme algoritmaları tek tek ele alınacaktır; söz konusu olan modellemelerde model başarısını her ele alınan modelde maksimum yapacak parametreler belirlenecek ve en yüksek başarıyı sağlayacak modellemenin belirlenmesi sağlanacaktır.

Veri setinin model başarısını incelerken önce bahsettiğimiz algoritmalar tek tek veri seti %80-%20 olarak eğitim ve test verisi için ayrılarak model başarıları gözlenmiş ayrıca çapraz doğrulama metodu ile kontrol edilmiş, buna ilave olarak her bir modelde Grid Search yöntemi kullanılarak uygun parametreler belirlenmiştir. Her bir model için bulunan uygun parametreler ile model başarıları ayrı ayrı hesaplanmıştır.

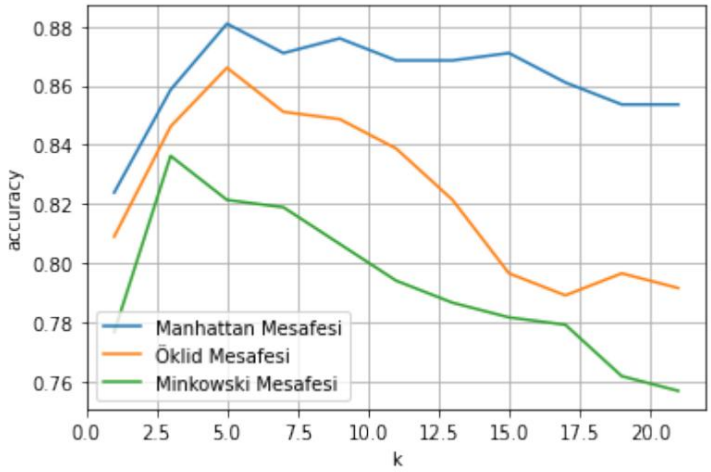
4.1 K-En Yakın Komşu Algoritması

K-NN algoritmasını uygularken ilk olarak veri seti %80-%20 olarak ayrılmış ve modellemenin başarısını ölçmek için farklı metrikler ile (Minkowski uzaklığı, Öklid uzaklığı ve Manhattan uzaklığı ile) hesaplama yapılmıştır. Söz konusu olan metriklerin komşu sayısına göre model başarıları ölçeklendirilerek elde edilen sonuçlar için grafikler aşağıdaki şekilde elde edilmiştir.



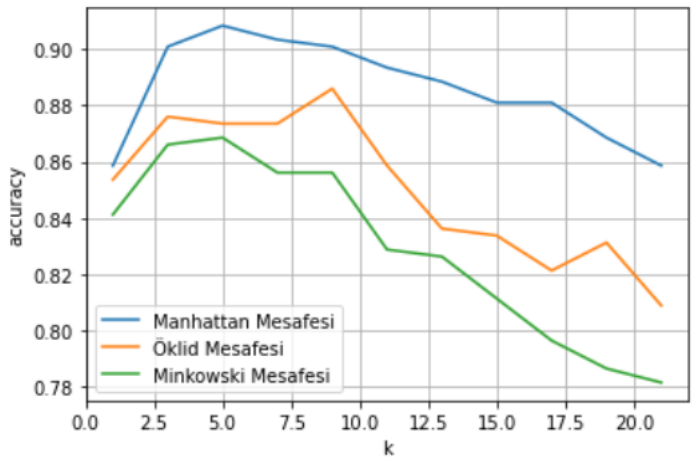
Şekil 4.5 Metriklerin komşu sayısına göre model başarılarının ölçümlenmesi

Bu metrikler içerisinde model başarısı en yüksek olan metrik Manhattan Metriği olmuştur. Yukarıda belirtilen işlem çapraz doğrulama ile modellendiğinde aşağıdaki grafikte özetlenen sonuçlar elde edilmiştir.



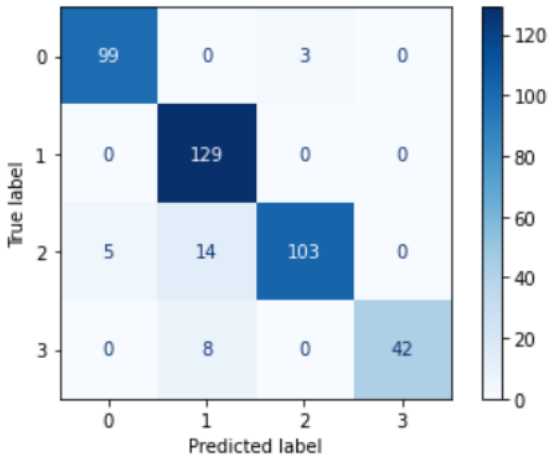
Şekil 4.6 Çapraz doğrulama sonucu metriklerin komşu sayısına göre model başarılarının ölçümlenmesi

Çapraz doğrulama ile birlikte stratified k-fold metodu birlikte kullanıldığında elde edilen sonuçlar aşağıdaki grafikte ifade edildiği gibi gözlemlenmiştir.



Şekil 4.7 Çapraz doğrulama ve stratified k-fold metodunun birlikte kullanımı ile metriklerin komşu sayısına göre model başarılarının ölçümlenmesi

K-en yakın komşu algoritması için en iyi başarı ölçütü farklı metriklere göre hesaplanarak grafik ile görselleştirilmesi sonrasında ve aynı zamanda Grid Search ile doğrulandığında en iyi başarı ölçütü için komşuluk seçiminin 5 olduğu belirlenmiştir. Tüm modeller arasından en iyi model başarısına sahip modelin model başarısı %88,2 olarak hesaplanmış olup hata matrisi şekil 4.8'de sunulmuştur. 403 veriden 373 tanesinin sınıfı doğru tahminlenmiştir.



