T.C.
ONDOKUZ MAYIS ÜNİVERSİTESİ
SAĞLIK BİLİMLERİ ENSTİTÜSÜ
BİYOİSTATİSTİK VE TIP BİLİŞİMİ ANABİLİM DALI

GENETİK ALANINDA ELDE EDİLEN VERİLERİN MAKİNE ÖĞRENİMİ ALGORİTMALARI YARDIMIYLA KARŞILAŞTIRILARAK EN ETKİN YÖNTEMİN BELİRLENMESİ

DOKTORA TEZİ

Senem KOÇ

Samsun
Aralık-2019

T.C.
ONDOKUZ MAYIS ÜNİVERSİTESİ
SAĞLIK BİLİMLERİ ENSTİTÜSÜ
BİYOİSTATİSTİK VE TIP BİLİŞİMİ ANABİLİM DALI

GENETİK ALANINDA ELDE EDİLEN VERİLERİN MAKİNE ÖĞRENİMİ ALGORİTMALARI YARDIMIYLA KARŞILAŞTIRILARAK EN ETKİN YÖNTEMİN BELİRLENMESİ

DOKTORA TEZİ

Senem KOÇ

Danışman
Doç. Dr. Leman TOMAK

II. Danışman
Prof. Dr. Erdem KARABULUT

Samsun
Aralık-2019

T.C.
ONDOKUZ MAYIS ÜNİVERSİTESİ
SAĞLIK BİLİMLERİ ENSTİTÜSÜ

Senem KOÇ tarafından Doç. Dr. Leman TOMAK danışmanlığında hazırlanan "Genetik Alanında Elde Edilen Verilerin Makine Öğrenimi Algoritmaları Yardımıyla Karşılaştırılarak En Etkin Yöntemin Belirlenmesi" başlıklı bu çalışma jürimiz tarafından 30 /12 /2019 tarihinde yapılan sınav ile Biyoistatistik ve Tıp Bilişimi Anabilim Dalında DOKTORA Tezi olarak kabul edilmiştir.

Başkan : Prof. Dr. Soner ÇANKAYA
Ondokuz Mayıs Üniversitesi

Üye : Doç. Dr. Leman TOMAK
Ondokuz Mayıs Üniversitesi

Üye : Prof. Dr. Sezgin GÜNEŞ
Ondokuz Mayıs Üniversitesi

Üye : Prof. Dr. Ergun KARAAĞAOĞLU
Hacettepe Üniversitesi

Üye : Prof. Dr. Atilla Halil ELHAN
Ankara Üniversitesi

ONAY:

Bu tez, Enstitü Yönetim Kurulunca belirlenen ve yukarıda adları yazılı jüri üyeleri tarafından uygun görülmüştür.

..../ ..../.....

Prof. Dr. Ahmet UZUN
Sağlık Bilimleri Enstitü Müdürü

TEŞEKKÜR

Doktora eğitim hayatım boyunca engin bilgi ve deneyimleriyle beni başından sonuna kadar destekleyen değerli danışman hocam Doç. Dr. Leman TOMAK'a en içten teşekkürlerimi sunarım.

Tezimin her aşamasında desteğini esirgemeyen değerli hocam ve diğer danışmanım Prof. Dr. Erdem KARABULUT'a teşekkür ederim.

Tez çalışmamda veri bulma konusunda bana yardımcı olan ve bu konuda bana bilgi aktaran her daim yardımcı olan değerli hocam Prof. Dr. Sezgin GÜNEŞ'e de teşekkür ederim.

Ayrıca, tüm hayatım boyunca her anımda yardımcı olan ve bana her türlü desteği vererek bu günlere gelmemi sağlayan, emeklerini asla ödeyemeyeceğim sevgili annem Şerife KOÇ'a ve babam Ergin KOÇ'a sonsuz teşekkür ederim.

ÖZET

Genetik Alanında Elde Edilen Verilerin Makine Öğrenimi Algoritmaları Yardımıyla Karşılaştırılarak En Etkin Yöntemin Belirlenmesi

Amaç: Makine Öğrenimi (MÖ) sağlık alanında karmaşık veri setlerini çözmek için farklı yöntemler sunmaktadır. Bu çalışmanın amacı sınıflama için kullanılan MÖ algoritmaları ile Super Learner (SL) algoritmasının performansının farklı özellikte genetik veriler üzerinde karşılaştırılmasıdır.

Materyal ve Metot: MÖ için farklı sınıflama algoritmaları kullanılmakta olup, bunlar K En Yakın Komşuluğu (EYK), Naive Bayes (NB), Destek Vektör Makineleri (DVM) ve Rastgele Orman (RO)'dır. Algoritmaların performansları eğri altında kalan alan (EAA) ile değerlendirilmiştir. Çalışmada dengesiz tipteki veriler için yeniden örnekleme yöntemleri kullanılmıştır. Veriyi analize hazırlamak için ön-işleme adımları uygulandıktan sonra, eğitim ve test verisi farklı oranlarda ayrılmıştır. Çalışmada genetik bilgiler içeren, örnek büyüklükleri 587 infertilite verisi ile 174 olan peridontitis veri seti ve iki farklı büyüklükte benzetim veri seti bulunmaktadır. Analizler için R yazılımı kullanılmıştır.

Bulgular: Analiz sonucunda en iyi performanslar, infertilite veri seti %80-%20 olarak ayrıldığında EAA için DVM'de %96, dengesiz veri özellikleri dikkate alındığında %60-%40 olarak ayrıldığında EAA için Sentetik Azınlık Yukarı Örnekleme Tekniği-EYK'de %96 ve SL'de %97 olarak elde edildi. Peridontitis veri seti %60-%40 olarak ayrıldığında EAA için RO %85 ve SL'de aynı sonuç saptandı. İlk benzetim verisi için %60-%40 olarak ayrıldığında EAA için NB'de %78 ve SL'de %81 elde edildi. İkinci benzetim verisi için tüm bölünmelerde NB'de %84 ve SL'de yaklaşık %86 di.

Sonuç: Bu çalışmada MÖ algoritmaları farklı veri setleri üzerinde farklı bölünme oranları ile değerlendirilmiştir. Sonuç olarak SL algoritmasının aynı ya da daha iyi performans gösterdiği saptanmıştır. SL algoritması temel öğreticiler arasında asimtotik olarak aynı ya da tüm öğreticiler arasında en iyi performansı vermektedir.

Senem KOÇ, Doktora Tezi

Ondokuz Mayıs Üniversitesi-Samsun, Aralık, 2019

ABSTRACT

ASSESING THE MOST EFFECTIVE METHODS BY COMPARING MACHINE LEARNING ALGORITHMS FOR DATA OBTAINED IN THE FIELD OF GENETICS

Aim: Machine Learning (ML) offers different methods to solve complex data sets in the field of health. The aim of this study is to compare the performances of ML algorithm used for classification and Super Learner (SL) algorithm on different genetic data.

Material and Method: Different classification algorithms are used for ML. K Nearest Neighbour (KNN), Naive Bayes (NB), Support Vector Machines (SVM) and Random Forest (RF) algorithms were used within the context of this study. Performances of the algorithms were assessed with area under curve (AUC). In the study, resampling methods were used for unbalanced data. Pre-processing steps were applied for analysis, the training and test data were divided in different proportions. Infertility data with a sample size of 587 and periodontitis data set with a sample size of 174, which included genetic information, and two simulation data sets with different sizes were used for analyses. R software was used for analyses.

Results: As a result of the analyses, the best performances were found in SVM for AUC as 96% when infertility data set was divided as 80%-20%, and when unbalanced data were taken into consideration as 96% in KNN with Syntetic Minority Over-Sampling Technique when it was divided as 60%-40% and 97% in SL for AUC. When periodontitis data set was divided as 60%-40%, they were found as 85% in RF and SL for AUC. They were as 78% in NB when divided as 60%-40% and 81% in SL for AUC for the first simulation data. For second simulation data, they were for all divisions 84% in NB for AUC and 86% in SL.

Conclusion: In this study, machine learning algorithms were assessed with different division rates on different data sets. As a conclusion, SL algorithm was found to show as well as or better performance. According to the theory of SL, it performs as well as or better than any of the candidate learners.

Senem KOÇ, Phd Thesis

Ondokuz Mayıs University-Samsun, December, 2019

SİMGELER VE KISALTMALAR

$ACC_{ortalama}$: Ortalama Doğruluk Oranı (Average Accuracy)

ACC_j : j.nci Test Seti için Doğruluk Oranı (Accuracy of the jth Test Set)

AÖ : Aşağı Örnekleme

$B_{ÇG}$: Bootstrap Çapraz Geçerleme (Bootstrap Cross Validation)

C : Cost Parametresi (Cost Parameter)

ÇADA : Çeyrekler Arası Dağılım Aralığı

CART : Sınıflama ve Regresyon Ağaçları (Classification and Regression Tree)

CSV : Virgülle Ayrılmış Değerler (Comma Separated Values)

ÇG : Çapraz Geçerleme

Ç1 : I. Çeyrek

Ç2 : II. Çeyrek

Ç3 : III. Çeyrek

DN : Doğru Negatif

DP : Doğru Pozitif

DVM : Destek Vektör Makineleri (Support Vector Machines (SVM))

EAA : Eğri Altında Kalan Alan

EKG : Elektrokardiyografi

ERR : Hata Oranı (Error)

EYK : K En Yakın Komşuluğu (K Nearest Neighbor (KNN))

HIV : İnsan İmmün Yetmezlik Virüsü (Human Immunodeficiency Virus)

ID3 : Yinelemeli Bölücü (Iterative Dichotomiser)

LARS : En Küçük Açı Regresyonu (Least Angle Regression)

μ : Ortalama

MAP : En Büyük Sonsal Sınıflandırma (Maximum A Posteriori Classification)

MIT : Massachusetts Teknoloji Enstitüsü

NB : Naive Bayes

r_B : Hata Terimi

σ : **S**tandart Sapma

$\mathbb{R}$: Reel Sayı Kümesi

RTF : Radial Temelli Fonksiyon

RO **:** Rastgele Orman

ROC **:** Alıcı İşlem Karakteristiği Eğrisi (Receiver Operating Characteristic Curve)

ROSE **:** Rastgele Yukarı Örnekleme Örnekleri (Random Over Sampling Examples)

S/D/E **:** Silme / Değişiklik / Ekleme (Deletion/Substitution/Addition D/S/A)

SL **:** Super Learner

SMOTE **:** Sentetik Azınlık Yukarı Örnekleme Tekniği (Syntetic Minority Over-Sampling Technique)

TXT **:** Metin Dosyası (Tab Separated Value)

UCI **:** Makine Öğrenimi Deposu (UC Irvine Machine Learning Repository)

W **:** Super Learner Çıktı Değişkeni

YÖ **:** Yukarı Örnekleme

YSA **:** Yapay Sinir Ağları

YN **:** Yanlış Negatif

YP **:** Yanlış Pozitif

YSABM **:** Yapay Sinir Ağları ve Bulanık Mantık (Artificial Neural Networks and Fuzzy Logic)

$\boldsymbol{\xi}$ **:** Gevşek Değişken

$\boldsymbol{\psi}$ **:** Super Learner Aday Parametre Değeri

$\boldsymbol{\Psi}$ **:** Super Learner Parametresi

İÇİNDEKİLER

1. GİRİŞ

Makine öğrenimi örnek veri seti kullanarak ya da geçmiş deneyimlerden yararlanarak performans kriterlerini optimize eden bilgisayar programıdır. Bir işlemi bilgisayar ile çözmek için bir algoritma gereklidir. Algoritma girdiyi çıktıya dönüştüren komutlar dizisidir (Alpaydın, 2010). Makine öğreniminin amacı, öğrenmeyi insan müdahalesi veya yardımı olmaksızın otomatik olarak yapan öğrenme algoritmalarını tasarlamaktır (Schapire, 2008).

Makine öğrenimi içerisinde farklı yöntemler olmakla birlikte denetimli öğrenme (supervised learning) ve denetimsiz öğrenme (unsupervised learning) yöntemleri bulunmaktadır (Mohri ve ark.,2012). Pratikte en çok kullanılan yöntem, denetimli öğrenmedir. Denetimli öğrenme sınıflama ve regresyon olmak üzere ikiye ayrılmaktadır (Murphy, 2012). Sınıflamada kullanılan başlıca algoritmalar karar ağaçları, K En Yakın Komşuluğu (EYK), Naive Bayes (NB) ve Destek Vektör Makineleri (DVM)'dir (Fumo, 2017; Chakraborty, 2019).

Sınıflama için kullanılan bir diğer yöntem topluluk yöntemleridir. Topluluk yöntemleri bir problemi çözmek için birçok modelin eğitilmesine dayanmaktadır (Zhou, 2012). Birçok yöntem bulunmakla birlikte farklı karar ağaçların birleştirilmesi olan Rastgele Orman (RO) bu çalışmada yer almaktadır. RO, rastgele örnekler oluşturmak için bagging yöntemini kullanmaktadır (Kırçiçek, 2019).

Günden güne sayıları artan ve farklı özellikleri bulunan birçok algoritma bulunmaktadır. Super Learner (SL) algoritması verilere birtakım aday öğreticiler uygulayarak elde edilen tahmin için çapraz geçerlilik riski ile optimal öğreticiyi belirleyen bir topluluk yöntemi algoritmasıdır. Temel öğreticiler arasında bu öğretici asimtotik olarak diğer tüm öğreticiler arasında en iyi performansı vermektedir. Birçok optimal öğreticiyi bir arada kullanarak en iyi performansı elde etmesi en önemli avantajıdır (Sinisi ve ark., 2007; Van der Laan ve ark., 2007).

Makine öğrenimi modellerinin değerlendirilmesinde kullanılan farklı performans ölçüleri vardır. Sınıflama performanslarının ölçülmesi için kullanılan en yaygın ölçüler doğruluk oranı, duyarlılık, seçicilik, eğri altında kalan alandır (EAA) (Zheng, 2015). Bir algoritmayı eğitmek için bir eğitim seti ve değerlendirilmesi için bir test seti gereklidir. Böylece kurulan modelin ne kadar başarılı olduğu saptanır

(Harrington, 2012). Genelde eğitim seti için %80, %70 ve %60 ve test seti için %20, %30 ve %40 oranları kullanılmaktadır. Bu çalışmanın bir amacı da bu oranların etkilerini incelemektir (Chakraborty, 2019).

Son yıllarda makine öğrenimi uygulamaları ile gündelik hayatta sıklıkla karşılaşılmaktadır. Bir arama motoruna girilen bir kelime ile ilgili web uygulamalarının açılması, bir metnin çevirisi, güvenlik şirketlerinin kullanmış oldukları yüz tanıma uygulaması ve bilgisayara gelen e-postaların çöp olup olmaması bunlara örnek olarak verilebilir (Shwartz ve Ben-David, 2014). Sağlık alanında sınıflama çalışmalarına Elekrokardiyografi (EKG) sonuçlarının analizi, radyolojide bir akciğer nodülünün bir göğüs grafisinden otomatik olarak algılanması ve genomik uygulamaları örnek olarak verilebilir (Deo, 2018; Sennaar, 2019). Makine öğrenimin en çok öne çıktığı alan doğru hastalık tanısının konulmasıdır (Sharma, 2019). Makine öğrenimi yönteminin avantajlarından biri alt genotipleri değerlendirmeye dâhil etmesidir. Bu sayede karmaşık yapıda tıbbi veri analizi yapılarak geleceğe yönelik tahmin performansı arttırılacaktır (Sajda, 2006). Son yıllarda ilişkilendirme çalışmalarında çok sayıda hastalıkla ilişkilendirilen veri elde edilmiştir. Dolayısıyla başlangıçta hastalıklara ait sonuçların değerlendirildiği bu veriler ile yüksek tahmin performansı daha sonrasında benzer şekilde etiyolojisi açıklanamayan diğer hastalıkların tahmin performansının değerlendirilmesinde sonuçlar kullanılabilecektir (Hekim ve ark., 2019).

Bu çalışma ile erkek infertilitesi ve peridontitis hastalığı için risk faktörleri makine öğrenimi algoritmalarıyla değerlendirilecektir. Çalışmanın amacı, tıp alanında tanı koymada performansı artırmak için makine öğrenimi yardımıyla genetik verilerin değerlendirilmesi, makine öğrenimi yöntemlerinden uygun algoritmalar yardımıyla analizler yapılarak bu sonuçların SL algoritması ile karşılaştırılmasıdır.

2. GENEL BİLGİLER

2.1. Makine Öğrenimi

2.1.1. Makine Öğrenimi Nedir?

Bilgisayarların icat edilmesi ile birlikte, makinelerin insanlar gibi öğrenip öğrenemeyeceği merak konusu olmuştur. Bilgisayarların insanlar gibi elde ettikleri deneyim ile öğrenmeleri mümkün olursa bu çok faydalı sonuçlar verebilir. Bunun gerçekleşmesi için öncelikle bilgisayarları programlamak gerekmektedir. O zaman elde edilen veriler yardımıyla yeni hastalıklar için en uygun tedavi yöntemleri belirlenebilir veya evlerde enerji kullanımını en aza indirgemek için bir sistem geliştirilebilir (Mitchell, 1997).

Makine öğrenimi örnek veri seti kullanarak ya da geçmiş tecrübelerden yararlanarak performans kriterlerini optimize eden bilgisayar programıdır. Bir problemi bilgisayarda çözmek için bir algoritmaya ihtiyacımız vardır. Algoritma girdiyi çıktıya dönüştüren bir dizi talimatlar bütünüdür (Alpaydın, 2010). Makine öğreniminin amacı, öğrenmeyi insan müdahalesi veya yardımı olmaksızın otomatik olarak yapan öğrenme algoritmalarını tasarlamaktır (Schapire, 2008). Öğrenme algoritmasında girdi eğitim verisi olarak adlandırılır ve bu deneyimi gösterir, çıktı ise elde edilen bilgidir (Shwartz ve Ben-David, 2014).

Öğrenme için başka bir değerli kaynak, algoritmanın oluşturabilmesi için gerekli olan veri miktarıdır (Schapire, 2008). Bilgisayar teknolojisinin gelişmesi ile büyük miktarlarda veri depolama imkanı sunulmaktadır. Son yıllarda veri tabanlarının büyümesi inanılmazdır. Günümüzde modern bir uydu vericisi geçmişteki uydu vericilerin toplamından daha çok veri sağlamaktadır. Bu durum küçük veri kümelerinden büyük veri kümelerine geçişi sağlamıştır (Welling, 2011).