Şekil 4.8 K en yakın komşu algoritması kullanılarak hesaplanan hata matrisi

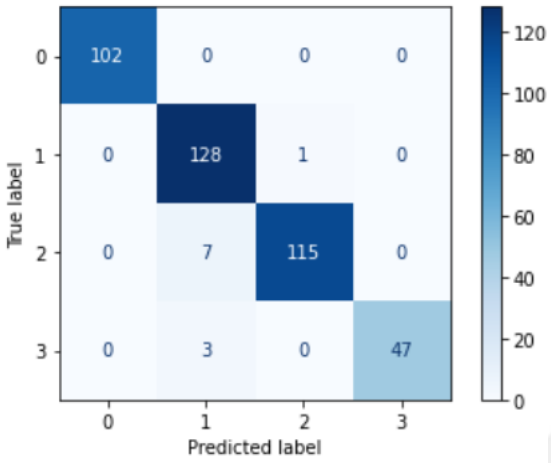
4.2 Rastgele Orman Algoritması

Bu algoritma ile önce değişkenlerin önem sırası ve önem derecesi şekil 4.9'da gösterildiği üzere belirlenmiştir.

	feature	importance
4	PEG	0.802940
3	LPR	0.168451
2	STR	0.017874
1	SCG	0.009370
0	STG	0.001366

Şekil 4.9 Rastgele orman algoritması ile belirlenen değişkenlerin önem sırası

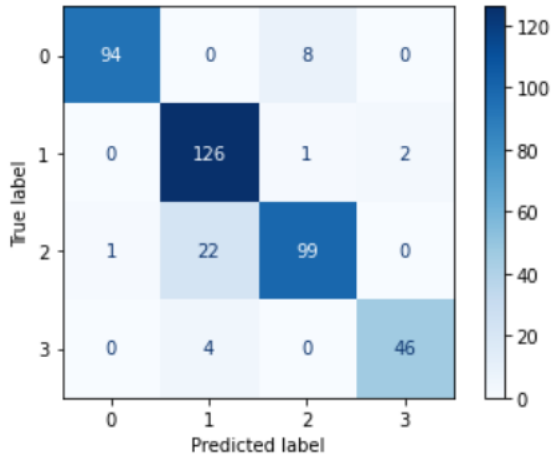
Daha sonra modellemede kullanılan en uygun parametreler Grid Search ile belirlenerek model başarısı %93,3 olarak elde edilmiş olup hata matrisi şekil 4.10'da sunulmuştur. 403 veriden 392 tanesinin sınıfı doğru tahminlenmiştir.



Şekil 4.10 Rastgele orman algoritması kullanılarak hesaplanan hata matrisi

4.3 Naive Bayes Algoritması

Naive Bayes algoritmasını uygularken ilk olarak veri seti %80-%20 olarak ayrılmıştır ve model başarısı %87,65 bulunmuştur. Daha sonra çapraz doğrulama ile model başarısı %87,59 olarak hesaplanmış olup hata matrisi şekil 4.11'de sunulmuştur. 403 veriden 365 tanesinin sınıfı doğru tahminlenmiştir.

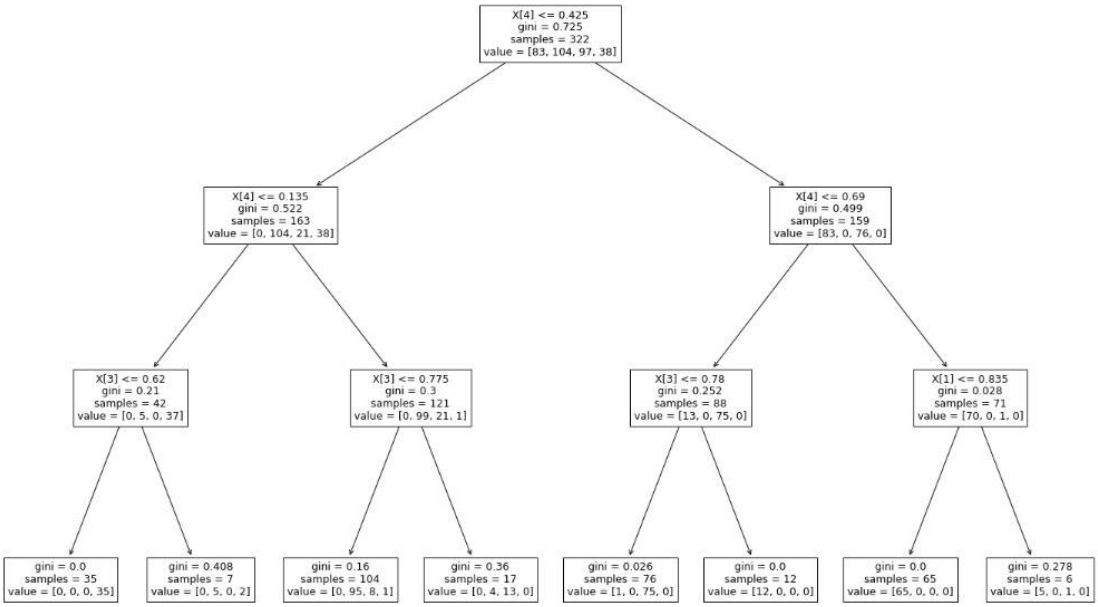


Şekil 4.11 Naive bayes algoritması kullanılarak hesaplanan hata matrisi

4.4 Karar Ağaçları

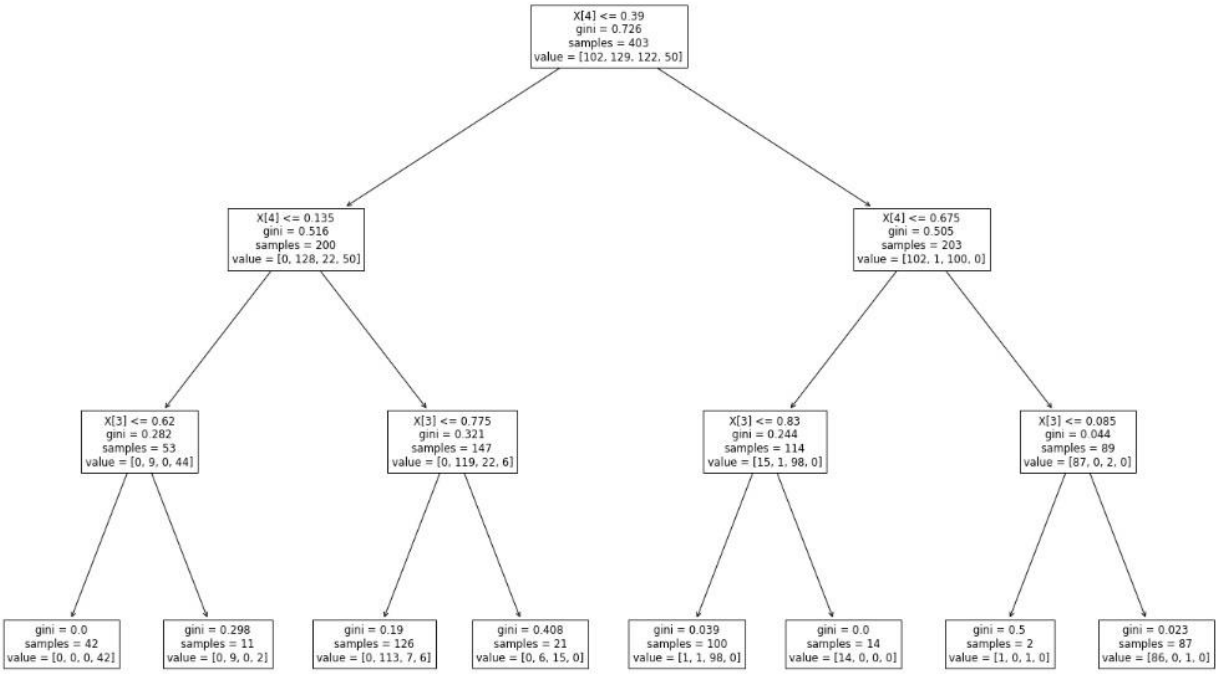
- **Gini ile Karar Ağacı**

Öncelikle veri %80 eğitim %20 test olarak ayrılarak ‘gini’ kriterine göre hesaplanan karar ağaçları model başarı ölçütü %.88,89 olarak bulunmuş, modelde aşırı öğrenme durumunun oluşup oluşmadığını kontrol etmek amacı ile model eğitim kümesi için model başarı başarı ölçütü tekrar hesaplanarak %94,72 bulunmuştur. Ağacın bu kritere göre görseli şekil 4.12’de sunulmuştur:



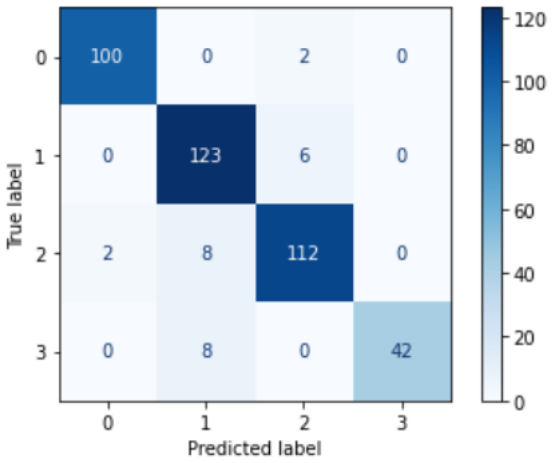
Şekil 4.12 Gini kriteri kullanılarak oluşturulan karar ağacı

Çapraz doğrulama metodu ile bu başarı ölçütü %90,81 olarak hesaplanmış olup karar ağacı şekil 4.13’te görselleştirilmiştir.



Şekil 4.13 Gini kriteri kullanılarak çapraz doğrulama ile oluşturulan karar ağacı

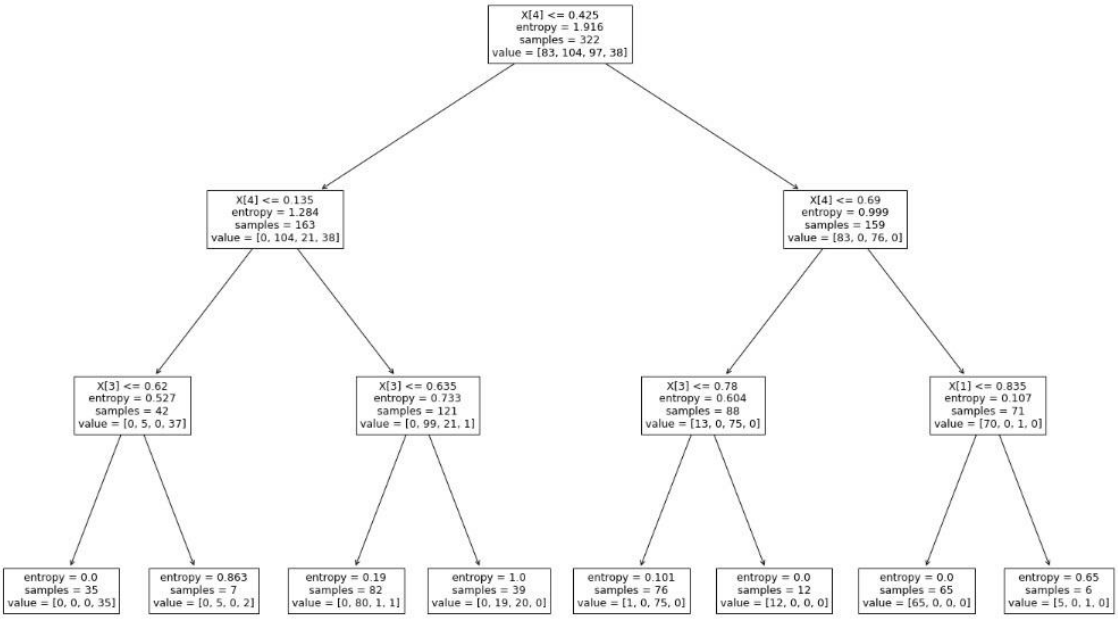
Hata matrisi şekil 4.14'te sunulmuş olup 403 veriden 377 tanesinin sınıfı doğru tahminlenmiştir.