Büyük veri setlerini bilgiye dönüştüren uygulama alanına veri madenciliği denilmektedir. Makine öğrenimi bunun bir alt dalıdır. Veri madenciliğinin başlıca uygulama alanları finans sektöründe bankalar, yatırım ve kredi servisleri, borsa ve sigortadır. Perakende endüstrisi de veri madenciliğin uygulama alanlarındandır. Çünkü satışlardan, müşteri davranışlarına kadar, malların nakliyatı tüketimi gibi konularda bugün büyük veri kümeleri bulunmaktadır (Han ve ark., 2012). Veri madenciliği tıbbi bilişim alanında en önemli metotlar arasında bulunmaktadır. Örneğin sağlık organizasyonlarının yönetimi, epidemiyoloji, hasta bakım ve izleme, büyük ölçekli

görüntü analizlerin gerçekleştirilmesi bu alana dahil olmaktadır (Acharya ve Yu, 2010). Makine öğrenimi yöntemleri geçmişteki veriyi kullanarak yeni veri için en uygun modeli bulmaya çalışırken veri madenciliği ise verinin incelenip içerisinden işe yarayan bilginin çıkarılmasıdır (Diri, 2014).

2.1.2. Makine Öğrenimin Tarihçesi

Makine öğrenimi Turing (1950), yayınlamış olduğu Bilgisayar Makineleri ve Zeka başlıklı makalesinde, "Bilgisayarların kendi zekâları var mı?" diye sorması ile ortaya çıkmıştır. Makalede bu soruya yanıt bulmak için "taklit oyunu" Turing tarafından geliştirilmiştir. Günümüzde bu oyun turing testi olarak bilinmektedir. Oyun erkek (A), kadın (B) ve soru soran (C) arasında yapılmaktadır. C kişisi farklı bir odada durmaktadır ve A ile B'den hangisinin erkek, hangisinin kadın olduğunu soru sorarak bulmaya çalışmaktadır (Namal, 2018).

İlk bilgisayar öğrenme programı 1952 yılında yazılmıştır. Bu program dama oyunudur ve Uluslararası İş Makineleri [International Business Machines (IBM)] programı oyun oynadıkça kendini geliştirmeye başlamıştır (Akbaş, 2017). Yapılan hamlelerden hangileri kazanma stratejisi oluşturmuşsa bu "denetimli öğrenme modu" olarak çalışılmış ve programa dahil edilmiştir (Sheth, 2017).

Rosenblatt (1958), ilk yapay sinir ağı "perceptron" örüntü ve şekil tanıma amacıyla tasarlamıştır. Perceptron bir sinir ağı tipidir. Bir sinir ağı, beyin gibi davranır. Beyin bir ağda birbirine bağlı nöron denilen milyonlarca hücre içermektedir. Nöronlar sayesinde daha kompleks bir problem çözülebilmektedir (Sheth, 2017). Samuel (1959), makine öğrenimi için ilk tanımı yapmıştır. Tanım "Bilgisayarlara açıkca programlanmadan öğrenme olanağı veren çalışma alanıdır" şeklindedir (Awad ve Khanna, 2015).

En yakın komşu algoritması, 1967 yılında yazılmıştır. Bu algoritma bilgisayarların basit kalıpları tanımalarını sağlamıştır. Bu algoritma gezici satış temsilcilerine harita yardımı ile rastgele bir şehirde gezilerine başlayıp kısa bir sürede tüm şehirleri gezmelerine imkan vermiştir (Marr, 2016; Foote, 2019).

Tek katmanlı sinir ağı girdi ve çıktı katmanlarından oluşan doğrusal bir fonksiyondur. Çok katmanlı algılayıcılar ise doğrusal olmayan aktivasyon fonksiyonuna sahip ve birçok nöronun birbirine hiyerarşik olarak bağlandığı bir yapıdır. Bir çok katmanın keşfi ve kullanılması 1960'lı yıllarda sinir ağları alanında gerçekleşmiştir. Bir

katman yerine iki katman kullanıldığında daha fazla işlem gücü elde edilmiştir. Ağlarda katmanların kullanılması ile farklı sinir ağları keşfedilmiştir. Bir çok katmanın kullanılması ile ileri besleme ve geri yayılım sinir ağları geliştirilmiştir. Geri yayılım sinir ağı bir ağın gizli nöron / düğüm katmanlarını yeni durumlara adapte olacak şekilde ayarlamasına izin vermektedir. Çıktının elde edilmesinde hata ile birlikte işlenmektedir ve "hataların geriye doğru yayılmasını" açıklayarak öğrenme amacıyla ağın katmanları arasında geriye doğru dağıtımını yapmaktadır (Foote, 2019).

Yapay sinir ağı (YSA), daha önce geliştirilen tek katmanlı sinir ağından daha karmaşık görevlere yanıt vermek için gizli katmanlara sahiptir. YSA makine öğrenmesi için kullanılan birincil araçtır. Sinir ağları giriş ve çıkış katmanlarını kullanır ve normalde girişi, çıkış katmanının kullanabileceği verilere dönüştürmek için tasarlanmış gizli bir katman (veya katmanlar) içermektedir (Foote, 2019).

Standford Üniversitesi öğrencileri 1979 yılında "Standford Cart" makinesini geliştirmişlerdir. Standford Cart makinesi uzaktan kumandalı TV donanımlı bir mobil robot olup, amacı bir odadaki engelleri tanıyıp onları bağımsız bir şekilde aşmaktır (Sheth, 2017).

Yapay zeka araştırmaları, 1980'lerin başında algoritmalar yerine mantıksal bilgiye dayalı yaklaşımlar kullanmıştır. Bu durum, yapay zeka ve makine öğrenmesi arasında bir ayrılmaya neden olmuştur. Makine öğrenmesi yapay zeka alanında bir eğitim programı olarak kullanılmıştır. Makine öğrenimi endüstrisi ayrı bir alanda yeniden düzenlenmiştir (Foote, 2019).

Schapire (1990), tarafından yayınlanan bir makalede "Boosting" algoritması tanıtılmıştır. Bu algoritma denetimli öğrenmede yanlılığı azaltarak zayıf algoritmaları güçlü algoritmalar şekline getirmektedir (Foote, 2019). Bilgisayar bilimi ve istatistik arasındaki kesişme, yapay zeka alanında olaslılıksal yaklaşımları meydana getirmiştir. Böylece makine öğrenimi bilgi odaklı bir yaklaşımdan, veri odaklı bir yaklaşıma geçmiştir. Büyük çapta veriye sahip olan bilim insanları, büyük miktarlarda veriyi analiz edip öğrenebilen akıllı sistemler kurmaya başlamıştır. Ayrıca bilgisayar alanında donanım ve teknoloji de gelişme göstermiştir (Provalis, 2017).

2.1.3. Makine Öğrenimi Uygulama Alanları

Son yıllarda gündelik hayatta sıklıkla makine öğrenimi uygulama alanları ile karşılaşılmaktadır. Bir arama motoruna girilen bir kelime ile ilgili web sayfaları

uygulamalarının açılması, bir metnin otomatik olarak istediğimiz dile çevirme işlemleri ya da verilen bir fotoğrafın kime ait olduğunu belirleyebilen yüz tanıma programlarıdır (Smola ve Vishwanathan, 2008). İnsan yüzü benzersiz değildir ve sayısız faktör ile tanımlama algoritmalar yardımıyla yapılmakta olup, sosyal ağlarda yüz etiketlemenin temelini oluşturmaktadır (Das ve ark, 2015).

Makine öğrenimine ilişkin en çok bilinen uygulama bilgisayara (makine) gelen e-postaların çöp olup olmamasıdır. Buna ilişkin süreçte makine öncelikle kullanıcısının, önceki e-postalarından spam olarak etiketlediklerini belirleyecektir. Yani etiketleme bu spam/ spam değil şeklindedir. Yeni bir e-posta geldiğinde makine daha önceki e-postalarla karşılaştıracaktır. Bunlarla eşleşenleri çöpe atacaktır spam olmayanları da gelen kutusuna gönderecektir. Ancak burada öğrenme hatırlamaya dayalıdır. Bu durumda daha önce karşılaşmadığı durumlar için yararsız olacaktır. Başarılı bir öğrenme, bireysel durumlardan genellemeler çıkarılan süreci göstermektedir. Spam için daha önceki e-postalardan spam olarak nitelendirdikleri arasından kelimeler belirler. Bu durumda sistem daha önce karşılaşmadığı durumlar için doğru olarak tahmin edebilir (Shwartz ve Ben-David, 2014).

Tavsiye sistemleri yapılan önceki aramalara göre ürün ve hizmetleri öneren sistemlerdir. Makine öğrenimin en sık kullanılan uygulama alanlarındandır. Bu tür sistemler Amazon ve Netflix dahil olmak üzere çeşitli çevrimiçi perakende ortamlarında kullanılmaktadır. Ayrıca, bir kullanıcının ilgi alanlarına karşılık gelen haberler gibi sosyal medya kullanıcılarına belirli içerik türlerini tanıtmak için de kullanılabilir (Shutz, 2017).

Müşteri ilişkileri yönetimi sistemleri, iş zekası analitiği, insan kaynakları sistemleri, sürücüsüz araçlar, etkili çalışanların özelliklerini tanımlamak için öğrenme modelleri kurgulama ve açık pozisyonlara en iyi adayları bulmak gibi alanlarda da makine öğrenimi algoritmaları geliştirilmektedir (Oluk, 2019).

2.1.4. Sağlık Alanında Makine Öğrenimi

Sağlık alanında daha fazla verinin toplanması ve depolanması, yeni hastalıkların bulunması, yeni tanı yöntemlerin geliştirilmesi, aynı zamanda veri tiplerinin karmaşıklığı makine öğrenimi yöntemlerine ihtiyacı arttırmıştır. Çünkü sistemlerde toplanan gigabaytlarca bilgi, çözümlenip gelecekle ile ilgili tahmin yapmaya yarayacak bilgi haline dönüşerek kullanılabilecektir. Tüm bu durumlarda

klinisyenlerin karşılaşmış oldukları yüksek boyutlu karmaşık veri setlerinin yorumlanması için makine öğrenimi yeni araçlar sağlamaktadır (Sajda, 2006).

Makine öğrenimi ve istatistiksel örüntü tanıma biomedikal alana büyük katkı sağlamaktadır. Çünkü hastalık tanısında duyarlılık ve seçicilik arttırmanın yanı sıra karar verme sürecini daha objektif hale getirmektedir (Sajda, 2006). Genomik ve proteomik düzeyde elde edilen bilgiler de hastalık tanı ve tedavisinde özellikle kanser teşhisinde çok önemli katkılar sağlamaktadır (Kourou ve ark., 2015).

Makine öğrenimin genomik uygulamaları iki kategoriye ayrılmakta olup genom diziliminde, araştırmacılar, yüksek boyutlu genetik veri setleri içindeki kalıpları tanımlamak için makine öğrenmesini kullanmaktadır. Bu modeller daha sonra bireyin belirli hastalıklar geliştirme olasılığını tahmin etmeye yardımcı olabilecek veya potansiyel tedavilerin tasarımını bilgilendirmeye yardımcı olabilecek bilgisayar modellerine dönüştürülmektedir. Hastalık teşhisinde genetik veriler çok önemli bir yere sahiptir. Tüketicinin ihtiyaçlarına göre genetik bilgisi, bir bireyin genlerinin vücut ağırlıklarını nasıl etkileyebileceği gibi genetik bilgilerin yorumlanmasında daha fazla derinlik kazanmak için makine öğrenimini kullanarak çalışmaktadır (Sennaar, 2019).

Sağlık alanındaki sınıflama çalışmalarına EKG sonuçlarının analizi, radyolojide bir akciğer nodülünün ve bir göğüs grafisinin otomatik olarak algılanması örnek olarak verilebilir. Yapılan bu çalışmalarda bir uzmanın elde edeceğe sonuçlara benzer doğruluk oranı elde edilmektedir (Deo, 2018).

Tıbbi bakım doğru tanı ile başlamaktadır. Makine öğrenimin en çok öne çıktığı alan doğru hastalık tanısının konulmasıdır (Sharma, 2019). Örneğin erkek infertilitesine ilişkin doğru tanı koymaya yönelik sınıflama çalışmaları farklı makine öğrenimi algoritmaları kullanarak yapılmıştır (Palechor ve ark., 2016). Bir başka çalışmada semen kalitesini ölçmek için topluluk yöntemlerinden meta sınıflama teknikleri kullanılmıştır (Priya, 2017). Başka bir çalışmada sınıf dengesizliği problemi olduğunda semen kalitesini ölçmek için farklı makine öğrenimi algoritmaları ile değerlendirilmiştir (Wang ve ark, 2014). Bir çalışmada ise erkek infertilitesi tahmin etmek için özellik seçimi yöntemleri ve sınıf dengesizliği analizinde Sentetik Azınlık Yukarı Örnekleme Tekniği [Syntetic Minority Over- Sampling Technique (SMOTE)] yöntemi kullanarak farklı makine öğrenimi algoritmaları kullanarak sonuçlar karşılaştırılmıştır (Karlık, 2016).

Tıp ve sağlık alanında uzaktan tanı ve görüntü analizi için de makine öğrenimini kullanılmaktadır. Çünkü geleneksel yöntemler çok zaman alıcıdır. Massachusetts Teknoloji Enstitüsü [Massachusetts Institute of Technology (MIT)] araştırma ekibi bir insandan 1000 kat hızlı görüntü analizi yapan bir makine öğrenimi algoritması geliştirmiştir. Günümüzde bazı ameliyatlar mini robotlar kullanılarak gerçekleştrilmektedir. Yapılan araştırmalarda robotların kullanımı ile gerçekleştirilen ortopedi ameliyatlarında komplikasyonlar 5 kat azalmıştır. Ayrıca sanal hemşirelerin ve hastane yönetim işlerinin makine öğrenimi kullanarak yapılması sonucunda milyonlarca dolar kar elde edilebilecektir (Sharma, 2019).

Uygulama alanı ne olursa olsun, çok miktardaki verinin analiz edilerek gelecek ile ilgili tahminlerde bulunması ve karar almada yardımcı olması ile makine öğrenmesi yöntemlerinin her geçen gün önemi artmaktadır (Diri, 2014).

2.2. Makine Öğrenimi Yöntemleri

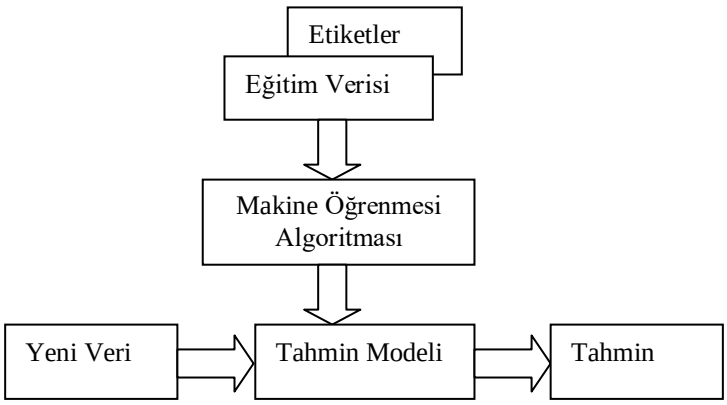
Makine öğrenimi, öğrenme yöntemine göre denetimli öğrenme (supervised learning), denetimsiz öğrenme (unsupervised learning), yarı denetimli öğrenme (semi-supervised learning), transdüktif çıkarsama (transductive inference), çevrim içi öğrenme (on-line learning), takviyeli öğrenme (reinforcement learning) ve aktif öğrenme (active learning) şeklinde sınıflanabilir (Mohri ve ark, 2012; Zhang, 2010). Bununla birlikte sıklıkla denetimli ve denetimsiz öğrenme kullanılmaktadır (Murphy, 2012).

2.2.1. Denetimli Öğrenme

Bu yöntem, eğitim verisi olarak bir dizi etiketlenmiş örneğe sahiptir ve daha önce görmediği noktalar için tahminlerde bulunur. Etiketlenmiş veri, örneklere atanan değerler ya da kategorilerdir. Örneğin ikili sınıflama problemlerinde kategoriler bir e-postanın spam/spam olmaması şeklinde olabilir (Mohri ve ark., 2012).

Makine öğrenme algoritmaları tarafından kullanılan herhangi bir veri kümesindeki örnekler, aynı özellik kümesini kullanarak temsil edilir. Bu özellikler sürekli, kategorik ya da ikili olabilir. Örnekler bilinen etiketlerle verilirse, öğrenme *denetimli öğrenme* olarak adlandırılır. Amaç, sınıf etiketlerinin özelliklerine göre verilerin dağılımına ilişkin özlü bir model oluşturmaktır. Ortaya çıkan sınıflayıcı, daha sonra, tahmin edici özelliklerin değerlerinin bilindiği test örneklerine sınıf etiketleri atamak için kullanılır, ancak sınıf etiketinin değeri bilinmemektedir (Kotsiantis, 2007). Burada *denetimli* kelimesi istenen çıktı değişkenlerinin etiketlerinin bilindiğini

göstermektedir. Aşağıda Şekil 1'de gösterilen model bir denetimli öğrenme modelinin sürecini göstermektedir (Raschka, 2015).



Şekil 1. Denetimli öğrenme modeli süreci (Raschka, 2015'den uyarlanmıştır)

Makine öğrenimi yöntemleri içerisinde en popüler ve kullanışlı yöntemdir. Bu yöntemin sınıflama ve regresyon olmak üzere iki türü vardır (Murphy, 2012).

Sınıflama

Tahmin için en sık kullanılan bu denetimli öğrenme yönteminde, bir örneğin hangi kategoriye ait olduğunun tespit edilmesidir. Bu, *sınıflama* olarak bilinmektedir. Aynı zamanda sayısal çıktı tahmini için de denetimli öğrenme yöntemi kullanılabilir, bu da regresyon olarak bilinmektedir (Lantz, 2015).

Sınıflama için örnekler bir e-postanın gereksiz olup olmaması, bir kişinin kanser olup olmaması, bir futbol takımının kazanıp kazanamaması şeklinde verilebilir (Lantz, 2015). Bu tip sınıflamalar ikili sınıflamayı göstermektedir. Çoklu sınıf problemleri ise bir girdinin çoklu kategorilerden hangisine ait olduğunun belirlenmeye çalışılmasıdır. Kaydedilmiş bir ses sinyali verildiğinde bir konuşmacının kimliğini belirlemek, bir resmin kime ait olduğunu belirlemek örnek olarak verilebilir (Hertzmann ve Fleet, 2010).

Bir sınıflandırıcı için eğitim kümesindeki değerler girdi olarak isimlendirilir ve her veri için etiketler bulunmaktadır. Değerler özellikler kümesini tanımlamaktadır. Amaç bu özelliklere göre her sınıf için bir model oluşturmaktır. Bu model daha sonra gelecekteki değerleri sınıflandırmak için kullanılır (Patil ve ark., 2010).

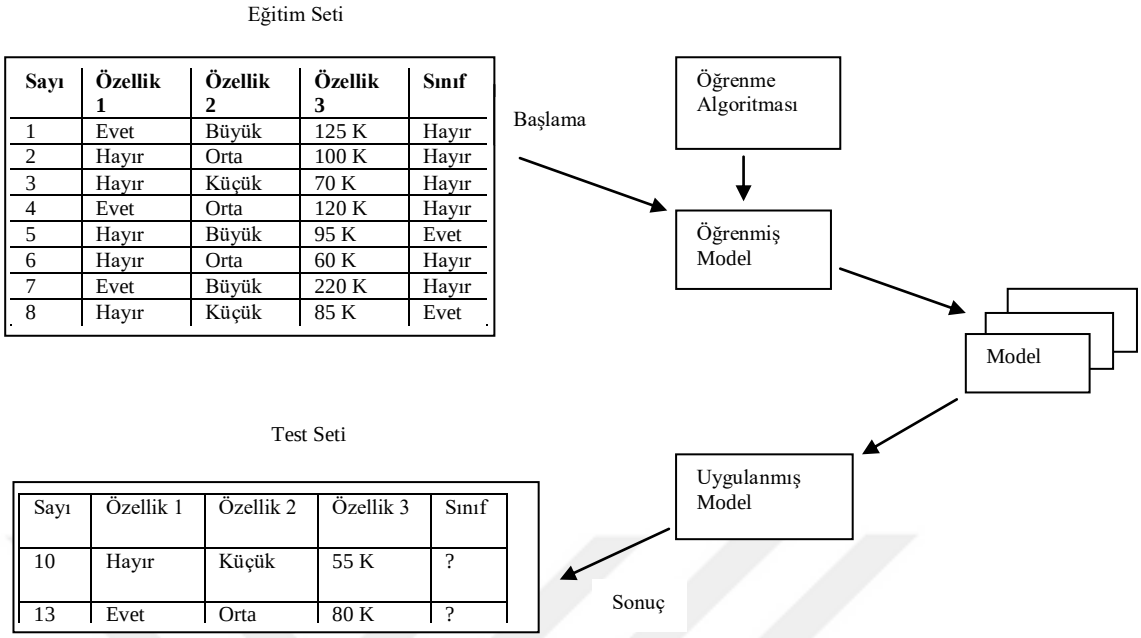
Elimizdeki bilgilere dayanarak geleceğe yönelik verilecek kararlar ya da yapılan tahminler sınıflama sürecini göstermektedir. Örneğin banka kredisi verilmesi

aşamasında kişilerin finansal bilgilerine ve kişisel özelliklerine bakılarak kararların verilmesi veya hastalık hakkında teşhis ve tedavi işlemleri sürecinin belirlenmesi için test sonuçlarına bakılması. Kısaca bilim, sanayi, finans alanında meydana gelen bu gibi problemler sınıflama problemi olarak bilinmektedir (Michie ve ark., 1994).

Tıbbi verilerinin çözümü için, özellikle hastalık teşhisi konusunda, doktorların verecekleri kararlara destek olabilecek önemli çalışmalar yapılmaktadır. Sınıflama algoritmaları geçmişe yönelik hasta verilerini kullanarak kendisini eğitir. Eğitimi tamamlanmış algoritma daha sonra olası hastalar veya hastalıklar için tahminde bulunur. Hastalık teşhisinde hekimlere yardımcı olan sınıflandırma algoritmaları temelli bu sistemlere *karar destek sistemleri* denir. Karar destek sistemlerini oluşturan sınıflandırma algoritmaları konusunda günümüze kadar birçok çalışma gerçekleştirilmiştir (Karakoyun ve Hacıbeyoğlu, 2014). Tıbbi verilerin sınıflandırılması ile ilgili farklı çalışmalar mevcuttur. Makine Öğrenimi Deposu [UC Irvine Machine Learning Repository (UCI)] elde edilen Wisconsin Meme Kanseri Teşhisi veri seti farklı algoritmalar kullanarak karşılaştırılmıştır. Sonuç olarak DVM algoritması kullanarak %97,72 başarı oranına ulaştığı görülmüştür (Aličković ve Subaşı, 2011). Bir başka çalışmada ise 15 farklı tıbbi veri kullanılarak 5 farklı sınıflandırma algoritmaları karşılaştırılmıştır. Bu çalışmada NB, algoritması ile en iyi sonuç elde edilmiştir (Al-Aidaros ve ark., 2012).

Huang ve ark. (2003), NB, karar ağaçları ve DVM kullanarak çeşitli veri kümeleri üzerinde sınıflandırmalar yapmıştır. DVM algoritmasının, uygulanan veri kümeleri için NB ve karar ağaçları algoritmalarından daha başarılı olduğu ancak yapılan istatistiksel testler sonucunda aradaki farkın kayda değer oranlarda olmadığı bulunmuştur (Huang ve ark., 2003).

Öğrenme için kullanılan veri setine *eğitim seti* denilmektedir. Öğrenme algoritması tarafından oluşturulan model girdi verisini iyi temsil etmelidir. Böylece daha önce karşılaşmadığı veriler için etiketleri doğru tahmin edebilir. İlk önce eğitim seti aşağıdaki Şekil 2'de gibi oluşturulur, görüldüğü üzere test setinde etiketler bulunmamaktadır (Tan ve ark., 2004).



Şekil 2. Sınıflama modeli oluşturmak için genel yaklaşımı (Tan ve ark., 2004'den uyarlanmıştır)

Bir model, öğrenme algoritması tarafından eğitim verilerinden öğrendikten ve oluşturulduktan sonra, model doğruluğunu değerlendirmek için bir dizi test verileri (veya görünmeyen veriler) kullanılarak değerlendirilir. Test veri setindeki veriler öğrenme aşamasında kullanılmaz. Test veri setinde de etiketler bulunmaktadır. Bu nedenle test verileri, öğrenilen modelin doğruluğunu değerlendirmek için kullanılabilir, böylece her bir test durumu için tahmin edilen sınıfın model tarafından test durumunun gerçek sınıfıyla aynı olup olmadığı kontrol edilebilir (Liu, 2011).

Test veri setindeki sınıflama modelinin doğruluğu hesaplanır. Doğruluk oranı test setindeki doğru sınıflanmış örneklerin tüm veri setindeki örneklere bölümüdür. Eğer eğitim seti bu doğruluk oranını elde etmek için kullanılırsa yanlış bir sonuca ulaşılır. Çünkü bu durumda sınıflayıcı veriye aşırı uyum sağlar (yani eğitim verisinde tüm veride olmayan bazı anomaliler eğitim setinde olabilir). Bunu önlemek için test seti kullanılır (Han ve ark., 2012). Bazı araştırmacılar hata oranını (1- doğruluk oranı) da kullanır. Farklı sınıflandırıcılarımız varsa en yüksek doğruluk oranını veren model seçilmektedir (Liu, 2011).

Sınıflamada kullanılan başlıca algoritmalar karar ağaçları, EYK, NB ve DVM algoritmalarıdır (Fumo, 2017; Chakraborty, 2019).

Karar Ağaçları

Quinlan (1986), makine öğrenimi dalında çalışarak karar ağacı algoritmasını geliştirmiştir. Bu algoritmaya Yinelemeli Bölücü [Iterative Dichotomiser (ID3)] adını vermiştir. Sebebi ise bu programın, iki farklı kategoriden birine sınıflandırılıncaya kadar verilerin kayıt kümelerini tekrar tekrar bölerek karar ağaçları oluşturduğu gerçeğinden yola çıkmasıdır. Diğer çeşitlerinde ikiden fazla kategori ile çalışabilmektedir (Nilson, 2009).

Quinlan (1993), daha sonra C4.5 (ID3 bir ileri aşaması) geliştirmiştir. C4.5 algoritması, ID3 algoritmasının üstesinden gelemediği sorunları çözmek amacıyla geliştirilmiştir. Bu yeni geliştirilen algoritma denetimli algoritmaların karşılaştırıldığı bir kriter haline gelmiştir (Han ve ark., 2012). Breiman ve ark. (1987), Sınıflama ve Regresyon Ağaçları [Classification and Regression Tree (CART)] kitabını yayınlamışlardır. Bu kitapta ikili karar ağaçların oluşturulması anlatılmıştır. ID3 ve CART birbirinden bağımsız olarak aynı zamanda, karar ağacı eğitim setlerine bağlı olarak benzer bir yaklaşımla oluşturulmuştur. ID3, C4.5 ve CART karar ağaçları yukarıdan aşağıya yinelemeli bölme biçiminde oluşturulan (yani geriye doğru olmayan) bir yaklaşımla çalışır (Han ve ark., 2012).

K En Yakın Komşuluğu

Verinin dağılımı hakkında bir bilgi bulunmadığı zaman kullanılabilecek en temel ve basit sınıflandırma yöntemlerinden birisidir. Bu algoritma diskriminant analizi yapılmak istendiğinde ancak olasılık yoğunluk fonksiyonların tahmin edilmesi güvenilir ya da elde edilmesi zor olduğu durumlar için geliştirilmiştir (Peterson, 2009). Fix ve Hodges (1951), tarafından örüntü sınıflandırma için geliştirilen parametrik olmayan bir metottur. Daha sonra Cover ve Hart (1967), düzenlemeler yaparak geliştirilmiştir. Bu düzenlemeler sınıflanmamış örnek noktalarını daha önce sınıflanmış en yakın örneklere göre atanmasını içeren bir karar kuralını içermektedir (Song ve ark., 2007).

Bu yeni yöntem ile ilgili yıllar içerisinde bir takım düzenlemeler yapılmıştır. Bunlar içerisinde uzaklık hesaplamalarında farklı ağırlıklandırmaların kullanılması, esnek hesaplama yöntemleri ve bulanık yöntemler yer almaktadır (Peterson, 2009).

Naive Bayes

Bu algoritma olasılığa dayanır ve model hakkındaki belirsizlikleri sonuçların olasılıklarını belirleyerek elde etmektedir (Vijaykumar, 2014). Bir veri seti için en iyi hipotezi belirlemenin en kolay yolu önceki bilgilere dayanarak sorunu çözmektir. Bayes teoremi önceki bilgilere dayanarak hipotez için olasılık değerlerini hesaplamak için bir yol sağlar. Makine öğreniminde bu teknik "naive" adını alır bunun sebebi ise her hipotez için olasılıkların kolay hesaplanması için basitleştirilmiş olmasıdır. Bu basitleştirme ile her değişken için birleşik olasılıkları yerine onları bağımsız olarak kabul ederek ayrı ayrı olasılıkları hesaplanmaktadır. Bağımsızlık varsayımı gerçek hayat verilerinde gerçeği yansıtmasa bile uygulamalarda bu varsayım ile oldukça iyi sonuçlar elde edilmektedir (Brownlee, 2016).

Destek Vektör Makineleri

Destek vektör makineleri 1964-1965 yılları arasında ilk olarak makalelerde yer almasına rağmen 1992 yılına kadar kullanılmamasının nedeni teorik olarak iyi bir alt yapıya sahip olmasına rağmen pratik uygulamalarda iyi sonuçlar vermeyeceği düşüncesiydi. Ancak sayısal tanımlama, bilgisayar vizyonu ve metin kategorizasyonu pratik uygulamalarında iyi sonuçlar elde edilmiştir. Bunun üzerine istatistiksel modeller ile yapılan karşılaştırmalarda bu durum ispatlanmıştır (Kecman, 2005).

DVM denetimli öğrenme yöntemleri içerisinde yer alan sınıflama ve regresyon yapmak için kullanılan bir algoritmadır. Cortes ve Vapnik (1995), bu algoritmayı bulmuştur. DVM sınıflama yaparken her iki kategoriden birine ait olarak işaretlenmiş bir dizi eğitim verisi verildiğinde, bu algoritma gelen yeni bir örneğin her iki kategoriden birine ait olduğunu tahmin eden bir model oluşturur (Wikibooks, 2018).

Topluluk yöntemleri bir problemi çözmek için birçok modelin eğitilmesine dayanmaktadır (Zhou, 2012). Son yıllarda hem makine öğrenimi alanında hem de bilgi işlem alanında çoklu sınıflayıcı sistemler yani topluluk sistemleri büyük ilgi çekmektedir. Bunun sebebi geniş bir alanda etkili olmaları ve gerçek dünya uygulamalarında başarılı sonuçlar vermeleridir. Bu sistemler, değişkenliğin azaltılarak doğruluğun arttırılması amacıyla geliştirilmiştir (Zhang ve Ma, 2012). Topluluk yöntemleri algoritmaları bagging, boosting-adaboost-gradient boosting, yığılım ve farklı karar ağaçların birleştirilmesi olan rastgele orman'dır (Kırçiçek, 2019). Bu çalışmada RO algoritması incelenmiştir.

Rastgele Orman

Ho (1995), karar ağaçlarının geleneksel yöntemlerle çözemediği temel problemi önermiş olduğu bir metot ile çözmüştür. Önerilen yöntemde eğitim veri setini doğruluğunu optimize etmek için oblik karar ağaçları kullanmaktadır. Rastgele alt kümelerde çoklu karar ağaçları oluşturmayı temel alan bir yöntemdir (Ho, 1995). Amit ve Geman (1997), şekil tanıma yaklaşımını önermişlerdir. Gerçekte özelliklerin sonsuz sayıya sahip olması sebebi ile tüm özelliklerin bulunduğu kümede hangi özelliğin daha bilgi verici olduğunun belirlenmesi mümkün olmadığı sonucuna varılmıştır. Özelliklerin sayısı sebebi ile standart karar ağaçların belirli uzunluktaki özellik vektörü ile çözüme ulaşması mümkün değildir. Başka bir yaklaşım da her düğümde bir küçük özellik seti rasgele oluşturularak bir çok ağaç oluşturulur. Son düğüm noktaları şekil sınıfına göre önsel dağılımların tahminlerini içerirler. Şekile bakılarak ve geriye kalan dağılımlar birleştirilerek görüntü sınıflandırılabilir (Fawagreh ve ark., 2014).

Ho (1998), aşırı uyum ve maksimum doğruluk oranı arasındaki ikilem için bir çözüm önermiştir. Buna göre eğitim veri seti maksimum doğruluk oranı için karar ağacına dayalı bir model oluşturulur aynı zamanda da karmaşıklık arttıkça bir genelleme doğruluk oranı elde edilir. Sınıflayıcı özellik vektörünün tamamlayıcısı olan pseudo-rastgele seçim yöntemi ile bir çok ağaçtan oluşur. Böylece rastgele alt kümeler için ağaçlar oluşturulur. Bu yöntem tekil ağaçların kullandığı yönteme göre üstünlük sağlamaktadır (Fawagreh ve ark., 2014).

Breiman (2001), tek bir karar ağacı üretmek yerine, her biri farklı eğitim kümelerinde eğitilmiş olan aynı dağılımlı çok sayıda karar ağaçlarını önermiştir. Bu yöntemde karar ağaçları için her düğümde rastgele değişken seçme yöntemi ile bagging yöntemi birleştirilmiştir. Her iki yöntemin aynı anda birleştirmesi ve her türlü problem için çok iyi çalışması sonucu makine öğreniminde en etkili algoritmalar arasına girmiştir (Louppe, 2015).

Topluluk yöntemleri içerisinde yer alan bir diğer algoritma yığılım algoritmasıdır. Wolpert (1992), yığılım genelleştirmesi adını verdiği bir ya da birden fazla genelleştiricinin hata oranını minimize eden bir metodu tanıtmıştır. Yığılım genelleştirmesi, belirli bir öğrenme setine göre genelleştiricilerin yanlılığını azaltarak (hatanın düzeltilmesi) çalışmaktadır. Tek bir genelleştirici kullanıldığında genelleştiricinin hatasını tahmin etmektedir. Birden fazla genelleştirici kullanıldığında ise çapraz- geçerlemenin farklı bir uyarlamasıdır (Wolpert, 1992).

Breiman (1996), regresyon yığını için farklı tahmin edicilerin doğrusal kombinasyonlarını oluşturarak tahmin doğruluğunu geliştirmek için bir metot geliştirmiştir. Bu metotta regresyon katsayılarını belirlemek için negatif olmayan kısıtlar altında çapraz- geçerleme verilerini ve en küçük kareleri kullanmaktadır (Breiman, 1996).

Van der Laan ve Dudoit (2003), çapraz- geçerleme ile parametre belirlenmesi, tahmini, seçimi ve performans değerlendirmesi için birleştirilmiş kayıp-temelli tahmin metodolojisini önermektedir. Bu yaklaşımda ilgilenilen parametre uygun bir kayıp fonksiyon için risk minimizeri olarak tanımlanmaktadır ve aday tahmin ediciler bu kayıp fonksiyon kullanılarak üretilir. Adaylar içerisinde en uygun tahmin edicinin belirlenmesi için çapraz- geçerleme uygulanmaktadır. Böylece elde edilen tahmin edicinin genel performansı değerlendirilir. Makalede mikroarray gen ekspresyonu ölçümleri kullanarak biyolojik ve klinik sonuçların tahmini bulunmaktadır (Van der Laan ve Dudoit, 2003).

Super Learner Algoritması

Van der Laan ve Dudoit (2003), Van der Laan ve ark. (2006) ve Sinisi ve ark. (2007), aday öğreticiler arasından seçimi çapraz-geçerleme (ÇG) kullanarak belirlenen SL algoritmasını önermişlerdir. Elde edilen teorik sonuçlara göre SL algoritması diğer aday öğreticiler kadar ya da daha iyi performans gösterdiği elde edilmiştir. Benzetim veri seti oluşturarak algoritma performansı ispatlanmıştır. Ayrıca virüs genotipine dayalı İnsan immün yetmezlik virüsüne [(Human Immunodefecency Virus (HIV)] ait fenotipik anti retroviral noktalar tahmin edilmiştir (Sinisi ve ark, 2007).

Van der Laan ve ark. (2007), bir dizi aday öğreticiler grubu arasından parametrik, yarı-parametrik veya veri özelliklerine adapte olan v-kat ÇG kullanarak tahmin için SL algoritmasını önermektedir. Elde edilen bu SL algoritması bir dizi aday öğreticiler grubunun (yarı) parametrik formu için asimtotik olarak oracle seçicisi ile aynı performans gösterdiği ispat edilmiştir. Bu yaklaşımda aday öğreticiler arasından ÇG riskini minimize edecek en küçük kareler regresyon problemi kullanarak seçim yapılmaktadır. Bu yöntemde parametrik ya da parametrik olmayan regresyon metodu ile ÇG riski için aşırı öğrenme önlenmiş olacaktır. Simülasyon çalışmaları ve iki gerçek veri seti uygulamaları yapılmıştır. Sonuçta bu yeni yöntem diğer yöntemlere göre daha üstün performans sergilemiştir (Van der Laan ve ark., 2007).

Polley ve Van der Laan (2010), SL algoritması tahmin için kullanmıştır. SL algoritmasının optimal birleşimini bulmak için oluşturulmuştur. Bu makalede pratik uygulamaların yanı sıra sınırlı örneklem performansı tahmin sonuçları da elde edilmiştir (Polley ve Laan, 2010).