Şekil 4.14 Gini kriteri kullanılarak karar ağacı ile hesaplanan hata matrisi

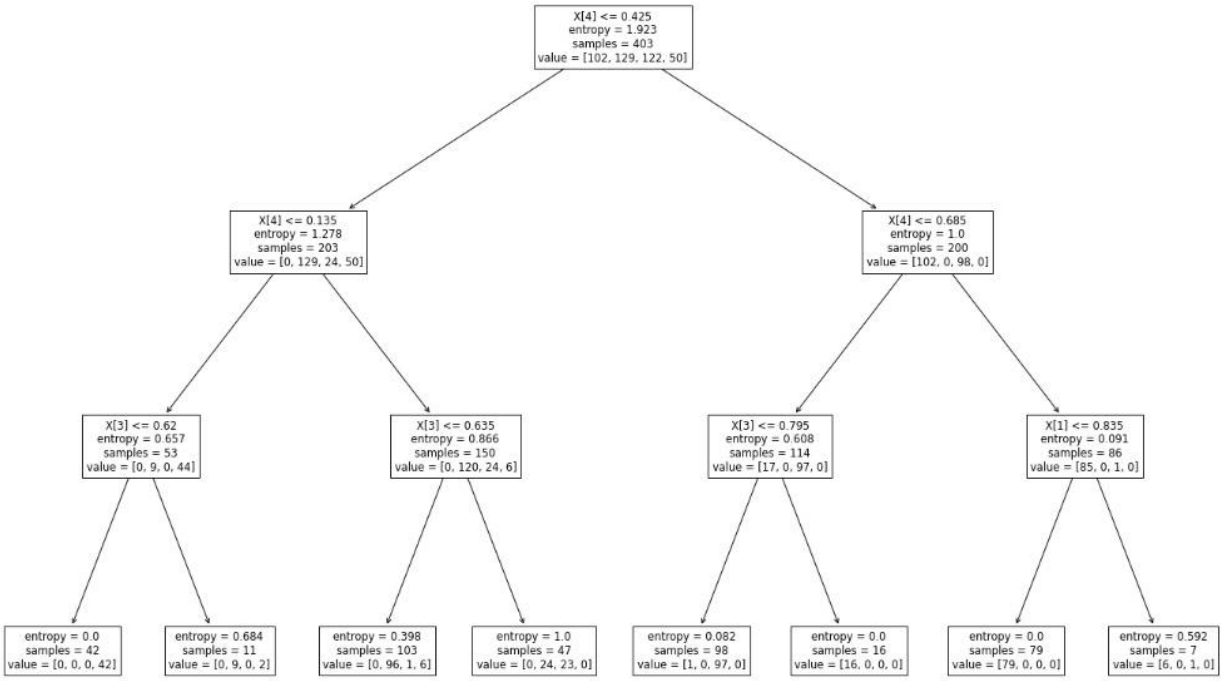
- **Entropi ile Karar Ağacı**

Öncelikle veri %80 eğitim %20 test olarak ayrılarak ‘entropi’ kriterine göre hesaplanan karar ağaçları model başarı ölçütü %85,19 olarak bulunmuş, modelde aşırı öğrenme durumunun oluşup oluşmadığını kontrol etmek amacı ile model eğitim kümesi için model başarı başarı ölçütü tekrar hesaplanarak %92,24 bulunmuştur. Ağacın bu kritere göre görseli şekil 4.15’te sunulmuştur:



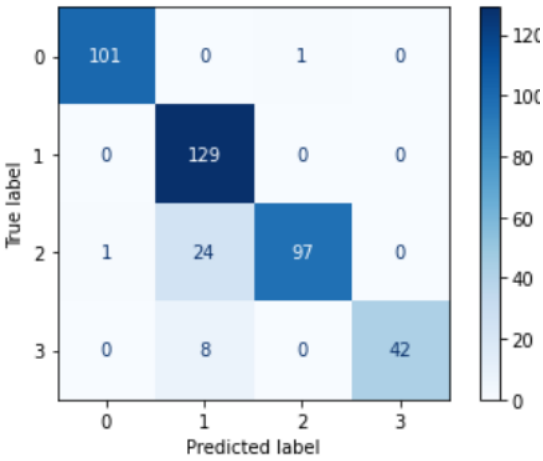
Şekil 4.15 Entropi kriteri kullanılarak oluşturulan karar ağacı

Çapraz doğrulama metodu ile başarı ölçütü %87.34 olarak hesaplanmış olup karar ağacı şekil 4.16’da görselleştirilmiştir:



Şekil 4.16 Entropi kriteri kullanılarak çapraz doğrulama ile oluşturulan karar ağacı

Hata matrisi şekil 4.17'de sunulmuş olup 403 veriden 369 tanesinin sınıfı doğru tahminlenmiştir.

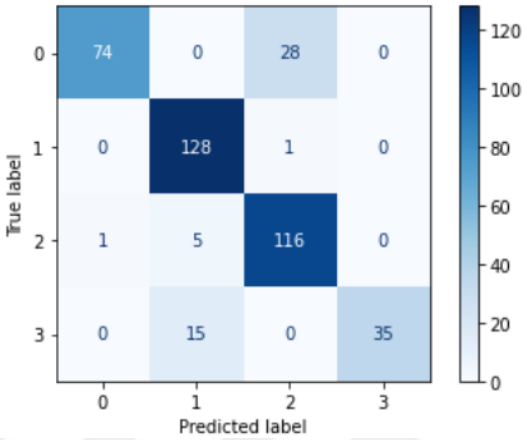


Şekil 4.17 Entropi kriteri kullanılarak karar ağacı ile hesaplanan hata matrisi

4.5 Boosting

- **AdaBoost**

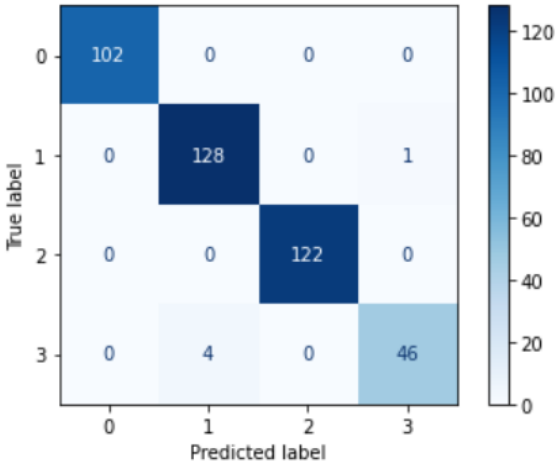
  GridSearch ile En İyi Parametreler: {'learning_rate': 3, 'n_estimators': 400} şeklinde belirlenmiş ve bu parametreler ile modelin başarısı %82,71 olarak hesaplanmış olup hata matrisi şekil 4.18'de sunulmuştur. 403 veriden 353 tanesinin sınıfı doğru tahminlenmiştir.



Şekil 4.18 Adaboost algoritması kullanılarak hesaplanan hata matrisi

- **XGBoost**

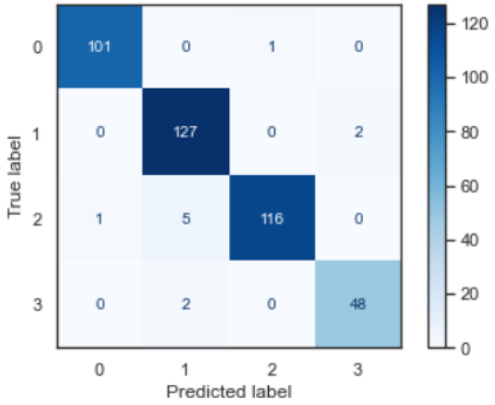
GridSearch ile En İyi Parametreler: {'learning_rate': 1, 'max_depth': 2, 'n_estimators': 300}şeklinde belirlenerek bu parametreler ile modelin başarısı %93.82 olarak hesaplanmış olup hata matrisi şekil 4.19'da sunulmuştur. 403 veriden 398 tanesinin sınıfı doğru tahminlenmiştir.