Regresyon

Tahmin için kullanılan bir diğer denetimli öğrenme yöntemi regresyondur. Burada istenen çıktı değeri sürekli değişkenlerden oluşuyorsa, bu öğrenme problemine regresyon denilmektedir. Regresyon için tipik örnekler borsa piyasası ile ilgili tahminler ve fiziksel bir ölçümün (basınç, sıcaklık vb.) tahmin edilmesidir (Camastra ve Vinciarelli, 2007). Regresyon veri setleri etiketli verilerden ve sayısal çıktı değişkenlerinden oluşmaktadır. Başka bir ifade ile algoritmayı denetlemek için her gözleme karşılık bir çıktı değerine sahiptir (Elitedatascience, 2019).

Regresyon algoritmaları için esas amaç kesikli ya da sürekli değişkenleri tahmin etmektir. Bazı durumlarda tahmin değerleri nitelikler arasındaki doğrusal ilişkiyi açıklamak amacıyla kullanılabilir. Örneğin satışları artarsa ürün alımı sonucu ayrılan bütçenin arttırılması gerekir (Polamuri, 2014). Girdi verileri ile hedef çıktılar arasında eğitim verisi kullanarak fonksiyonel ilişki belirlenir. Hedef çıktılar bilindiğinde fonksiyona ait parametreler öğrenilir. Böylece makineler hedef çıktılar bilinmediği zamanda da tahmin yapabilir (Turiplatform, 2017).

2.2.2. Denetimsiz Öğrenme

Denetimli öğrenmede amaç, girdinin çıktıyla eşleşmesinin bir denetleyici tarafından yapılmasıdır. Denetimsiz öğrenme de ise böyle bir denetmen yoktur ve sadece girdi verisi mevcuttur. Amaç girdideki farklılıkları saptamaktır. Girdi verisi için belirli bir yapı oluşmaktadır ve bu yapı belirlenmeye çalışılır (Abdallh ve ark., 2016).

Kısaca girdi verisi etiketli değildir ve bilinen bir sonucu yoktur. Bir model, girdi verilerinde bulunan yapıları çıkarmak için hazırlanmıştır (Brownlee, 2016).

Denetimsiz öğrenmede makineye girdiler x_1, x_2, …x_t gelir fakat denetimli öğrenmedeki gibi hedef çıktılar yoktur. Bir makineden herhangi bir geri bildirim almadan öğrenme gerçekleştirilebilir. Bununla birlikte makinenin amacı, karar vermede kullanılabilecek girdilerin temsillerini oluşturmak, gelecekteki girdileri öngörmek, girdileri başka bir makineye verimli bir şekilde iletmek gibi sunulabilir. Bu verilerden

herhangi bir desen oluşturulmasıdır. Denetimsiz öğrenme için örnekler ise kümeleme ve boyut indirgemedir (Abdallh ve ark., 2016).

2.3. Makine Öğrenimi Süreci

2.3.1. Makine Öğrenimi Adımları

Makine öğreniminin amacı veriden bilgi elde etmektir. Bu sebeple "veriler" makine öğrenimi için en önemli kaynaktır. Literatürde önerilen makine öğrenimi adımları hep bu veriler kullanılarak değerlendirilmiştir (Van Rijmenam, 2019). Örnek olarak Nath (2016), Brownlee (2013) ve Yufeng (2017) önermiş olduğu adımlar gösterilebilir. Nath (2016), tarafından önerilen adımlar da verinin toplanması, veri hazırlanması, modelin eğitilmesi, modelin değerlendirilmesi ve performans iyileştirmesi olarak 5 adımdan bahsedilmektedir.

Kullanılan algoritmaya bakılmaksızın makine öğrenimi uygulamaları için bu adımlar kullanılmaktadır. Bu adımlar ile amaç veri setlerini analiz için uygun hale getirmektir (Merih, 2017).

2.3.2. Dengesiz Veri Problemi

Büyük çoğunluktaki öğrenme algoritmaları eğitim setlerinin dengeli dağıldığını varsaymaktadır. Ancak bazı alanlarda elde edilen veriler için bir sınıf çok sayıda örnek ile temsil edildiği halde diğer sınıfta sadece birkaç örnek bulunmaktadır (Japkowicz, 2000). Az sayıda örneğin temsil ettiği sınıf azınlık sınıf ve çok sayıda örneğin bulunduğu sınıf ise çoğunluk sınıf olarak bilinmektedir (Mountassir ve ark., 2012). Bu veri setinde sınıfların eşit olmadığını yani dengesiz olduğunu gösterir. Sınıf dengesizliği bu açıdan birçok alanda karşılaşılan bir problemdir. Bu problem göz ardı edildiğinde ise standart öğrenme yöntemleri ile performansta bir azalma olduğu saptanmıştır (Japkowicz, 2000).

Dengesiz veri setinde, sınıflandırıcılar eğitilirken örnek sayısının daha fazla olduğu sınıf setine doğru bir eğilim gösterir. Yani sınıflandırıcıda örnek sayısı çoğunlukta olan sınıf ile eğitileceği için bir önyargı oluşmaktadır. Ayrıca daha az olan sınıf ile sınıflandırıcı tam olarak eğitemediği için sonraki aşamalarda başarılı sınıflandırma yapamamaktadır (Bulut, 2016).

Genelde makine öğreniminde kullanılan algoritmalar her sınıftaki örnek sayıları eşit olduğunda iyi çalışmaktadır. Bunun sebebi algoritmalar doğruluğu

maksimum yapmak ve hatayı azaltmak için oluşturulmuştur. Bu sebeple değerlendirme aşamasında performans ölçüsü olarak doğruluk oranı yerine farklı bir ölçünün kullanılması önerilmektedir (Boyle, 2019).

Daha önceki çalışmalarda dengesiz veriler için farklı performans metrikleri kullanılmıştır. Akosa (2017) ve Jeni ve ark. (2013) yapmış oldukları çalışmalarda ortak kullandıkları performans ölçümleri doğruluk oranı, kappa, EAA ve F1 skorudur (Akosa, 2017; Jeni ve ark.,2013). Bu çalışmada karşılaştırmalar EAA ile yapıldı.

Dengesiz veri problemini çözmek için literatürde kullanılan iki yaklaşım vardır. İlk yaklaşım sınıflayıcıyı değiştirmek için kullanılan maliyet- duyarlılık analizidir. İkinci yaklaşımda ise veri setinin kendisi değiştirilir. Bunun için kullanılan iki yöntem Aşağı Örnekleme (AÖ) ve Yukarı Örneklemedir (YÖ). AÖ yönteminde eğitim setinde bulunan çoğunluk sınıf üyelik sayısı azaltılırken, YÖ ise azınlık sınıf üyelik sayıları çoğaltılmaktadır (Mountassir ve ark., 2012).

Chawla (2002) SMOTE yöntemini önermiştir. Eğitim setindeki her azınlık gözlem için özellik uzayında en yakın komşularından birine bağlayan rastgele seçilmiş noktalar seçilerek yapay örnekler üretilir (Menardi ve Torelli, 2010). Lunardon ve ark. (2014)'te Rastgele Yukarı Örnekleme Örnekleri [Random Over Sampling Examples (ROSE)] önermiştir. Burada yapay olarak düzleştirilmiş boosted yöntemine göre veriler azınlık sınıfına göre üretilir (Prasad ve Rao, 2017).

Bu çalışmanın amacı, tıp alanında tanı koymada performansı artırmak için makine öğrenimi yardımıyla genetik verilerin değerlendirilmesi, makine öğrenimi yöntemlerinden uygun algoritmalar yardımıyla analizler yapılarak bu sonuçların SL algoritması ile karşılaştırılmasıdır.

3. MATERYAL ve METOT

3.1. Uygulama Verisi

3.1.1. Gerçek Veri

Çalışmada kullanılacak olan ilk veri seti, Ondokuz Mayıs Üniversitesi Tıbbi Biyoloji Anabilim Dalı'nda 2007-2018 yılları arasında infertil ve fertil erkeklerden elde edildi. Fizik bakı bulguları, hormon analizi, rutin semen parametreleri ve genetik varyasyonlara ait bilgiler Tablo 1'de yer almaktadır. Veriler elde edilirken kişilerden aydınlatılmış onam alınmıştır. Bu çalışma Ondokuz Mayıs Üniversitesi Etik Kurulu tarafından onaylanmıştır (OMÜ KAEK 2017/208).

Tablo 1. Infertilite verisine ait açıklamalar

	Değişken	**Veri Tipi**	**Eksik Gözlem**
1	Sonuç (fertil, infertil)	Kategorik	-
2	Yaş	Sayısal	-
3	FSH düzeyi	Sayısal	3
4	Total testesteron düzeyi	Sayısal	3
5	gr/gr (polimorfizmi)	Kategorik	138
6	gr/gr+b2/b3 (polimorfizmi)	Kategorik	137
7	sy1191(polimorfizmi)	Kategorik	137
8	b2/b3 (polimorfizmi)	Kategorik	138
9	sy1291(polimorfizmi)	Kategorik	137
10	Sperm konsantrasyonu	Sayısal	87
11	LH düzeyi	Sayısal	3

Veri incelendiğinde toplam 587 gözlem, 530 infertil ve 57 fertil erkek, hedef değişken ile birlikte 11 değişken yer almaktadır. Eksik gözlemler için gerekli analizler yapılmıştır ve bu değerler analiz aşamasında çıkarılmıştır.

Bu çalışmada kullanılan ikinci veri seti, Ondokuz Mayıs Üniversitesi Diş Hekimliği Fakültesi Ağız Tanı ve Radyoloji ve Periodontoloji Kliniğine 2003-2006 yılları arasında dişeti şikâyetiyle başvuran 174 kişiden elde edilmiş olup, bunların 72'si kronik peridontitis hastası ve 102'si sağlıklı kontrollerden oluşmaktadır. Veri seti cinsiyet, D vitamini reseptörü geni varyantları, haplotip dizileri, mevcut diş sayısı ve çürük indeksi değişkenlerinden oluşmaktadır (Tablo 2). Bu çalışma da Ondokuz Mayıs Üniversitesi Etik Kurulu tarafından onaylanmıştır (OMÜ KAEK 2017/208). Onay belgesi Ek 1'de bulunmaktadır.

Tablo 2. Peridontitis verilerine ait açıklamalar

	Değişken	Veri Tipi	Eksik Gözlem
1	Sonuç (peridontitis hastası, sağlıklı)	Kategorik	-
2	Cinsiyet	Kategorik	-
3	Yaş	Sayısal	-
4	Bsm1 (Polimorfizmi)	Kategorik	-
5	Bsm2 (Polimorfizmi)	Kategorik	-
6	Apa1 (Polimorfizmi)	Kategorik	-
7	Apa2 (Polimorfizm)	Kategorik	-
8	Taq1 (Polimorfizmi)	Kategorik	-
9	Taq2 (Polimorfizmi)	Kategorik	-
10	Mevcut diş sayısı	Sayısal	16
11	Çürük diş indeksi (DMF)	Sayısal	-

3.1.2. Benzetim Verisi

Peridontitis veri seti özellikleri dikkate alınarak benzetim çalışmaları yapıldı. Bu çalışmada kullanılmak üzere örnek büyüklüğü 500 olan ilk benzetim verisi ve örnek büyüklüğü 1000 olan ikinci benzetim veri seti üretildi. Bu veride sonuç üzerine etkili olduğu düşünülen 7 tane ikili değişken ve 3 tane sayısal değişken bulunmaktadır. Veri setinde hasta sayısı örnek büyüklüğü 500 iken, 199 (% 39,80) peridontitis hastası ve sağlıklı birey 301 (% 60,20)'dir. Örnek büyüklüğü 1000 olduğunda ise 389 (%38,90) peridontitis hastası ve sağlıklı birey 611(% 61,10)'dir.

Öncelikle sayısal değişkenler için ortalama ve varyans değerleri elde edilmiştir. İkili değişkenler için yüzdelik değerleri ve gerçek veriden korelasyon matrisi elde edildi.

Performans ölçütleri için karmaşıklık matrisinden elde edilen duyarlılık, seçicilik, pozitif ve negatif kestirim değeri, doğruluk oranı değerleri kullanıldı, aynı zamanda alıcı işlem karakteristiği eğrisi [Receiver Operating Characteristic Curve (ROC)] dengesiz veri seti için çizildi ve EAA değerleri elde edildi. SL algoritmasının diğer algoritmalar ile karşılaştırılması için EAA değeri kullanıldı.

Ön işlem aşamasında sayısal verilere z- skoru normalleştirme işlemi uygulandı. Tüm analizler için 10 tekrarlı 10 katlı çapraz geçerleme yapıldı. Böylece aşırı uyum

problemi önlenmeye çalışılmıştır. SL analizi için 10 katlı ÇG yapıldı. Algoritmaların karşılaştırılması için gerçek veri yanında benzetim verileri de kullanıldı.

Analizler için R 3.5.3 programından yararlanıldı. Makine öğrenme adımları için 'plyr' ve 'ggplot2', sınıflama performansları için de 'caret', 'Superlearner', 'e071' ve 'rpart' paketleri kullanıldı. Dengesiz veriler için kullanılan paketler 'ROSE', 'SMOTE', 'Dmwr' ve 'pROC'dır. Gerçek veri özellikleri göz önünde bulundurularak 'Binnor' paketi ile farklı büyüklükte veri seti elde edildi. Benzetim ile veri üretmede gereksinim duyulan değişkenlere ilişkin tanımlayıcı istatistikler ile korelasyonlar ve eksik gözlemler için analizler Sosyal Bilimler için İstatistik Programı (SPSS) 21.0 yazılım kullanılarak gerçek veri setinden elde edildi (IBM, 2012).

3.2. Adımlar

3.2.1. Verinin Toplanması

Makine öğrenimi problemleri için tercih edilen büyük veri setleridir. Ayrıca denetimli öğrenmede veri setinde etiketli verilerin olması gerekmektedir. Bazen veriler etiketli şekilde toplanamaz. Verilerin toplanması, hazırlanması ve hedef değişkenin belirlenmesi bu sebeple en önemli adımdır. Bir tahmin elde edebilmek için örnek verinin elde edilmek istenen veriyi temsil etmesi beklenmektedir. Veri elde edildikten sonra algoritmaya ya da kullanılacak olan programa göre veri formatı belirlenir (Amazon, 2017). Örneğin R programı için veri setinin virgülle ayrılmış [Comma Separated Values (CSV)] veya metin dosyası [Tab Seperated Value (TXT)] formatı kullanılır (Peng, 2015).

3.2.2. Veri Hazırlama

Ham verinin analiz öncesi işlenmesi gerekmektedir. Bu işleme *ön-işleme* denir ve burada yapılan işlemler veri temizleme, veri entegrasyonu, veri dönüşümü ve veri indirgeme olarak sıralanmaktadır (Data, 2017; Han ve ark., 2012).

Veri Temizleme

Bu aşamada veri eksik gözlemler ve aykırı değerler yönünden değerlendirilir (Han ve ark., 2012).

Eksik Gözlemler: NB ve C4.5 algoritmaları eksik gözlem olduğunda işlem yapabilirken, SVM, EYK vb. algoritmalar çalışmaz (Michie ve ark.,1994). Veride eksik

gözlemler bulunduğunda kullanılabilecek farklı yöntemler bulunmaktadır (Han ve ark., 2012).

Gözlemin silinmesi, sınıflama yapılacaksa ve sınıf etiketlerini içeren gözlemler eksik olduğunda kullanılırken elle yazılması zaman alıcıdır ve büyük veri setlerinde fazla eksik gözlem olduğunda tercih edilmez. Sabit değer atanması, eksik gözlemler yerine "bilinmeyen" ya da "∞" gibi bir sabit atanarak yapılır. Eğer "bilinmeyen" etiketi tüm eksik gözlem yerine konulursa bu durumda algoritma bunun özel bir örüntü oluşturduğunu düşünür. Çünkü değerler hep aynı şekildedir. Bu durum basittir ancak güvenli değildir. Eksik değer için merkezi eğilim ölçüsü kullanılması da tercih edilebilir. Normal dağılım gösteren veriler için ortalama, çarpık dağılımlar için ortanca kullanılır. Son yöntem ise, eksik değer yerine en olası değer atanır. Bu değer regresyon, bayes formülasyonu kullanılarak elde edilen işlemler ile ya da karar ağaçları ile belirlenir (Han ve ark., 2012; Oğuzlar, 2003).

Aykırı Değerler: İstatistiksel olarak aykırı değer, bir dizi verinin geri kalanıyla tutarsız görünen bir gözlem anlamında kullanılır (Ben-Gal I, 2005). Bir veri setinde aykırı değerlerin olmasının sebepleri arasında veri girişi sırasında hatalar (insan kaynaklı hatalar), ölçüm hataları (ölçek kaynaklı hatalar), deneysel hatalar (veri çıkarma veya deneme planlama/uygulama hataları), veri işleme hataları ve örnekleme hataları bulunmaktadır (Santoyo, 2017).

Bir veriyi dört eşit parçaya bölen üç değere *çeyrekler* denir. 25. yüzdeliğe Ç1 (I. Çeyrek), 50. yüzdeliğe Ç2 (II. Çeyrek) ya da ortanca ve 75. yüzdeliğe Ç3 (III. Çeyrek) denir. Aykırı değerleri belirlemek için çeyrekler arası dağılım aralığı (ÇADA) denilen Ç3-Ç1 farkından yararlanılır (Alpar, 2016). $Ç1 - 1.5(\text{ÇADA})$ alt uç değer ile $Ç3 + 1.5(\text{ÇADA})$ üst uç değeri aşan değerler varsa buna aykırı değer denilebilir. Veri setindeki sayısal değerlere ait kutu grafikleri çizilerek bu değerler belirlenir. Bu yaklaşım dışında aykırı değerleri belirlemek için çeşitli istatistiksel yaklaşımlar da bulunmaktadır (Ovla ve Taşdelen, 2012).

Veri Birleştirmesi

Veri madenciliğinde birden fazla veri deposundan gelen verilerin bir araya getirilmesi gerekir. Bu işlemin dikkatlice yapılması fazlalıkların ve tutarsızlıkların azaltılmasına ve önlenmesine yardımcı olabilir. Bu doğruluğu artırır ve sonraki işlemlerde kolaylık sağlar (Han ve ark., 2012).

Veri Dönüşümü

Bu aşamada sayısal veriler normalleştirilir (standartlaştırma). Uzaklık ölçümlerinin kullanıldığı metotlar için bu işlem büyük aralıklara sahip değişkenler ile küçük aralıklara sahip olan değişkenlere uygun ağırlık verilmesini sağlamaktadır. Makine öğrenimi yöntemlerinde değişkenlerin normalleştirilmesi aşağıdaki şekilde yapılır (Data, 2017; Han ve ark., 2012).

Minimum-maksimum normalleştirilmesi Eşitlik 3.1' de verilmiştir.

$$v' = \frac{v - min_v}{max_v - min_v}(yeni\_max_v - yeni\_min_v) + yeni\_min_v \quad (3.1)$$

Z skoru ile normalleştirme Eşitlik 3.2' de verişmiştir.

$$v' = \frac{v - \bar{v}}{\sigma_v} \quad (3.2)$$

$\bar{v} =$ verinin aritmetik ortalaması ve $\sigma_v =$ verinin standart sapmasıdır.