Şekil 4.19 XGboost algoritması kullanılarak hesaplanan hata matrisi

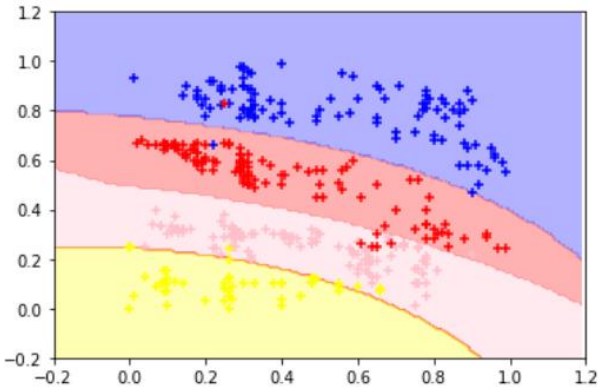
4.6 Destek Vektör Makineleri

cDestek vektör makineleri metodu ile modelleme oluştururken kullanılacak parameterlerin optimize edilmesi amacı ile bu bölümde de Grid Search kullanılmış ve beş öznitelik, dört sınıf için elde edilen {'C': 100, kernel='linear'} parametreleri için model başarı ölçütü %96 olarak elde edilmiş olup hata matrisi şekil 4.20'de sunulmuştur. 403 veriden 392 tanesinin sınıfı doğru tahminlenmiştir.



Şekil 4.20 Destek vektör makineleri kullanılarak hesaplanan hata matrisi

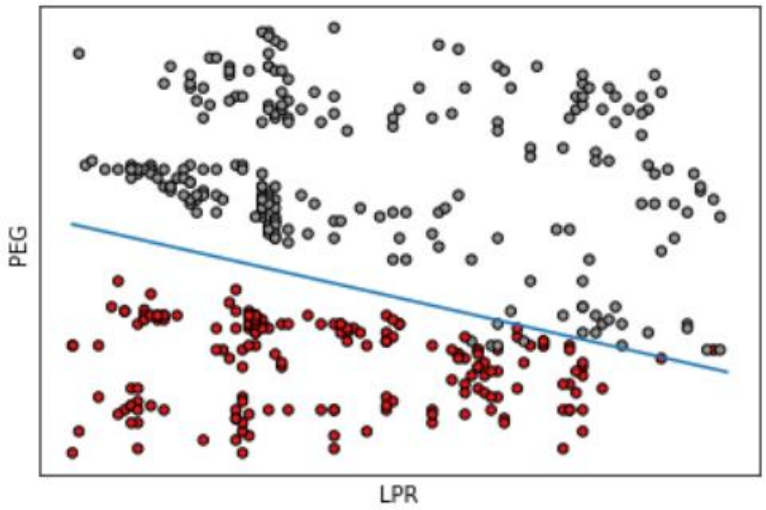
Aynı işlemler öznitelik sayısını sırası ile üç ve ikiye düşürülerek tekrar edildiğinde: üç öznitelik için {'C':17, kernel='linear'} parametreleri model başarısı %95,2 olarak bulunmuştur. İki öznitelik için ise {'C':5, kernel='rbf'} parametreleri model başarısı %95 olarak gözlemlenmiştir.



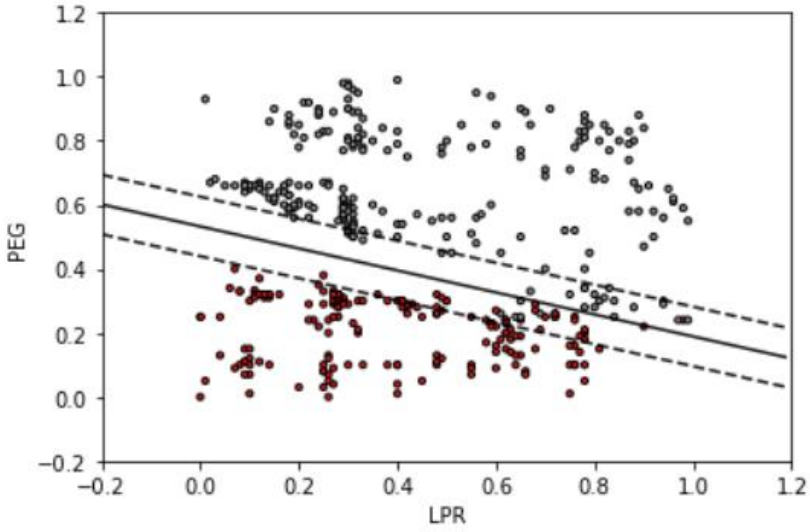
Şekil 4.21 Dört sınıflı modelin {'C':5, kernel='rbf'} için düzlem üzerinde görselleştirilmesi

Veriyi düzlemde görselleştirmek amacı ile beş öznitelik sayısını daha önce rastgele orman bölümünde incelenen önem sırasına göre sırası ile iki ve üç özniteliğe azaltarak; sınıf sayısını orta altı ve orta üstü olmak üzere iki sınıf sayısına indirerek işlemler yinelenmiştir. İki sınıf sayısı beş öznitelik için {'C': 1000, kernel='linear'} parametreleri ile model başarısı %98,7; iki sınıf sayısı üç öznitelik için {'C': 17, kernel='linear'} parametreleri ile model başarısı %98,5; iki sınıf sayısı iki öznitelik için {'C': 4, kernel='linear'} parametreleri ile model başarısı %98,5 olarak elde edilmiştir.

Böylece verilerin düzlemde hem doğrusal olarak, hem de eğrisel olarak iki bölgeye ayrılması hedeflenmiştir [53, 54]. Bu amaçla hem lineer hem de rbf için en uygun C parametresi belirlemek için tekrar grid search kullanılmıştır. kernel='linear' için en uygun C parametresi 100 olarak bulunmuş, model başarısı %98,76 olarak belirlenmiştir. Bu durumda verinin görseli şekil 4.22'de sunulmuştur.

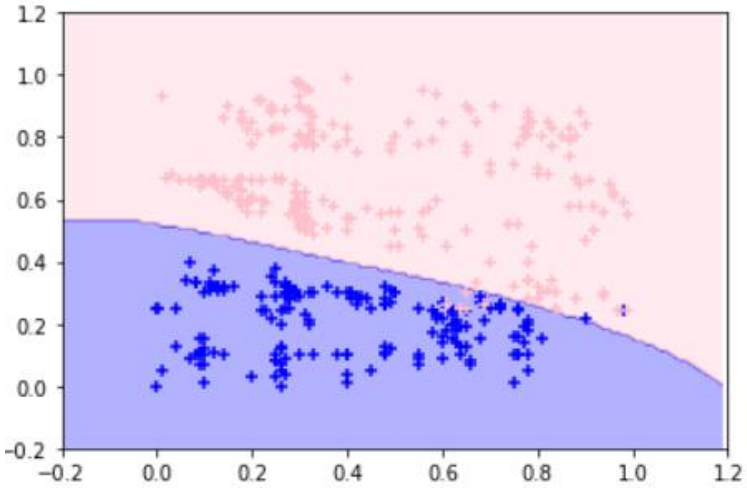


Şekil 4.22 İki sınıflı modelin {'C': 4, kernel='linear'} düzlem üzerinde görselleştirilmesi