Ondalık işlemler ile normalleştirme Eşitlik 3.3' de verilmiştir.

$$v' = \frac{v}{10^j} \quad (3.3)$$

$\max(|v'|) < 1$ koşulu için en küçük tam sayı j'dir.

Veri İndirgeme

Veri indirgeme, boyut açısından çok daha küçük olan, ancak aynı (veya hemen hemen aynı) analitik sonuçları veren veri kümesinin farklı bir gösterimidir. Veri indirgeme, boyut indirgeme ve sayıların azaltılmasını içerir. Boyut indirgeme, orijinal verinin azaltılması veya "sıkıştırılmış" gösterimini elde etmek için veri kodlama şemaları içerir. Buna örnek olarak temel bileşen analizini verebiliriz. Sayıların azaltılması, parametrik modeller (örn. regresyon) veya parametrik olmayan modeller (örn. histogramlar, kümeler, örnekleme veya veri toplama) ile yapılmaktadır (Han ve ark., 2012).

3.2.3. Modelin Eğitilmesi

Bir algoritmayı eğitmek için bir eğitim seti oluşturulur. Eğitim setinde hedef değişken bilinmektedir. Algoritma değişkenler ve hedef değişkenler arasında ilişki kurarak öğrenir. Makine öğrenme algoritmalarını test etmek için bir de test setinin olması gerekmektedir. Öncelikle programa eğitim seti verilir ve burada makine öğrenmesi gerçekleşir. Sonra test seti programa verilir. Hedef değişkenleri içeren örnekler programa verilmez, program hangi örneğin hangi sınıfa ait olduğuna karar verir. Test setindeki hedef değişken sınıfı, test setinde tahmin edilen değer ile kıyaslanır ve algoritmanın ne kadar doğru olduğu hakkında bilgi sahibi olunur (Harrington, 2012).

Eğitim için kullanılacak olan verinin değerlendirme aşamasında kullanılması yararlı değildir, çünkü burada modeli hatırlayacak veriler bulunacaktır. Burada kullanılan genel strateji mümkün tüm etiketli verileri eğitim ve test verisi olacak şekilde alt kümelere ayırmaktır. Genelde eğitim seti için %80, %70 ve %60 ve test seti için de %20, %30 ve %40 oranları kullanılmaktadır. Makine öğreniminde eğitim seti örüntü bulmak için kullanılırken test seti de eğitilmiş modelin tahmin performansını hesaplamak için kullanılmaktadır. Farklı ölçüler kullanılarak test setindeki tahminler gerçek değerlerle karşılaştırılarak tahmin performansı elde edilir. Genelde en iyi model kullanılarak test seti için bilinmeyen hedef çıktı etiketleri tahmin edilir (Amazon, 2017; Raschka, 2018).

3.2.4. Modelin Değerlendirilmesi

Farklı makine öğrenimi algoritmaları için farklı performans ölçüleri vardır. Örneğin sınıflama, regresyon, sıralama ve kümeleme için farklı ölçüler bulunmaktadır. Bu çalışmada sınıflama performanslarının ölçülmesi için kullanılan ölçüler açıklanacaktır. Doğruluk oranı, duyarlılık (geri çağırma), seçicilik, kesinlik ve EAA bunlardan en popüler olanlarıdır (Zheng, 2015).

Karmaşıklık Matrisi

Bir modelin performansını elde etmek için kullanılır. Özellikle sınıflandırma problemlerinde çıktı değişkeni iki sınıfa sahip olduğunda kullanılır. Performans ölçümlerinin çoğu Tablo 3'e göre ve içerisindeki sayılara dayanmaktadır (Sunasra, 2017).

Tablo 3. Karmaşıklık matrisi tablosu (Sunasra, 2017'den uyarlanmıştır)

		Gerçek	
Kestirim		**Pozitif**	**Negatif**
	Pozitif	Doğru Pozitif	Yanlış Pozitif
	Negatif	Yanlış Negatif	Doğru Negatif

Doğru Pozitif (DP): Gerçekte pozitif olan ve aynı zamanda sınıflandırıcı tarafından da pozitif olarak sınıflandırılmış gözlemlerin sayısını gösterir.

Doğru Negatif (DN): Gerçekte negatif olan ve aynı zamanda sınıflandırıcı tarafından da negatif sınıflandırılmış gözlemlerin sayısını gösterir.

Yanlış Pozitif (YP): Gerçekte negatif olan ancak sınıflandırıcı tarafından pozitif sınıflandırılmış gözlemlerin sayısını gösterir.

Yanlış Negatif (YN): Gerçekte pozitif olan ancak sınıflandırıcı tarafından negatif sınıflandırılmış gözlemlerin sayısını gösterir (Kaynar ve ark., 2016).

Doğruluk ve Hata Oranı

Model performansının değerlendirilmesinde kullanılan en basit ve en popüler ölçü, modele ait doğruluk oranıdır. Doğru sınıflandırılmış örnek sayısının (DP+DN), toplam örnek sayısına (DP+DN+YP+YN) oranıdır ve Eşitlik 3.4'de verilmiştir. Hata oranı ise bu değerin 1'e tamamlayanıdır. Yani yanlış sınıflandırılmış örnek sayısının (YP+YN), toplam örnek sayısına (DP+DN+YP+YN) oranıdır ve Eşitlik 3.5'de verilmiştir (Nizam ve Akın, 2014).

$$Doğruluk\ Oranı = \frac{DP+DN}{DP+DN+YP+YN} \tag{3.4}$$

$$Hata\ Oranı = \frac{YP+YN}{DP+DN+YP+YN} \tag{3.5}$$

Verideki hedef değişken sınıfları yaklaşık olarak dengede ise bu ölçünün kullanılması gerekmektedir. Ancak hedef değişken sınıfların büyük çoğunluğu aynı sınıftan oluşuyorsa, bu ölçünün kullanılmaması gerekmektedir (Sunasra, 2017).

Kesinlik

Kesinlik (pozitif kestirim değeri) doğru pozitif sayısının model tarafından pozitif olarak sınıflandırılmış toplam örnek sayısına oranıdır ve Eşitlik 3.6'da verilmiştir (Kaynar ve ark., 2016).

$$Kesinlik = \frac{DP}{DP+YP} \quad (3.6)$$

Duyarlılık

Pozitif olarak etiketlenmiş örneklerin gerçekten pozitif olan örneklerin toplam sayısına oranıdır ve Eşitlik 3.7'de verilmiştir (Kaynar ve ark., 2016). Geri çağırma doğru sınıflandırılmış örneklerin (doğru pozitiflerin) ve yanlış sınıflandırılmış örneklerin (yanlış negatiflerin) bir fonksiyonudur (Sokolova, 2006).

$$Duyarlılık = \frac{DP}{DP+YN} \quad (3.7)$$

Kesinlik, bir kişinin sonucu pozitif (hasta) olarak kestirildiğinden bunun doğru olması olasılığını gösterir. Geri çağırma ise modelin hastaların ne kadarını hasta olarak yakaladığı hakkında bilgi verir (Sunasra, 2017).

Seçicilik

Doğru sınıflandırılan negatif örneklerin toplam negatif örnek sayısına bölümüdür ve Eşitlik 3.8'de verilmiştir (Balaban ve Kartal, 2015). Seçicilik doğru negatifleri doğru negatif olarak tanımlama yeteneğini göstermektedir. Dolayısıyla, yüksek seçicilik, kullanılan herhangi bir algoritmanın, gerçek negatifleri tanımlama yeteneğinin yüksek olduğunu gösterir (Raza ve Hasan, 2013).

$$Seçicilik = \frac{DN}{DN+YP} \quad (3.8)$$

Alıcı İşlem Karakteristiği Eğrisi

İstatistikte alıcı işlem karakteristik eğrisi ikili sınıflama sistemlerinin performanslarını farklı eşik değerlerine göre grafiksel olarak göstermektedir. Eğri doğru pozitif oranına karşılık yanlış pozitif oranı için farklı eşik değerlerine göre çizilmektedir. ROC eğrisi, yanlış pozitiflik oranına karşın doğru pozitiflik oranını göstermektedir. Başka bir deyişle, daha fazla sayıda yanlış pozitiflik yapmaya izin verdiğinizde kaç doğru pozitif sınıflandırmanın elde edilebileceğini göstermektedir (Turiplatform, 2018).

Tek bir sınıflayıcının performansını hesaplamanın yanı sıra, ROC ile farklı sınıflayıcıları karşılaştırabiliriz. Aynı sınıflayıcıyı farklı öğrenme metotlarına göre karşılaştırabileceğimiz gibi farklı sınıflayıcıları da birbiri ile karşılaştırabiliriz.

Mükemmel sınıflayıcı (0,1) noktasında olmaktadır (%100 doğru pozitif %0 yanlış pozitif), tüm sınıflamaların yanlış olduğu nokta ise (1,0)'dır. Sol üst köşeye eğrinin yakın olması sınıflayıcının performansının iyi olduğun göstermektedir. Köşegen üzerindeki (0,0) – (1,1) doğru ise şans doğrusunu göstermektedir. Yani bu durumda %50 doğru pozitif ve %50 yanlış pozitif şeklindedir (Marsland, 2015).

ROC eğrisinin altında kalan alanın hesaplanması ile EAA elde edilir. EAA veri madenciliği literatüründe çok sık kullanılan sıralamaya dayalı (ranking) bir performans kriteridir. 0 ile 1 arasında değerler alabilir; 0.5 değeri rastgele bir tahmin olduğunu, 1'e yakın değerler modelin tahmin gücünün yüksek olduğunu gösterir (Coşgun ve Karaağaoğlu, 2011). EAA Eşitlik 3.9 ile elde edilmektedir (Goncalves ve ark., 2014).

$$EAA = \int_0^1 ROC(u)du \tag{3.9}$$

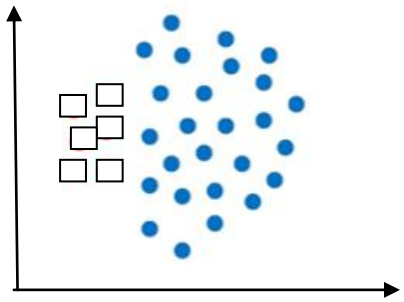
ROC kullanmanın bir avantajı duyarlılık ve seçicilik gibi ölçütleri kullanmasının yanı sıra sınıflar arasındaki dengesizliğe karşı duyarsız olmasıdır. Dezavantajı ise bilginin belirsiz olmasıdır. ROC eğrisi, eğrinin belirli kısımlarına odaklanan bir alternatiftir (Kuhn ve Johnson, 2013).

3.2.5. Performans İyileştirilmesi

Farklı algoritmalar için bu algoritmaları oluşturan parametreler bulunmaktadır. Bu sebeple bir algoritmanın performansı büyük oranda bu parametrelerin değerlerine bağlıdır. Bu parametreleri ayarlayarak algoritmaların performansları optimize edilebilir ve bir problem için minimum yineleme sayısı elde edilebilir (Yang ve ark., 2013). Örneğin RO algoritması için optimize edilmesi gereken parametre rastgele seçilen kaç değişken ile modelin kurulacağını belirten *mtry* dir. DVM için iki parametre bulunmaktadır; bunlar *sigma* ve *C* (Cost) (Korkmaz, 2017). Sınıflama performansını arttırmak için topluluk yöntemleri de kullanılabilir (Han ve ark., 2012).

3.3. Dengesiz Veri Probleminde Kullanılan Yöntemler

Veri setlerinde sonuç değişkenin kategorilerindeki gözlem sayıları her zaman dengeli olmayabilir. Dengesiz veri setine örnek Şekil 3'te verilmiş olup, kare ile azınlık sınıfı ve daire ile çoğunluk sınıfı temsil edilmektedir. (Bulut, 2016).



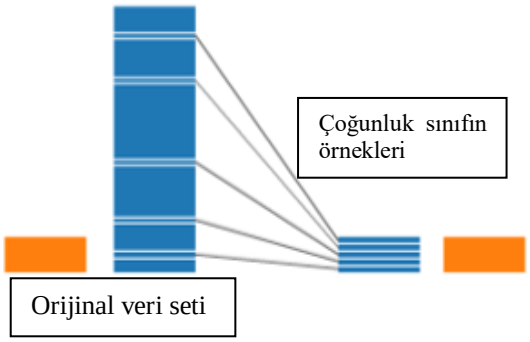
Şekil 3. Örnek dengesiz veri seti (Tripathi, 2019'dan uyarlanmıştır)

Analiz aşamasında dengesiz veri seti örnek sayısı fazla olan sınıf etiketi ile eğitileceği için modelde bir önyargı oluşmaktadır. Ayrıca azınlıkta olan sınıf etiketi ile model kendini yeterince eğitemediği için başarılı sınıflandırma yapamamaktadır. Bu durum performans üzerinde etkilidir (Bulut, 2016). Bu problemi çözmek için literatürde karşılaştırmalar yapılırken farklı performans ölçülerinin, algoritmaların ve yeniden örnekleme tekniklerinin kullanılması önerilmektedir (Boyle, 2019). Yeniden örnekleme için kullanılan yöntemler AÖ, YÖ, ROSE ve SMOTE şeklinde sayılabilir (Tantithamthavorn ve ark., 2018).

3.3.1. Aşağı Örnekleme

Bu teknikte çoğunluk sınıfından rastgele örnekler seçilir ve geriye kalan örnekler atılır. Burada yapılan rastgele seçimin verinin dağılımını gösterdiği varsayılmaktadır. Sınıf dağılımını çoğunluk sınıfından rastgele seçilen örnekler ile dengeleyen klasik bir yöntemdir. Ancak sınıflayıcı için yararlı olabilecek bazı örnekler yok edilmektedir (Tantithamthavorn ve ark., 2018). Bu yönteme ait adımlar Şekil 4 ile gösterilmiştir.

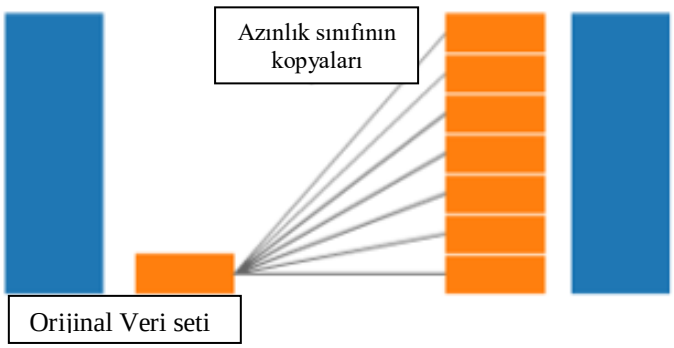
Bu yöntemde öncelikle veri setine ait azınlık ve çoğunluk sınıfları belirlenir. Aşağı örnekleme oranına göre silinecek örnek sayısı hesaplanır. Daha sonra çoğunluk sınıfından bir örnek rastgele seçilir ve çoğunluk sınıfından silinir. Son adım, bir önceki adım da aşağı örnekleme oranı kadar örnek silinene kadar tekrarlanır (Kaur ve Gosian, 2018).



Şekil 4.Aşağı örnekleme tekniğinin işleyişi (Badr, 2019'dan uyarlanmıştır)

3.3.2. Yukarı Örnekleme

Bu teknikte azınlık sınıfından rastgele ve yerine koyarak örnekler seçilir. Elde edilen örnekler çoğunluk sınıf sayısı ile eşit hale getirilir (Tantithamthavorn ve ark., 2018). Bu yönteme ait adımlar Şekil 5'te verilmiştir.



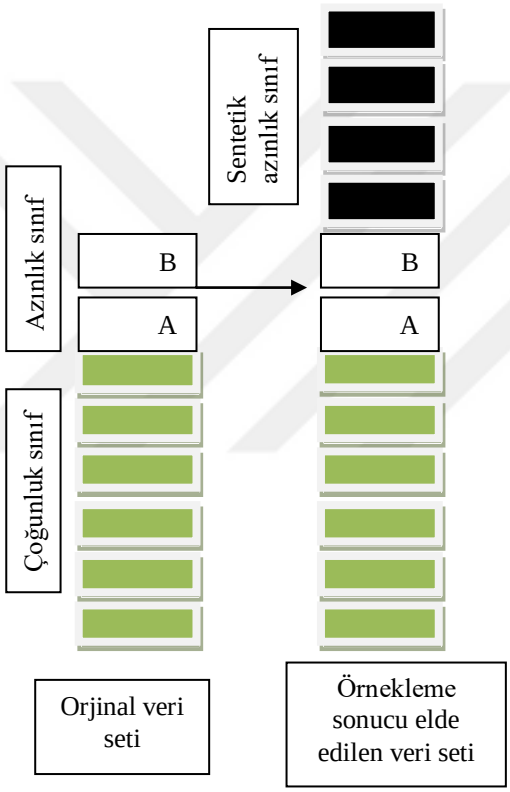
Şekil 5.Yukarı örneklem tekniğinin işleyişi (Badr, 2019'dan uyarlanmıştır)

Bu tekniğin olumlu yönü herhangi bir bilgi kaybına sebep olmamasıdır. Olumsuz yönü ise veri setinden örnekler çoğaltıldığı için eğitim setinde benzer verilerin olması sebebi ile aşırı öğrenmeye sebep olabilmektedir. Bir başka problem ise örnek sayısı artarken modelin karmaşıklığı da artmaktadır. Bu durumda modelin çalışma süresi de artmış olacaktır (Tantithamthavorn ve ark., 2018).

İki tür yukarı örnekleme tekniği bulunmaktadır. Biri rastgele yukarı örnekleme tekniği, yukarıda bu teknikten bahsedilmiştir, diğeri ise sentetik yukarı örneklemedir. Bu teknikte azınlık sınıfından sentetik veriler elde edilmektedir (Babar ve Ade, 2015).

3.3.3. SMOTE Örnekleme

Yukarı örnekleme ile elde edilen tekrarlı örnekler azınlık sınıfındaki örneklerin eşleniği olduğu için analiz aşamasında aşırı örnekleme problemi oluşmaktadır. SMOTE tekniğinde azınlık sınıfından sentetik veriler üretilir. Azınlık sınıfındaki örneklerin k en yakın komşulukları rastgele belirlenir ve buna göre sentetik yeni örnekler oluşturulur (Maimon ve Rokach, 2005). Üretilecek olan sentetik örnek sayısı yukarı örnekleme oranına göre belirlenir (Kaur ve Gosain, 2018). Bu yönteme ait adımlar Şekil 6'da verilmiştir.

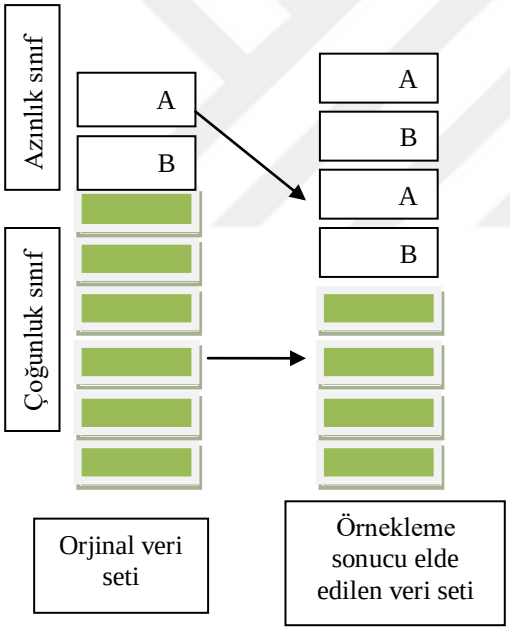


Şekil 6. SMOTE örnekleme tekniğinin işleyişi (Tantithamthavorn ve ark., 2018'den uyarlanmıştır)

SMOTE örneklemede öncelikle veri setine ait azınlık ve çoğunluk sınıfları belirlenir. Aşağı örnekleme oranına göre üretilecek örnek sayısı ve yukarı örnekleme örnek sayısı hesaplanır. Azınlık sınıfından rastgele bir örnek belirlenir ve k en yakın komşulukları bulunur. En yakın komşu seçilir ve rastgele örnek ile seçilen en yakın komşu arasındaki fark hesaplanır. Bu fark 0 ile 1 arasında rastgele bir sayı ile çarpılır. Bu fark seçilen rastgele örneğe eklenir. Son olarak verilen oran değeri kadar adımlar tekrarlanır (Kaur ve Gosain, 2018).

3.3.3. ROSE Örnekleme

ROSE tekniği azınlık sınıftaki örneklerden düzeltilmiş bootstrap yaklaşımı ile örnekler oluşturur. Burada aşağı ve yukarı örnekleme birleştirilerek örnek veri seti artırılmaktadır. Burada artan azınlık sınıfındaki örneklerdir. Aşağıdaki adımlar ile yapay veri üretilmektedir. Bu örneklemede öncelikle çoğunluk sınıftaki %50 oranındaki gereksiz örneği çıkarmak için çoğunluk sınıf üzerinde bootstrap örnekleme yapılır (aşağı örnekleme). Sonra azınlık sınıftaki %50 oranındaki örnek tekrarlanır ve azınlık sınıf üzerinde bootstrap örnekleme yapılır (yukarı örnekleme). Komşuluğu bulunan yeni örnekler R paketi yardımıyla ve rose.real fonksiyonu kullanılarak üretilir. Bu adımlar her eğitim seti için orjinal veri setinin büyüklüğüne ulaşana kadar tekrarlanır. Bu durumda her sınıf için sentetik olarak eşit şekilde temsil edecek veriler elde edilmiş olur (Tantithamthavorn ve ark, 2018; Lunardon, 2014). Şekil 7 ile bu tekniğe ait adımlar gösterilmiştir.



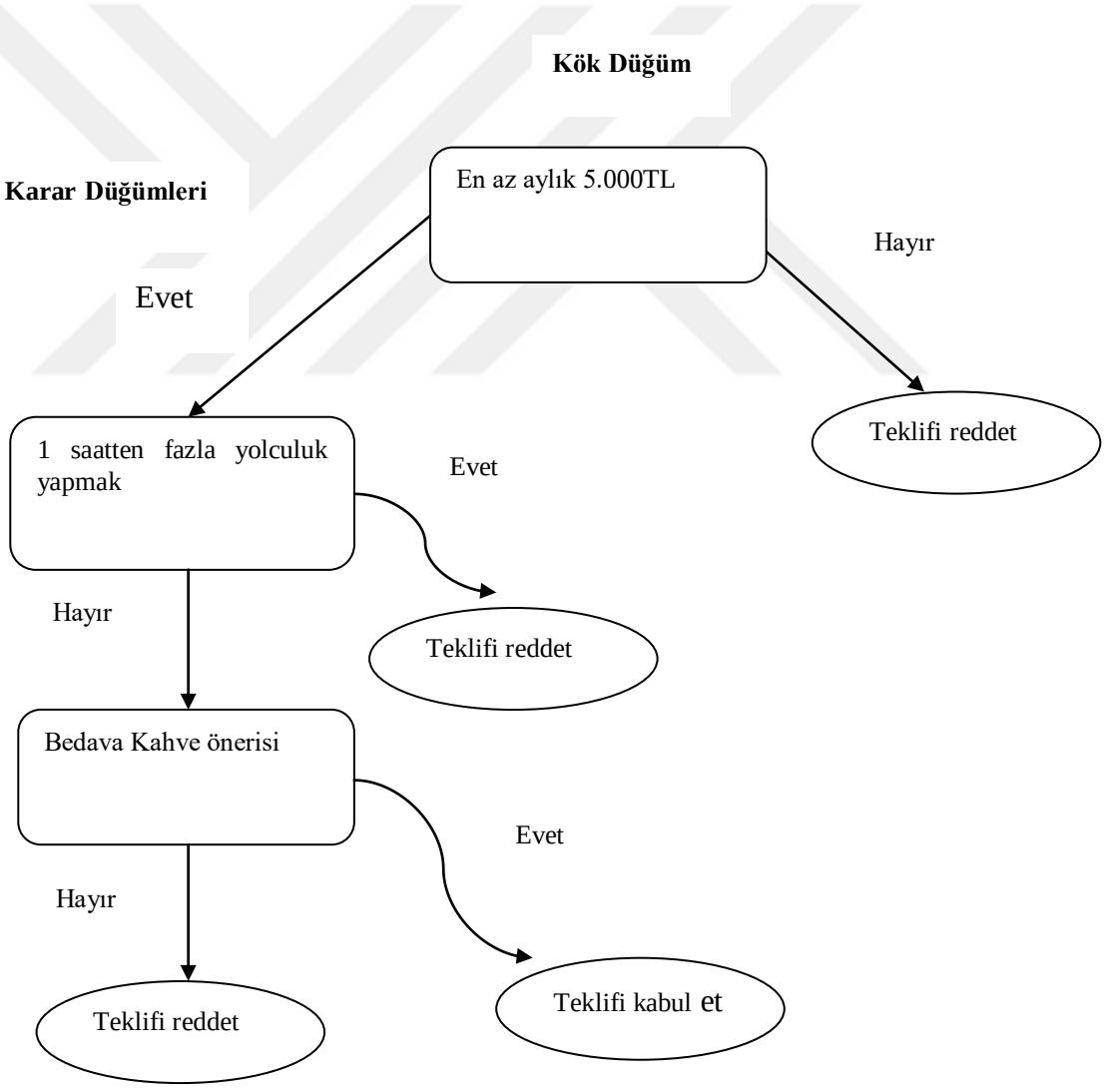
Şekil 7.Verilen ROSE tekniğin işleyişi (Tantithamthavorn ve ark., 2018'den uyarlanmıştır)

3.4. Makine Öğreniminde Kullanılan Algoritmalar

Bu çalışmada makine öğreniminde sınıflama için kullanılan altı farklı algoritma dikkate alınmıştır. Bunlar karar ağaçları, EYK, NB, DVM, RO ve SL algoritmalarıdır.

3.4.1. Karar Ağacı

Karar Ağacı özellikler ve potansiyel çıktılar arasındaki ilişkileri modellemek için ağaç şeklinde bir yapı kullanan güçlü sınıflandırıcılardır. Şekil 8 bir iş önerisinin kabul edilip edilmeyeceği ile ilgili bir karar ağacı modelini göstermektedir (Lantz,2015).



Şekil 8. Karar ağacı modeli (Lantz, 2015'den uyarlanmıştır)

Buna göre iş önerisi kök düğümden başlar, karar düğümler ise işin niteliklerine dayalı olarak yapılması gereken seçenekleri gösterir. Bu seçenekler veriyi bir kararın olası sonuçlarını gösteren dallara böler, burada çıktılar evet ve hayır olarak alınmıştır. Son kararın verilmesi için ağaç, yaprak düğümlerle sonlandırılır. Bu kararlar dizisi sonucunda yapılacak olayı göstermektedir. Yani tahmin modeli için yaprak düğümler beklenen sonucu göstermektedir (Lantz, 2015).

Çeşitli karar ağaçları algoritmaları bulunmaktadır. Bunlardan en çok kullanılan algoritmalar ID3, C4.5 algoritması ile CART algoritması. (Singh ve Giri, 2014).

Yinelemeli Bölücü (ID3) Algoritması

ID 3 algoritması en popüler karar ağacı tasarımlarından biridir. Kayıp değerlere ve gürültülü verilere karşı toleranslı değildir. Bölünme ve örneklemin homojenliği için entropi (belirsizlik) ölçüsü kullanılmaktadır (Makhabel, 2015).

Shannon (1948) öncülüğünü ettiği bilgi teorisinde Bilgi Entropi ölçeğini, özellikler kümesinin saf olmama miktarı Eşitlik 3.10'da tanımlamıştır (Marsland, 2015);

$$Entropi(X) = -\sum_{i=1}^{k} p_i log_2(p_i) \quad (3.10)$$

Bir veri seti $C_1, C_2, \dots, C_k$ şeklinde birkaç sınıftan oluşsun ve X sınıf değerini göstersin. Bir sınıfa ait olasılık değeri $p_i = \frac{C_i}{|X|}$ dir (Kavzoğlu ve Çölkesen, 2010).

Eşitlik 3.10 kullanılarak her özellik için entropi değerleri hesaplanır. Böylece hangi özelliği seçeceğimize dair uygun bir ölçüt elde edilmiş olur. Seçilen özelliğin entropisi Eşitlik 3.11'de verilmiştir. Bir sonraki sınıflama adımı için seçilecek olan belirli bir özellik tüm eğitim setindeki entropinin ne kadar azalacağını belirlemektedir. Bu bilgi kazancı olarak bilinmektedir ve tüm kümenin entropisi eksi seçilen özelliğin entropisine eşittir. Bu, Eşitlik 3.12'de verilmiştir (Marsland, 2015);

$$Entropi_A(X) = \sum_{j=1}^{v} \frac{|X_j|}{|X|} \times Entropi\,(X_j) \quad (3.11)$$

Bilgi kazancı, aşağıdaki gibi hesaplanır;

$$Bilgi\ Kazancı\ (X, A) = Entropi(X) - Entropi_A(X) \quad (3.12)$$

A özelliği $\{a_1, a_2, \dots, a_v\}$ gibi v farklı değerlere sahiptir. A özelliği X'i $\{X_1, X_2, \dots, X_v\}$ şeklinde v parçaya ayırır (X_j' de A özelliği için a_j değerine sahip veriler vardır) (Balaban ve Kartal, 2015). Son ağacın kökü en yüksek bilgi kazancına sahip özelliğe göre belirlenir. Buna bağlı olarak alt ağaçlar yinelemeli olarak köke bağlı oluşturulur (Makhabel, 2015).

ID3 Algoritmasında her özellik için bilgi kazancı hesaplanır. Her aşamada en yüksek bilgi kazancı seçilir ve ağaç oluşturulur. Bu ağacı oluşturan ise düğümler ve dallardır. Her aşamada en iyi özellik seçilir ve geri kalan veri üzerinde yinelemeli olarak işlem yapılır. Bu işleme veride tek bir sınıf kalana kadar devam edilir (Marsland, 2015).

Bu algoritmada öncelikle orijinal veri seti kök düğüm olarak belirlenir. Her adımda algoritma kullanılmamış olan veri setinde, A nitelikleri için Entropi hesaplar ya da bilgi kazancı hesaplanır. En küçük entropi değerini sahip olan nitelik seçilmektedir. Veri seti X seçilen niteliğe göre bölünür böylece veri alt kümelere bölünür. Algoritma, her alt kümede tekrarlanmaya devam eder. Daha önce seçilmemiş nitelikler kullanılır. Bu adımlar alt kümedeki her eleman aynı sınıfa ait olduğunda, o zaman düğüm yaprağa dönüşür ve o örnekler sınıfı ile temsil edilir. Seçilecek nitelik kalmadığında ancak örnekler aynı sınıfa ait değilse, bu durumda düğüm yaprağa dönüşür ve alt küme içerisinde en sık tekrar edilen sınıf ile temsil edilir. Alt küme içerisinde örnek kalmadıysa, bu durum üst kümedeki hiçbir örnek seçilen özellik için belirli bir değer ile eşleştirilemeyince sona erer. Böylece ağaç bölünen niteliklerin meydana getirmiş olduğu düğüm ve yapraklardan oluşur (Zprzydatek, 2014).

C4.5 Algoritması

Bu algoritma ID3 algoritmasının çözemediği bir takım problemleri çözmek amacıyla geliştirilmiştir. Bu sorunlardan en önemlileri eksik gözlem olduğunda ve sürekli veriler içinde bu algoritmanın çözüme ulaşmasıdır (Makhabel, 2015). ID3 algoritması büyük sayı değerleri olan özelliklere karşı aşırı hassas bir algoritmadır. Bu durumu düzeltmek için C4.5 algoritması "Kazanç Oranı" ölçütünü kullanmaktadır. Kazanç oranı Eşitlik 3.13'de verildiği gibi hesaplanır (Hssina, 2014).

$$Kazanç\ Oranı\ (A) = \frac{Bilgi\ Kazancı\ (A)}{Bölme\ Bilgisi(A)} \tag{3.13}$$

Bölme bilgisi ise;

$$Bölme\ Bilgisi_A(X) = -\sum_{j=1}^{v} \frac{|X_j|}{|X|} \times log_2\left(\frac{|X_j|}{|X|}\right) \quad (3.14)$$

Burada bölme bilgisi Eşitlik 3.14'de verilmiştir, eğitim verisinin A özelliğinin aldığı v farklı değere bağlı olarak v parçaya ayrıldığında oluşturulan esas bilgiyi temsil etmektedir. En yüksek kazanç oranını sağlayan özellik, ayrımın yapılacağı özellik olarak seçilmektedir (Balaban ve Kartal, 2015; Kavzoğlu ve Çölkesen, 2010).

C4.5 algoritması ile karar ağaçları oluşturulurken eğitim verileri kümesi kullanılmaktadır. Ağaçtaki her düğüm için örneklemi en iyi alt kümelere ayıran sınıf belirlenir. Yani en yüksek bilgi kazancına sahip olan nitelik belirlenir ve bölünme yapılır. Bu algoritma sürekli değişkenler olduğunda kullanılabilir. C_i niteliği sürekli değere sahip olsun. Eğitim aşamasında bu değerler incelenir. $A_1, A_2, \dots, A_m$ artan değerler olacak şekilde diziler. Sonra tüm bu değerler için, C değerine sahip olan değerler içerisinden A_j'den küçük ya da eşit olacak şekilde, A_j'den büyük olacak şekilde bölünürler ve her biri için kazanç oranı hesaplanır ve bölünmeyi maksimize eden kazanç seçilir (Hssina, 2014).

Karar ağaçlarında çoğu algoritmalar sürekli değerler için orta noktayı bölünme değeri (eşik) olarak almaktadır. Bu değer Eşitlik 3.15'de verildiği gibi hesaplanır. Aynı zamanda C4.5 algoritması eğitim seti içerisinde A niteliği için orta noktayı aşmayan en üst değeri eşik değer olarak Eşitlik 3.16 ile gösterilmiştir (Berzal ve ark., 2004).

$$t_i = \frac{v_i + v_{i+1}}{2} \quad (3.15)$$

$$t_i = max\left\{v | v \leq \frac{v_i + v_{i+1}}{2}\right\} \quad (3.16)$$

Karar ağaçları gürültülü veriler ile baş edebilirler çünkü etiketler hedef niteliğin en çok tekrar eden değerine atanırlar. Bir başka özelliği ise kayıp veriler ile baş edebilmesidir. Bir örneğin değerinin eksik olduğunu varsayalım. Bu durumda ağaçta o düğüm kullanılmaz ancak alabileceği tüm değerler göz önüne alınır (Marsland, 2015).

Ağaç Budanması

Birçok araştırma alanında olduğu gibi tıpta da kullanılan veriler belirsizdir yani kurallara uymayan gürültülü veriler bulunmaktadır. Karar ağaçlarında bu durum sınıflarla benzerlik göstermesi ama gerçekte farklı olması anlamına gelmektedir. Bu durumda karar ağacı algoritmaları birkaç örneğin bulunduğu düğümler oluştururlar ve geri kalan ağaç çok büyük olur. Özellikle alt bölüme yakın bazı dallar örnek değişkenliği nedeniyle mevcut olup istatistiksel açıdan anlamsızdır. Budama yöntemleri ile bu dallar kesilerek bu durum önlenmeye çalışılır (Fournier ve Cr´emilleux, 2002) .

Veri setlerinde özellik sayısı arttığında kurulacak olan ağaç gereksiz düğüm noktalarını da oluşturmaktadır. Aşırı öğrenme olarak adlandırılan bu durum başarı oranını olumsuz yönde etkilemektedir. Budama işlemi genellikle aşırı öğrenmeden kaynaklı olumsuz etkileri yok etme amacıyla uygulanmaktadır (Gümüşçü ve ark., 2016.)

Budanmış ağaçlar daha az karmaşık ve küçüktür ve daha kolay anlaşılabilir. Ayrıca verileri daha hızlı ve yüksek doğruluk ile sınıflayabilmektedir. Ağaç budanması için iki genel yaklaşım vardır. Bunlar önce budama ve sonra budamadır (Han ve ark., 2012).

Önce budama yaklaşımında ağaç oluşturulurken budama yapılır. Burada alt ağaçlar oluştururken ne zaman durulması gerektiğine karar verilir (Witten ve Frank, 2005). Bu yaklaşım gereksiz işlem yapmayı önlediği için caziptir. Ancak önemli yapılar bu sayede ihmal edilebilir. Sonra budama da ise ağaç çok büyük olacak şekilde oluşturulur ve sonrasında fazla dallar budanır. Bu yaklaşım ilk yaklaşıma göre daha etkilidir çünkü bir ağaç oluşturulmadan derinliğini belirlemek mümkün değildir. Sonradan budama ile algoritma ağaçta yer alan tüm yapıların ortaya çıkmasını sağlar (Lantz, 2015).

3.4.2. K En Yakın Komşuluğu

K en yakın komşuluğu benzetme yolu ile öğrenmedir. Yani belirli bir test grubunu ona benzeyen eğitim verileri ile karşılaştırılarak elde etmeyi amaçlamaktadır. Eğitim setleri n değişken ile tanımlanmaktadır. Her set n boyutlu uzayda bulunmaktadır. Bilinmeyen bir set verildiğinde, EYK, K eğitim seti olan modeli elde etmeye çalışmaktadır. Bu K eğitim setleri bilinmeyen set için “k” en yakın komşuları olmaktadır (Han ve ark., 2012).

EYK örnek tabanlı öğrenme yöntemleri arasında yer almaktadır. Bu yöntemde öğrenme, depolanan veri setine göre yapılmaktadır. Yeni veri seti geldiğinde depolanan bilgiye benzer örnekler belirlenir ve yeni gelen örnekler bu benzer örneklere göre sınıflandırılır (Mitchell, 1997).