Şekil 4.23 İki sınıflı model için maksimum marjlı hiper düzlem

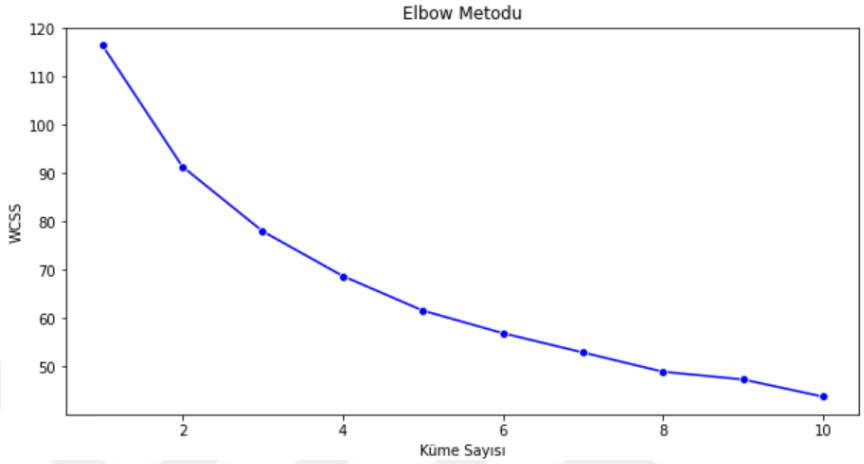
Aynı işlemler tekrarlanmış, kernel='rbf' için en uygun C parametresi 1000 olarak bulunmuş, model başarısı %97.53 olarak belirlenmiştir. Bu durumda verinin görseli aşağıdaki şekil 4.24'te sunulmuştur.



Şekil 4.24 İki sınıflı modelin {'C':4, kernel='rbf'} için düzlem üzerinde görselleştirilmesi

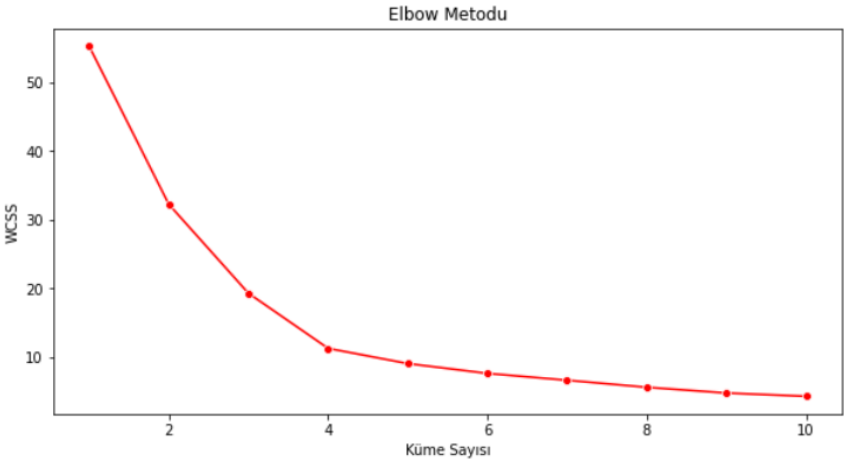
4.7 K- Ortalamalar Algoritması

K ortalamalar algoritması ile en uygun k-küme (sınıf) sayısının belirlenmesi hedeflenmiştir. Bu amaçla sınıf etiketleri kapatılarak beş özniteliğe göre dirsek metodu kullanılmış ve aşağıdaki grafik oluşturulmuştur.



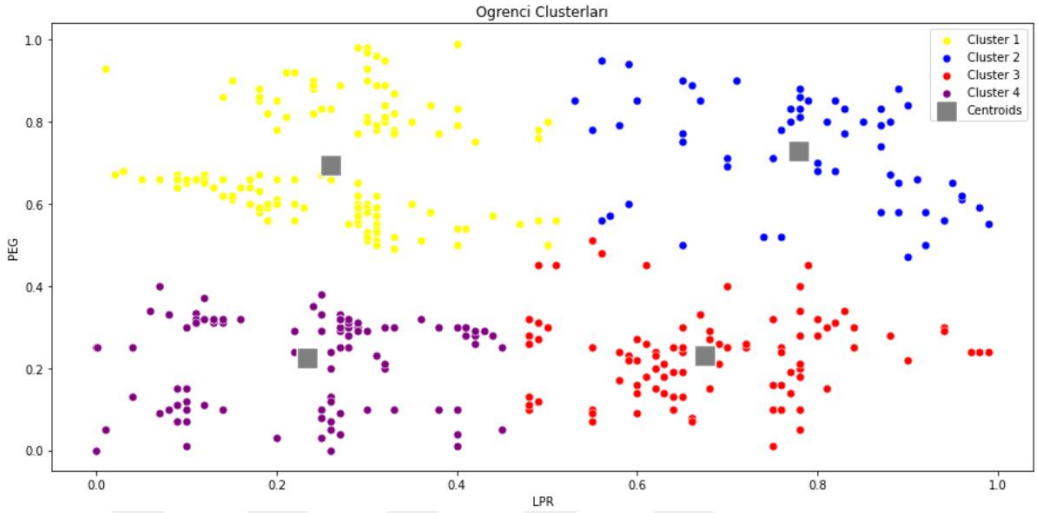
Şekil 4.25 Beş öznitelikli model için dirsek metodunun uygulanması

Kümeleme sayısının belirgin hale getirilmesi amacı ile öznitelik sayısının daha önce rastgele orman algoritmasında elde ettiğimiz önem sırasına göre azaltılarak tekrar uygulanması sonucunda aşağıdaki grafik elde edilmiştir.



Şekil 4.26 İki öznitelikli model için dirsek metodunun uygulanması

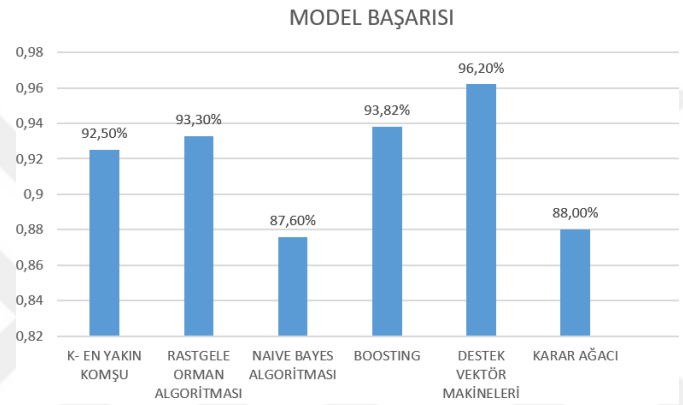
Dirsek noktası 4 civarında bulunmuştur ve bu sayı veri kümesi için seçilmesi gereken optimal küme sayısı olarak belirlenmiştir. 4 küme sayısı ile algoritma çalıştırıldığında elde edilen sınıflandırma aşağıdaki şekilde belirtilmiştir.