EYK algoritması örüntü tanıma, metin kategorizasyonu, nesne tanıma ve olay tanıma uygulamalarında yaygın olarak kullanılmaktadır (Bhatia ve Vandana, 2010). EYK parametrik olmayan algoritmalardan biridir ve optimal "k" değeri için iyi performans gösterir (Kataria ve Singh, 2013). Burada parametrik olmayan verilerin boyutundan bağımsız olarak, hiçbir parametre veya sabit sayıda parametre bulunmadığı anlamına gelir. Bunun yerine parametreler, eğitim veri setinin boyutuna göre belirlenir. Herhangi bir varsayıma sahip değildir. EYK, verilerin dağılımı hakkında önceden bilgi sahibi olmadığımız herhangi bir sınıflandırma çalışması için en iyi seçim olabilir. Bu metot tembel öğrenme yöntemlerinden biridir. Bu yöntemde, tüm eğitim verileri depolanır ve bir öğrenme modeli oluşturmaya gerek kalmadan test verisi üretilene kadar bekler (Wettschereck ve ark., 1997). EYK, örüntü tanıma ve sınıflandırma için en eski, en basit ve en doğru sonuçları veren algoritmalardan biridir (Prasath ve ark., 2017).

EYK algoritması aşağıdaki adımlarla özetlenebilir: k değeri belirlenir ve uzaklık ölçüsü hesaplanır, sınıflandırmak istenilen örnek için en yakın k komşulukları bulunur, oy çoğunluğuna göre sınıflar atanır (Raschka, 2015).

k parametre değeri deneysel olarak belirlenir. Her probleme göre farklı komşuluk k değerleri denenir ve en yüksek doğruluk oranı veren değer seçilir (Hassanat ve ark., 2014). Küçük, orta veya büyük veri setlerinde uygun k parametresini belirlemek için teorik bir yol bulunmamaktadır. Optimum değer veriye ve seçilen eğitim verisine göre belirlenmektedir. Pratikte ÇG kullanarak yanlış sınıflama değerlerine göre farklı k değerleri elde edilir. Bu işlem sonunda iki ya da daha fazla k değeri elde edilir (Ghosh, 2006). k değeri genelde 3, 5 gibi tek sayı değerleri ile ifade edilir. Eşitlik olmaması için k değeri tek sayı olarak seçilir (Kılınç ve ark., 2016).

EYK sınıflayıcısının performansı öncelikle seçilen k değerine ve kullanılan uzaklık ölçüsüne bağlı olarak elde edilir (Song ve ark., 2007).

Uzaklık Ölçüsü

x ve y vektörleri arasındaki uzaklık fonksiyonu $d(x,y)$, iki vektör arasındaki uzaklığı negatif olmayan bir reel sayı olarak tanımlamaktadır. Bir fonksiyon her $x,y,z \in X$ için $d: X \times X \to \mathbb{R}$ ise X'in metrik olması için aşağıdaki özellikleri sağlamalıdır (Deza ve Deza, 2009).

1) Negatif Olmama: x ve y arasındaki uzaklık her zaman sıfırdan büyük ya da eşittir.

$$d(x,y) \geq 0$$

2) Belirsizlik: x ve y arasındaki uzaklık sadece " $x = y$ " olduğunda sıfıra eşittir.

$$d(x,y) = 0 \quad eğer\ x = y$$

3) Simetriklik: x ve y arasındaki uzaklık y ve x arasındaki uzaklığa eşittir.

$$d(x,y) = d(y,x)$$

4) Üçgen eşitsizliği: Bir z noktası göz önüne alındığında, x ve y arasındaki uzaklık x ve z ile y ve z arasındaki uzaklıkların toplamlarından küçük ya da eşittir.

$$d(x,y) \leq d(x,z) + d(z,y)$$

Literatürde 8 ana başlık olmak üzere 54 tane uzaklık fonksiyonu yer almaktadır. Prasath ve ark. (2017) EYK sınıflayıcılar için bu uzaklık fonksiyonlarını incelemiştir. Gerçek veri setleri üzerindeki etkilerin yanı sıra veri içerisinde gürültü olduğundaki sonuçlar elde edilmiştir. Bu çalışma sonucunda EYK sınıflayıcısının başarısının kullanılan uzaklık fonksiyonlarından etkilendiği ve farklı fonksiyonların arasında büyük farklılıkların elde edildiği görülmüştür (Prasath ve ark., 2017).

A ve B gibi iki nokta arasındaki uzaklığı özellik uzayında ölçmek için en çok kullanılan ölçü Öklid uzaklık fonksiyonudur. $A = (x_1, x_2, \dots, x_m)$ ve $B = (y_1, y_2, \dots, y_m)$ özellik vektörleri ile temsil edilmek üzere; m ise özellik uzayında boyutu temsil ettiğinde A ve B için normalleştirilmiş öklit uzaklık ölçüsü Eşitlik 3.17 ile verilmiştir (Hu ve ark., 2016);

$$d(A,B) = \sum_{i=1}^{m} \sqrt{\frac{(x_i - y_i)^2}{m}} \tag{3.17}$$

Öklit uzaklığı en çok kullanılan uzaklık ölçüsüdür ve Eşitlik 3.18 ile verilmiştir (Kuhn ve Johnson, 2013);

$$d_{öklit} = \sum_{i=1}^{m} \sqrt{(x_{ai} - y_{bi})^2} \tag{3.18}$$

x_a ve y_b iki ayrı örneği temsil etmektedir.

Diğer bir uzaklık ölçüsü ise Minkowski uzaklık ölçüsüdür ve Eşitlik 3.19 ile verilmiştir (Batchelor, 1978);

$$d_{Minkowski(X,Y)} = \left(\sum_{i=1}^{m} |x_i - y_i|^r\right)^{1/r} \tag{3.19}$$

$X = (X_1, X_2, \dots, X_m)$ $Y = (Y_1, Y_2, \dots, Y_m)$ iki vektörü göstermektedir.

r ; 1 değeri aldığında Manhattan (City Block), 2 değerini aldığında ise Öklid uzaklığı olur ve ∞ da Kare uzaklığı adını alır.

Korelasyon uzaklık fonksiyonu ve ki-kare uzaklık fonksiyonu ise sırasıyla Eşitlik 3.20 ile ve 3.21'de verilmiştir (Michalski ve ark., 1981; Marin-Reyes ve ark., 2016);

$$d_{Korelasyon(A,B)} = \frac{\sum_{i=1}^{m}(x_i-\mu_i)(y_i-\mu_i)}{\sqrt{\sum_{i=1}^{m}(x_i-\mu_i)^2 \sum_{i=1}^{m}(y_i-\mu_i)^2}} \tag{3.20}$$

$$d_{Kikare(A,B)} = \sum_{i=1}^{m} \frac{1}{sum_i}\left(\frac{x_i}{sizeQ} - \frac{y_i}{sizeJ}\right) \tag{3.21}$$

Burada sum_i; eğitim setindeki i değişkenlerin toplamı; $size_Q$, Q vektörü toplamı; $sizeJ$, J vektörü toplamını gösterir.

EYK algoritmasında yeni bir örnek (q) en yakın komşusunun çoğunluk oyuyla sınıflandırılır, yani q en yakın komşuları arasında en yaygın olan sınıfa atanır. Eğer k = 1 ise örnek en yakın komşusunun sınıfına atanır (Dhriti ve Kaur, 2012). q sınıfını belirlemek için en yakın komşuya büyük ağırlık atanması ile de sınıflar belirlenebilir. Bunun için genel bir teknik komşulara uzaklıklarının tersi ağırlık oylarının verilmesi ile Eşitlik 3.22'deki gibi sınıf belirlenebilir (Cunningham ve Delany, 2007);

$$oy(y_j) = \sum_{c=1}^{k} \frac{1}{d(q,x_c)^n} 1(y_j, y_c) \tag{3.22}$$

Burada, sınıf y_j ve komşu x_c uzaklık değeri ile 1'e bölündüğünde oy değerini göstermektedir yani $1(y_j, y_c)$ sınıf etiketleri aynı ise 1 olurken aynı olmaz ise 0 olur.

EYK Algoritması

Eğitim verisi D iken test verisi d olmak üzere öncelikle d ile D için uzaklıkları hesaplanır. k burada komşuluğu göstermektedir. d'ye en yakın k örnekleri belirlenir. Daha sonra d sınıfı en çok tekrarlanan sınıf olarak atanır. Eğitim aşamasında eğitim verileri ve bunlara ait sınıf etiketleri depolanır. Eksik gözlemler ve sayısal olmayan veri bu aşamada yer alamaz. Sınıflama aşamasında ise test verileri çoğunluk oyuna göre test verisi ile depolanan tüm eğitim verileri arasındaki uzaklık formülüne göre ya da benzerlik fonksiyonları kullanılarak hesaplanır. Sonra test verisi için en yakın k komşusu belirlenir, daha önceden k değeri için küçük bir tamsayı değeri seçilir. Bu k komşuların en çok tekrar eden sınıfı test verisine atanır (Prasath ve ark.,2017).

3.4.3. Naive Bayes

Naive Bayes Algoritması, Bayes teoremine dayanan basit bir olasılıksal sınıflandırma metodudur. Mevcut sınıflandırılmış örnek verileri kullanarak yeni bir verinin mevcut sınıflardan herhangi birine ait olma olasılığını hesaplayan bir algoritmadır (Karakoyun ve Hacıbeyoğlu, 2014). NB algoritmasının kullanılması için bir takım kabuller yapılır. Bunlardan en önemlisi niteliklerin birbirinden bağımsız olduğudur. Niteliklerin hepsinin aynı derecede önemli olduğu kabul edilir (Mademir, 2011). Bayes Sınıflaması denetimli öğrenme yöntemlerinin yanı sıra istatistiksel sınıflama metodunu temsil etmektedir (Software, 2017).

NB bilinmeyen gözlem veya eksik gözlem olduğu durumlarda kolayca işlem yapmaktadır. Verilen sınıflara göre nitelikler birbirinden bağımsız olduğunda da iyi sonuçlar elde edilmektedir. Tıbbi veriler içinde tercih edilen bir yöntemdir (Michie ve ark., 1994).

Bayes öğrenme algoritmaları, hipotezler için olasılıklar hesapladığı için NB sınıfandırıcıları belirli tipteki öğrenme problemleri için en uygun yaklaşımlar arasındadır. Örneğin Michie ve ark. (1994), çalışmalarında karar ağaçları ve sinir ağları algoritmaları ile NB sınıflayıcısını karşılaştıran detaylı bir çalışma yapmıştır. Bu çalışmada NB sınıflandırıcıların bu öğrenme algoritmaları ile rekabet edebilecek bir algoritma olduğu gösterilmiştir (Mitchell, 1997).

Bayes Kuralı

Herhangi bir örnek uzayında A ve B iki olay olmak üzere B olayının bilinmesi durumunda A olayının gerçekleşme olasılığı Eşitlik 3.23, A olayı biliniyorken B olayının gerçekleşme olasılığı Eşitlik 3.24 ile verilmiştir (Demirci, 2016);

$$P(A/B) = \frac{P(A \cap B)}{P(B)}, P(B) > 0 \quad (3.23)$$

$$P(B/A) = \frac{P(A \cap B)}{P(A)}, P(A) > 0 \quad (3.24)$$

Rastgele bir A olayının herhangi bir olaydan bağımsız olarak gerçekleşme olasılığını göstermek üzere $P(A)$ notasyonu A olayının olasılığını gösterir. Bu ifade önsel olasılık, koşulsuz olasılık veya marjinal olasılık isimleriyle kullanılabilir. Eşitlik 3.23 ve 3.24'de kullanılan $P(A/B)$ ve $P(B/A)$ gösterimi koşullu olasılığı göstermektedir. Birbirinden bağımsız ve rastgele iki olayın (A ve B) birbiri ardı sıra gerçekleştiği durumlarda bu iki olaydan birinin gerçekleşmesi durumunda ikinci olayın gerçekleşme olasılığı P(A,B) veya P(B,A) ya da $P(A \cap B)$ ifadesi ile gösterilebilir. Değişme özelliği sayesinde çarpım kuralı Eşitlik 3.25'de gösterildiği gibidir (Orhan, 2012);

$$P(A \cap B) = P(B)P(A/B) = P(A)P(B/A) \quad (3.25)$$

Eğer $(A_1, A_2, \ldots, A_k)$ oluşan bir örneklem uzayı için $P(B) > 0$, B olayı biliniyorken herhangi bir A_i olayının Bayes teoremine göre gerçekleşme olasılığı Eşitlik 3.26 ile verilmiştir. $P(A_i/B)$ sonsal olasılığı göstermektedir (Menzel, 2009);

$$P(A_i/B) = \frac{P(A_i)P(B/A_i)}{P(B)} = \frac{P(A_i)P(B/A_i)}{\sum_{j\geq 1}^{k} P(A_j)P(B/A_j)} \quad (3.26)$$

NB sınıflandırıcı için T öğrenme kümesinde bulunan her örnek n boyutlu uzayda tanımlı olmak üzere, $X = (x_1, x_2, .., x_n)$ olsun. Veri kümesinde m adet sınıf bulunuyor olsun. Bu sınıflar $C_1, C_2, \ldots, C_m$ şeklindedir. Bu Bayes teoremine göre Eşitlik 3.27 ile gösterilir. Burada $P(X)$ örnek girdi vektörünün gözlemlenmesi olasılığını, $P(C_i/X)$ hipotezinin kabul edilebilir (doğru) olma olasılığıdır (Mail, 2018);

$$P(C_i\ /X) = \frac{P(X\ /C_i)P(C_i)}{P(X)} \quad (3.27)$$

Eşitlik 3.27'de $P(X)$ değeri tüm sınıflar için aynı olduğu için ve olasılık değerlerin göreceli değerleri etkilenmediğinden ihmal edilebilir. Verilen bir sınıfa göre niteliklerin birbirinden bağımsız olduğunu kabul edersek, $P(X/C_i)$ için bu durumda Eşitlik 3.28 ile elde edilebilir. Ancak normalde nitelikler birbirinden bağımsız değildir, bu nedenle bu algoritma *naive* adını almıştır (Domingos ve Pazzani, 1997).

$$P(X/C_i) = \prod_{k=1}^{n} P(x_k \setminus C_i) \quad (3.28)$$

X' i sınıflandırmak için Eşitlik 3.28 içinde yer alan paydalar birbirine eşit olduğu için sadece pay değerleri karşılaştırılabilir. Bu değerler içinde en büyük olanı Eşitlik 3.29'de verilmiştir. Bu eşitlik sonsal olasılıkları kullanır bu da *En Büyük Sonsal Sınıflandırma* [Maximum A Posteriori Classification (MAP)] olarak bilinir. Bayes sınıflandırıcısı olarak Eşitlik 3.29 kullanılır (Olgun ve Özdemir, 2012);

$$C_{MAP} = argmax_{c_i} = \{P(X/C_i)P(C_i)\} \quad (3.29)$$

Veri yapısı kategorik olduğunda değişkenlerin birbirinden bağımsız olduğu formül geçerli olmaktadır. Ancak sürekli veri yapısı olduğunda bu formül geçerli olmamaktadır. Sürekli değişkenler için mümkün değer sayısı sonsuz olduğunda, tüm sınıflamanın sonsal olasılıkların sıfır olma ihtimali vardır çünkü eğitim setindeki hiçbir niteliğin gelecekteki niteliklerle aynı sürekli değere sahip olması mümkün değildir. Bu sebeple sürekli değişkenler için farklı bir formül gerekmektedir. NB sınıflayıcısı değişkenler sürekli yapıya sahip olduğunda değerlerin normal dağıldığını varsayar. Sürekli değişkenler normal dağıldığında ortalama μ , varyans da σ için Eşitlik 3.30'da verilen formül kullanılır. Aşağıdaki formül normal dağılım için olasılık yoğunluk fonksiyonunu Gaussian göstermektedir. Bu modele göre eğitim setinden az sayıdaki parametre tahmini yapılabilir (John,1995).

$$P(x_k / C_i) = \frac{1}{\sqrt{2\pi\sigma_{C_i}^2}} exp - \frac{(x_k - \mu_{C_i})^2}{2\sigma_{C_i}^2} \quad (3.30)$$

3.4.4. Destek Vektör Makineleri

DVM farklı örüntüleri (yani doğrusal ve doğrusal olmayan örüntüleri) sınıflamak amacıyla kullanılır. Doğrusal örüntüler düşük boyutlarda kolayca ayrılabilen kalıplardır, yüksek boyutlu örüntüler ise kolayca ayırt edilemezler. Bu tür örüntüler bir takım fonksiyonlarla dönüştürülerek ayırt edilebilecek hale getirilirler. DVM'nin arkasındaki ana fikir, doğrusal olarak ayrılabilen modeller için sınıflandırmada kullanılabilecek optimal bir çoklu düzlemin elde edilmesidir. Elde edilen çoklu düzlemler içerisinde marji'ni maksimize eden çoklu düzlem seçilir. Bu algoritmanın ana amacı marji'nin maksimize edilmesi ile doğru sınıflandırmanın yapılmasıdır (Pradhan, 2012). Doğrusal olmayan uzayda ise örneklerin doğrusal olarak ayrılabileceği bir yüksek boyuta geçirerek, farklı örnekler için maksimum sınırın bulunmasına dayanmaktadır (Bayer ve Çoban, 2015).

Girdi ve çıktı değişken çiftlerimiz $Z = \{(x_1, y_1), (x_2, y_2), \dots, (x_l, y_l)\}$ olmak üzere, girdi vektörleri, $x \in X$ ve $y \in Y$ etiketleri ile eşlenmektedir. İkili sınıflamada etiketler $Y = \{-1,1\}$ şeklindedir. Amaç yeni (x, y) örneklerini doğru sınıflayan $f \in F$ sınıflayıcılarını bulmaktır yani $f(x) = y$ (Howley ve Madden, 2004). Farklı sınıf üyelikleri arasındaki nesneleri ayırmak için ayırma çizgilerinin çizilmesine dayanan sınıflandırma işlevi, çoklu düzlem olarak bilinir. DVM çoklu düzlemlere göre belirlenmektedir. Bu çoklu düzlemler farklı sınıf üyeliklerine sahip nesneleri birbirinden ayırmaktadır (Statsoft, 2018).