Şekil 4.27 K ortalamalar algoritması ile verilerin kümeleştirilmesi

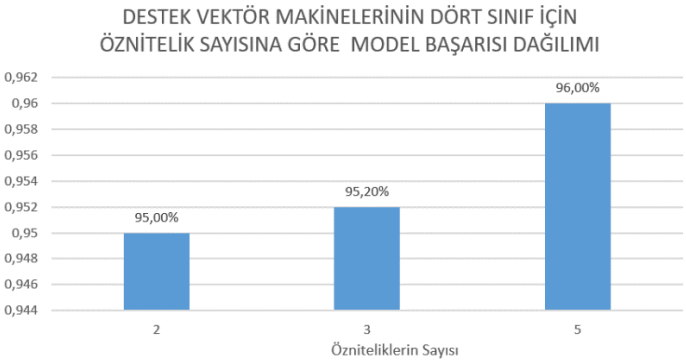
5
SONUÇ VE ÖNERİLER

Veri seti bahsi geçen çeşitli algoritmalar kullanılarak ele alınmış; model doğruluk ölçütlerinden 'accuracy' seçilerek model başarısı aşağıdaki şekilde ölçümlendirilmiştir. Bu doğrultuda model başarısı en yüksek olan algoritma Destek Vektör Makineleri olmuştur.

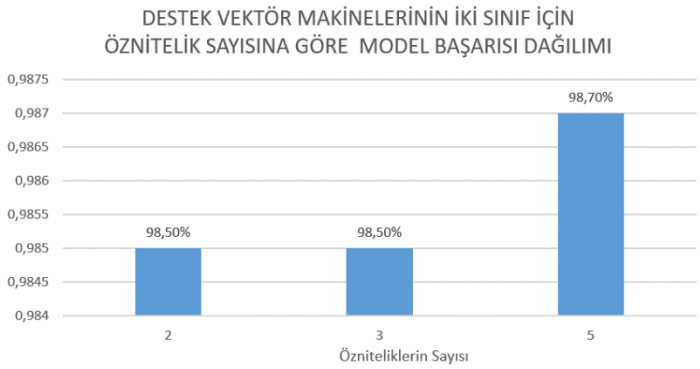


Şekil 5.1 Uygulanan algoritmaların model başarıları

Destek vektör makineleri ayrıca ele alınmış öznitelik sayısı önem sırasına göre azaltılmış, öznitelik sayısı ve model başarısı arasındaki ilişki incelenmiş ve aşağıda sunulmuştur.



Şekil 5.2 Destek vektör makinelerinin dört sınıf için öznitelik sayısına bağlı model başarı grafiği



Şekil 5.3 Destek vektör makinelerinin iki sınıf için öznitelik sayısına bağlı model başarı grafiği

KAYNAKÇA

[1] A. M. Legrende, “Nouvelles méthodes pour la détermination des orbites des comètes; par AM Legendre...” chez Firmin Didot, libraire pour lew mathematiques, la marine, l'architecture, et les editions stereotypes, rue de Thionville, 1806.

[2] A. M. Turing, “The essential turing”, Oxford University Press, 2004.

[3] W. S. McCulloch ve W. Pitts, “A logical calculus of the ideas immanent in nervous activity.” The bulletin of mathematical biophysics, 5(4), 115-133, 1943.

[4] A. L. Samuel, “Some studies in machine learning using the game of checkers.” II-Recent progress. IBM Journal of research and development, 11(6), 601-617, 1967.

[5] A. G. Ivakhnenko ve V. G. Lapa, “Cybernetic predicting devices.” Purdue Univ Lafayette Ind School Of Electrical Engineering, 1966.

[6] J. H. Holland, “Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence.” MIT press, 1992.

[7] U. Fayyad, G. ve Piatetsky-Shapiro, and, P. Smyth, “From data mining to knowledge discovery in databases” , AI magazine, 17(3), 37-37, 1996.

[8] J. Han, J. Pei, ve M Kamber, “Data mining: concepts and techniques.”, Elsevier, 2011.

[9] L. Breiman, “Random forests”, Machine learning, 45(1), 5-32, 2001.

[10] K. Mittal, D. Khanduja ve P. C. Tewari, “An insight into ‘Decision Tree Analysis’”. World Wide Journal of Multidisciplinary Research and Development, 3(12), 111-115, 2017.

[11] Y. Freund, and R. E. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting.”, Journal of computer and system sciences, 55(1), 119-139, 1997.

[12] B. Kégl, “The return of AdaBoost. MH: multi-class Hamming trees.” arXiv preprint arXiv:1312.6086, 2013.

[13] T. Chen, ve C. Guestrin, "Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining", 2016, 785-794.

[14] C. Cortes and V. Vapnik, " Support-vector networks. Machine learning " , 20(3), 273-297, 1995

[15] A. Y. Chervonenkis, "Early history of support vector machines. In Empirical Inference Springer, Berlin, Heidelberg, 2013,13-20.

[16] J. MacQueen, "Some methods for classification and analysis of multivariate observations." In Proceedings of the fifth Berkeley symposium on mathematical statistics and probability Vol. 1, No. 14, 1967, 281-297

[17] D. T. Larose ve C. D. Larose "Discovering knowledge in data: an introduction to data mining "(Vol. 4). John Wiley & Sons, 2014

[18] W. Kresse, "Springer handbook of geographic information ", D. M. Danko (Ed.). Berlin: Springer, 2012, 118-120

[19] G. Xu, Y. Zong, & Z. Yang, "Applied data mining.", CRC Press, 2013.

[20] D. Ienco, R. G. Pensa, ve R. Meo, "From context to distance: Learning dissimilarity for categorical data clustering."ACM Transactions on Knowledge Discovery from Data (TKDD), 6(1), 2012, 1-25.

[21] M. A. Hall ve G. Holmes, "Benchmarking attribute selection techniques for discrete class data mining." IEEE Transactions on Knowledge and Data engineering, 15(6), 2003, 1437-1447.

[22] E. Taşcı ve A. Onan, "K-en yakın komşu algoritması parametrelerinin sınıflandırma performansı üzerine etkisinin incelenmesi.", Akademik Bilişim, 1(1), 2016, 4-18.

[23] J. Maillo, J. Luengo, S. García, F. Herrera, ve I. Triguero, "Exact fuzzy k-nearest neighbor classification for big datasets." IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), IEEE, 2017, 1-6

[24] G. Biau ve E. Scornet, “A random forest guided tour.” Test, 25(2), 2016, 197-227.

[25] H. T. Kam, “Random decision forest.”, In Proceedings of the 3rd international conference on document analysis and recognition (Vol. 1416, p. 278282). Montreal, Canada, August, 1995.

[26] M. E. Taşçı, ve R. ŞAMLI, “Veri madenciliği ile kalp hastalığı teşhisi.”, Avrupa Bilim ve Teknoloji Dergisi, 2020, 88-95

[27] S. Atan, “Knn, Naive Bayes Ve Karar Ağaci Makine Öğrenme Algoritmalari, Bu Algoritmalarin Sosyal Bilimlerde Kullanim İmkânlari.”, Paper DOI: 10.31235/osf.io/8r5pu, 2020.

[28] Yamuk B., “ Elektronik Postalarin Ayriştirilmasinda Naive Bayesian Ve Bulanik Mantik Yöntemlerinin Karşilaştirilmasi”, Yüksek Lisans Tezi, 2011.

[29] P. E. Hart, D. G. Stork ve R. O. Duda, “Pattern classification.” , Hoboken: Wiley, 2000.

[30] E. Alpaydın, “Yapay Öğrenme”, Boğaziçi Üniversitesi Yayınevi, İstanbul, 2010.

[31] H. Schütze, C. D. Manning ve P. Raghavan, “ Introduction to information retrieval” , Vol. 39, Cambridge: Cambridge University Press, 2008, 234-265

[32] Z. Güner, “Veri madenciliğinde CART ve lojistik regresyon analizinin yeri: ilaç provizyon sistemi verileri üzerinde örnek bir uygulama.”, Sosyal Güvence Dergisi, (6), 2014, 53-99.

[33] Yangın G, “ Xgboost ve Karar Ağacı Tabanlı Algoritmaların Diyabet Veri Setleri Üzerine Uygulaması.”, İstanbul : Mimar Sinan Güzel Sanatlar Üniversitesi, Fen Bilimleri Enstitüsü, 2019.

[34] Yılmaz S. “Topluluk Öğrenme Yöntemini Kullanarak Twitter Verisi Üzerinde Duygu Algılama ve Tanıma.” , İzmir : Computers and Electrical Engineering ; Ege Üniversitesi, Fen Bilimleri Enstitüsü, 2019.

[35] E. Ekiz, “Makine Öğrenmesi Teknikleri ile Tahsilat Davranışı Tahmini: Telekomünikasyon Sektörü Örneği.” , İstanbul : İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, 2019.

[36] S. Metlek ve K. Kayaalp, “Makine Öğrenmesinde, Teoriden Örnek MATLAB Uygulamalarına Kadar Destek Vektör Makineleri.”, İksad Yayınevi, 2020.

[37] L. Kaufman and P. J. Rousseeuw, “Clustering by means of Medoids. Statistical data analysis based on the L1–norm and related methods”, edited by Y. Dodge, 1987.

[38] P. N. Tan, M. Steinbach ve Kumar, V “Introducing to Data Mining.”, 2006.

[39] A. Gersho, ve R. M. Gray, “Vector quantization and signal compression”, Vol. 159, Springer Science & Business Media, 2012.

[40] Y. Linde, A. Buzo, ve R. Gray, “An algorithm for vector quantizer design.” IEEE Transactions on communications, 28(1), 1980, 84-95.

[41] G. Sariman, “Veri madenciliğinde kümeleme teknikleri üzerine bir çalışma: k-means ve k-medoids kümeleme algoritmalarının karşılaştırılması.” ,Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 15(3), 2011, 192-202.

[42] T. H. Sarma, P. Viswanath ve B. E. Reddy, “Speeding-up the kernel k-means clustering method: A prototype based hybrid approach.”, Pattern Recognition Letters, 34(5), 2013, 564-573,

[43] I. S. Dhillon, Y. Guan, ve B. Kulis, “ Kernel k-means: spectral clustering and normalized cuts.” , In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining , 2004, 551-556.

[44] Ö. F. Rençber, “ Veri Madenciliğinde Kullanılan Kümeleme Algoritmaları ve R ile Uygulamalı Örnekler.” , Nobel Yayın Evi, 978-625-7126-26-7, 2020.

[45] P. Tan, M. Steinbach, V. Kumar, “Performance Measure” in Introduction to Data Mining, Pearson Education Limited (UK), 2014.

[46] F. Uyanık, M. C. Kasapbaşı, “ Telekomünikasyon Sektörü için Veri Madenciliği ve Makine Öğrenmesi Teknikleri ile Ayrılan Müşteri Analizi ”, Düzce Üniversitesi Bilim ve Teknoloji Dergisi, 9 (2021) 172-191

[47] Sokolova, M. and Lapalme, G., “A systematic analysis of performance measures for classification tasks.” Information Processing & Management, 45(4), 427- 437 2009.

[48] Powers, David M. W., "Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation.", 2011.

[49] J. Cohen, " A coefficient of agreement for nominal scales." , Educational and psychological measurement, 20(1), 1960, 37-46.

[50] UCI Machine Learning Repository; Center For Machine Learning and Intelligent Systems: https://archive.ics.uci.edu/ml/datasets/User+Knowledge+Modeling

[51] H. T. Kahraman, " Web-tabanlı uyarlanır zeki öğretim sistemi tasarımı ve uygulaması.", Doktora Tezi, 268176, (2009).

[52] H. T. Kahraman, S. Sagiroglu, ve I. Colak, "The development of intuitive knowledge classifier and the modeling of domain dependent data. "Knowledge-Based Systems, 37, 283-295, 2013.

[53] C. Albon, "Machine learning with python cookbook: Practical solutions from preprocessing to deep learning. " , O'Reilly Media, Inc., 2018.

[54] J. Vanderplas, "Python data science handbook: Essential tools for working with data. " O'Reilly Media, Inc.", 2016

TEZDEN ÜRETİLMİŞ YAYINLAR

Konferans Bildirileri

1. Göksu M. , Kanbay F. , " Mathematical Modelling and Improvement Suggestions in Clustering and Classification Methods for Data Analysis", 5 th International Conference on Mathematical Advances and Applications, İstanbul, Turkey, pp. 210, 2022