Çoklu düzlem iki sorun oluşturmaktadır. İlk sorun, verilen örnek veri seti için birden fazla çoklu düzlem elde edilmesidir. Sınıfları en iyi ayıran çoklu düzlemler içerisinden maksimum marjin'e sahip olan düzlem, optimal çoklu düzlemdir. Eşitlik 3.31 de yeni bir x_i'i sınıflamak üzere çoklu düzlem çözümü bulunmaktadır (Howley ve Madden, 2004).

$$f(x) = \langle w.x_i \rangle + b \quad (3.31)$$

Burada $\langle w.x_i \rangle$ ağırlıklandırılmış vektör w ile girdi örneğinin iç çarpımı, b ise yanlılık (bias) miktarıdır. w'nin her elemanın değeri sınıflama için sahip oldukları nisbi

değerinin bir ölçüsünü göstermektedir. Optimal çoklu düzlem için aşağıda ikinci dereceden kısıtlı optimizasyon problemi Eşitlik 3.32 ve 3.33 ile yer almaktadır (Howley ve Madden, 2004).

$$Minimize = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}\xi_i \tag{3.32}$$

$$k.s \begin{cases} y_i(w^T x_i + b) \geq 1 - \xi_i \\ \xi_i \geq 0, i = 1, \dots, l \end{cases} \tag{3.33}$$

Optimizasyon probleminin amacı bir *x* değişkenine göre *f* fonksiyonunun minimize ya da maksimize edilmesidir. Doğrusal olarak ayrılabilen veriler olduğunda DVM algoritmasında optimal çoklu düzlemin bulunması için geometrik marjin (M) en büyük olandır (Kowalczyk, 2017). Optimizasyon problemi w'nin normunu minimize ederek yassılığı artırmaktadır, böylece genelleşme özelliği artırılmaktadır. Sert marjin ile $\langle w.w \rangle$ için minimum değerinin bulunması gerçekleşirken böylece $f(x)$ hiper düzlemi eğitim verisindeki *l* örnek veriyi doğru bir şekilde sınıflamış olacaktır. Gevşek değişken ξ_i olmasının sebebi ise bazı örnekleri yanlış sınıflayan hiper düzlemin bulunmasıdır. Çünkü bazı veri setleri doğrusal olarak ayrılamamaktadır burada yumuşak marjin kullanılmaktadır. Dual problemin çözümü Eşitlik 3.34'de verilmiştir. Problemin kısıtı ise Eşitlik 3.35'de yer almaktadır (Howley ve Madden, 2004).

Dual problemi

$$W(a) = \sum_i^l \alpha_i - \frac{1}{2}\sum_i^l \alpha_i \alpha_j y_i y_j \langle x_i . x_j \rangle \tag{3.34}$$

$$k.s\ C \geq \alpha_i \geq 0, i = 1, \dots, l \ \ \sum_i^l \alpha_i y_i = 0 \tag{3.35}$$

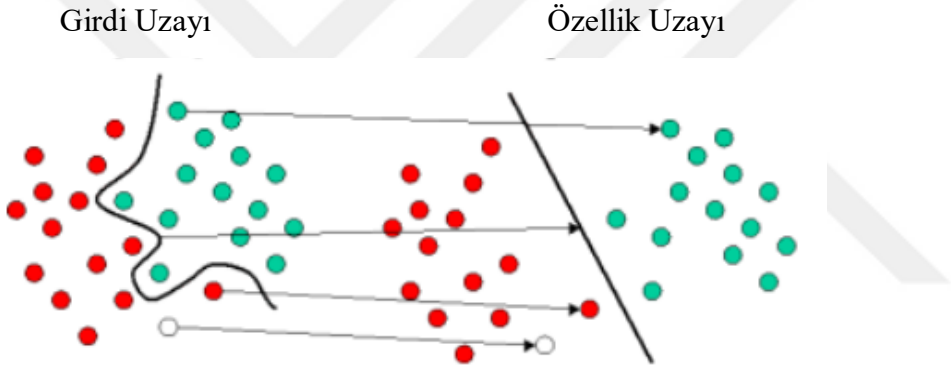
Burada α_i ikili problemin çözümüdür ve karar fonksiyonu Eşitlik 3.36'te gibidir.

$$f(x) = \sum_{i=1}^{l} \alpha_i y_i \langle x . x_i \rangle + b \tag{3.36}$$

Burada b yanlılık miktarı, y_i için $f(x) = 1$ herhangi i için $C > \alpha_i > 0$ olacaktır. Bu durumda yeni bir x_s, $f(x_s)$ sıfırdan küçükse negatif olarak sınıflanırken $f(x_s)$ sıfırdan büyük ya da eşitse o halde pozitif olarak sınıflanmaktadır. Örneklemdeki x_i'ler ve eşlenikleri olan α_i sıfırdan farklı ise bunlar destek vektörleri olarak bilinmektedir ve

hiper düzleme çok yakın bir konumda bulunmaktadırlar. Destek vektör olmayan örnekler ise karar fonksiyonu üzerinde herhangi bir etkileri bulunmamaktadır (Howley ve Madden, 2004).

Problemlerden ikincisi, verinin doğrusal olarak ayrılmaması ve bu durumda girdi uzayındaki nesnelerin birtakım matematik fonksiyonları kullanılarak yeniden düzenlenmesidir. Aşağıda Şekil 9'da bu işlem gösterilmiştir. Matematik fonksiyonları yardımıyla yapılan bu işlem dönüşüm adını almaktadır. Özellik uzayında düzenlenen nesneler doğrusal olarak ayrılabilir hale getirilmektedir. Bu dönüşüm çekirdek (kernel) adını almaktadır (Statsoft, 2018). DVM doğrusal özellik uzayındaki veriyi ayırabildiği için çekirdek fonksiyonun görevi verilerin doğrusal olarak ayrılabildiği daha yüksek bir boyuta eşlemektir (Hofmann, 2006).



Şekil 9.DVM eşlenmenin nasıl gerçekleştiğini gösterir (Statsoft, 2018'den uyarlanmıştır)

Burada öğrenme özellik uzayında gerçekleşir ve veri noktaları diğer noktalar ile iç çarpım şeklindedir. Bu *çekirdek hilesi* adını almaktadır. Bu $\Phi: X \rightarrow H$ gösterimi kullanılırsa iç çarpım $\Phi(x)\Phi(x')$ dönüşür, çekirdek fonksiyonu k için Eşitlik 3.37'de gösterildiği gibidir (Karatzoglou ve ark.,2006).

$$k(x, x') = \langle \Phi(x)\Phi(x') \rangle \tag{3.37}$$

Bu gösterim, x ve x' için H özellik uzayındaki gösteriminden daha fazla tercih edilmektedir (Karatzoglou ve ark., 2006).

Çekirdek kavramı makine öğrenme alanına büyük marjin sınıflayıcısı olarak bilinmektedir. Sabit doğrusal olmayan özellik uzayında modelimiz için eşleme $\Phi(x)$

olmak üzere çekirdek fonksiyonu, Eşitlik 3.38'deki gibidir. Bir çekirdek fonksiyonun en basit örneği ise $\Phi(x) = x$ olmak üzere $k(x, x') = x^T x'$ dir, buna da doğrusal çekirdek fonksiyonu adı verilmektedir (Bishop, 2006).

$$k(x, x') = \Phi(x)^{\mathrm{T}} \Phi(x') \tag{3.38}$$

Çekirdek fonksiyonu veriyi istenilen formata dönüştürmektedir. Farklı DVM algoritmaları için farklı çekirdekler kullanılmaktadır. Bunlar polinom, radial temelli fonksiyon (RTF) ve sigmoid çekirdektir. En çok kullanılan çekirdek ise RTF'dir. DVM için en sık kullanılan çekirdekler aşağıda verilmiştir (Data Flair, 2018; Statsoft, 2018, Hasan ve ark., 2016).

Polinom Çekirdek: Çekirdek doğrusal olarak ayrılamayan DVM için kullanılan genel bir yöntemdir. Sınıflamada çekirdek ile eğitilmiş doğrusal olmayan veri tıpkı doğrusal olarak sınıflanabilen veri gibi işlem görür ancak burada destek vektörleri bulunmaktadır. Dolayısıyla destek vektörleri yüksek boyutta vektörlerin iç çarpımını hesaplamada kullanılmaktadır. Sınıflandırma denklemi modeldeki tüm destek vektörlerinin toplamı haline gelir (Barnett ve ark., 2017). Polinom çekirdek Eşitlik 3.39 ile verilmiştir. Bu daha çok görüntü işlemede kullanılmaktadır. Çok tercih edilen bir fonksiyondur (Alpaydın, 2010).

$$k(x^t, x) = (x^T x^t + 1)^q \tag{3.39}$$

Burada q, polinomun derecesini gösterir ve kullanıcı tarafından seçilmektedir. Polinom çekirdek kullanıldığında girdi uzayında eğriler de bulunmaktadır (Brownlee, 2016).

Gaussian Çekirdek: Veri hakkında herhangi bir ön bilgi olmadığında kullanılan genel amaçlı bir fonksiyondur ve Eşitlik 3.40'da gösterilmiştir. Doğrusal olmayan çoklu düzlem için kullanılmaktadır (Data Flair, 2018).

$$k(x, y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right) \tag{3.40}$$

Gaussian Yarıçap Temelli Fonksiyon (RTF): Aşağıda verilen Eşitlik 3.41'de x^t, merkezi gösterir. $\gamma > 0$ ve $\gamma = \frac{1}{2}\sigma^2$ olacak şekilde de kullanılır (Alpaydın, 2010; Data Flair, 2018).

$$k(x^t, x) = \exp\left[-\gamma\|x^t - x\|\right] \tag{3.41}$$

Laplace RTF Çekirdek: Veri hakkında herhangi bir ön bilgi yoksa kullanılan genel amaçlı laplas çekirdek fonksiyonu Eşitlik 3.42'de verilmiştir (Data Flair, 2018).

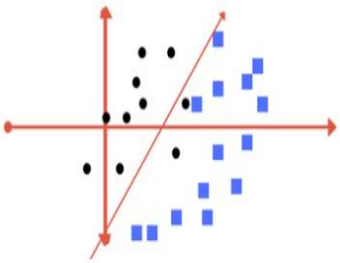
$$k(x, y) = \exp\left(-\frac{\|x-y\|^2}{\sigma}\right) \quad (3.42)$$

Sigmoid Çekirdek: Aşağıdaki Eşitlik 3.43'de sigmoid çekirdek verilmiştir. tanh sigmoid ile benzer şekildedir tek farkı ise $(-1; 1)$ arasında yer almaktadır (Alpaydın, 2010; Data Flair, 2018).

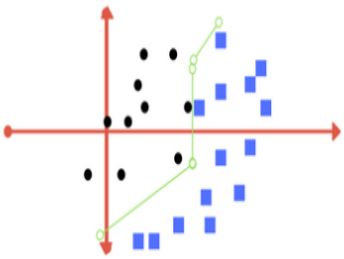
$$k(x^t, x) = \tanh\left(2\, x^T x^t + 1\right) \quad (3.43)$$

Çekirdek fonksiyonlarının parametreleri karar fonksiyonları üzerine etkilidir. Polinom çekirdeğin derecesi, Gaussian ya da Laplace çekirdek için kullanılan σ genişlik parametresi, elde edilen sınıflayıcının esnekliğini kontrol eder. Özellikler arasında doğrusal olmayan bir ilişki olduğunda en düşük dereceli (yani doğrusal) polinom yeterli olmamaktadır. Yüksek dereceli polinomlar iki sınıf arasında ayrımda bulunurken daha esnek olacaktır ve marjin büyüyecek ve yumuşak marjin sabiti için sabit değer daha eğimli olacaktır. Gaussian yada Laplace çekirdek bir sabit değer için yumuşak marjin sabiti σ'nin büyük değeri için karar fonksiyonları doğrusala yakın olacaktır. σ değeri azaldıkça karar fonksiyonun esnekliği artacaktır ve σ'nın küçük değerleri aşırı uyuma sebep olacaktır. Bir problem için öncelikle doğrusal çekirdek uygulanmalıdır daha sonra diğer çekirdek fonksiyonların performansları incelenmelidir (Hasan ve ark., 2016).

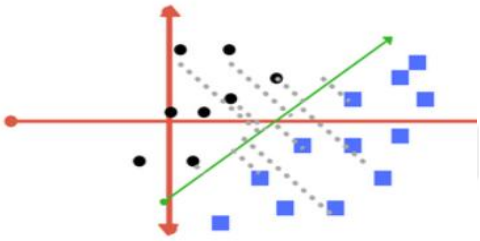
DVM'nin doğruluğunu arttırmak için normalleştirme parametreleri bulunmaktadır. Bu sayede hatalı sınıflama yapma oranını azaltmak mümkündür. Bu parametrelerden biri *C*'dir. *C*'nin büyük değerleri kullanılırsa optimizasyonu elde etmek için küçük marjin değerli çoklu düzlem seçilmektedir (Şekil 10a). Tersine *C*'nin küçük değerleri kullanılırsa optimizasyonu sağlamak için bu çoklu düzlem yanlış sınıflama yapsa dahi büyük marjine sahip çoklu düzlem seçilmektedir (Şekil 10b). Bir diğer parametre gammadır. Gamma değeri ile eğitim verisindeki bir değerin etkisinin ne kadar uzağa gittiğini göstermektedir. Düşük değerli gamma ile ayırma çizgisinin hesaplanmasında ayırma çizgisinden çok uzak noktalar da dikkate alınır (Şekil 10c). Büyük gamma ise hesaplamada çizgiye yakın noktaların hesaba katıldığı anlamına gelir (Şekil 10d) (Patel, 2017; Hasan ve ark., 2016).



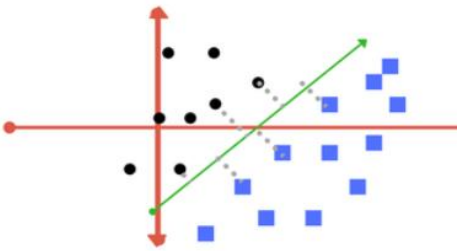
Şekil 10a



Şekil 10b



Şekil 10c



Şekil 10d

Şekil 10a.DVM'nin normalleştirilmesi için kullanılan C parametresi (küçük marjin); Şekil 10b. DVM'nin normalleştirilmesi için kullanılan C parametresi (büyük marjin); Şekil 10c gamma parametresi için uzak noktalar; Şekil 10d gamma parametresi için yakın noktalar (Patel, 2017'den uyarlanmıştır)

Bu algoritmada ilk olarak optimal çoklu düzlem belirlenir. Sonra doğrusal olmayan durum için uygun çekirdek belirlenir. Son olarak veriyi doğrusal olarak ayırabilecek yüksek boyuta çıkararak ayırma yapılır (Sayad, 2018).

3.4.5. Rastgele Orman

Rastgele Orman denetimli öğrenme yöntemleri arasında yer almaktadır. Orman, karar ağaçlarından oluşmaktadır. RO rastgele seçilmiş veri örnekleri için karar ağaçları oluşturmaktadır. Her ağaç için tahminler elde edilerek bu sonuçlar çoğunluk oyuna göre belirlenmektedir (Navlani, 2018).

RO algoritması topluluk modelleri içerisinde yer almaktadır. Genel olarak rastgeleliğe dayanan topluluk yöntemleri temel prensibi, tek bir öğrenme setinden (L) birkaç farklı modelin üretilmesi ve daha sonra bu modellerden elde edilen tahminleri birleştirerek topluluk modelin tahmin edilmesidir (Louppe, 2015). RO, tek bir karar

ağacı üretmek yerine, her biri farklı eğitim kümelerinde eğitilmiş olan aynı dağılımlı çok sayıda karar ağaçların birleştirilmesidir (Breiman, 2001).

RO popüler olmasının nedeni birçok tahmin problemi için çözüm getirmesinin yanı sıra ayarlanacak parametre sayısının az olmasından kaynaklanmaktadır. Uygulanmasının kolay olması, küçük örneklem büyüklüğü, yüksek boyutlu özellik uzayı ve karmaşık veri yapılarında yüksek doğruluk oranı sonucu vermesi başka bir avantajıdır. RO algoritması için önemli iki kriter CART'ın kullandığı ayırma kriteri ve baggingdir. Bagging, orijinal veri setinden alt kümeler seçilip, seçilen bu alt kümelerden bir tahmin edici belirlendikten sonra bunların ortalamasına dayanan bir yöntemdir. Her ağacın düğüm noktası gini saflığına dayanan ayırma kriterini optimize ederek seçilmektedir. Bir süreçte kolaylıkla model belirlenemeyen, karmaşık ve ölçekleme sorunu olan problemler için etkili bir algoritmadır (Scornet ve ark., 2015).

CART ikili karar ağacı algoritmasıdır. Hem kategorik hem de sürekli değişkenleri işleyebilmektedir. CART yinelemeli çalışan bir algoritmadır. Veriyi daha homojen hale getirmek için iki alt kümeye ayırmaktadır. Daha sonra bölünme homojenlik kriteri karşılanana kadar devam eder (Kayri ve Kayri, 2015). Gini indeksi ağacın büyüme fazında düğümlerin bölünmesini gerçekleştirmek için kullanılmaktadır. Gini indeksi, hedef özelliklerin değerlerinin olasılık dağılımları arasındaki farklılığı ölçen, safsızlaştırma temelli bir ölçüttür (Mahmood ve ark., 2011).

Bir T veri setinde n sınıf mevcut ise gini indeksi Eşitlik 3.44 ile hesaplanmaktadır (Paginas, 2008; Kayri ve Kayri, 2015);

$$gini(T) = 1 - \sum_{j=1}^{n} p_j^2 \tag{3.44}$$

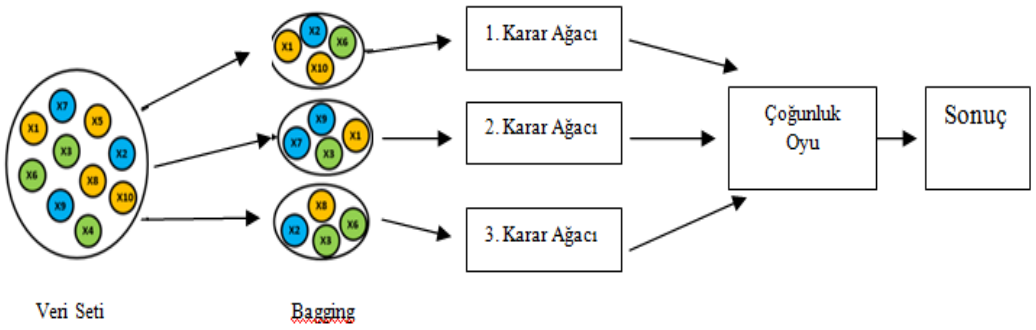
Burada p_j, T veri seti için j.sınıfın göreli frekansıdır.

T veri seti T_1 ve T_2 olmak üzere iki alt kümeye ayrıldığında, büyüklükleri de sırasıyla N_1 ve N_2 olmak üzere, bölünmüş veri için gini indeksi Eşitlik 3.45'deki gibi hesaplanmaktadır. Düğümü en küçük $gini_{\text{bölünmüş}}$ değerine sahip olan özellik seçilir (Paginas, 2008).

$$gini_{\text{bölünmüş}}(T) = \frac{N_1}{N} gini(T_1) + \frac{N_2}{N} gini(T_2) \tag{3.45}$$

RO sınıflayıcısı, gini indeksini özellik seçim kriteri olarak kullanmaktadır. Sınıflara göre bir özelliğin saf miktarını ölçmektedir. RO modelinde maksimum ölçüde olacak şekilde yeni eğitim verisi ile farklı özelliklerin kombinasyonu kullanılarak ağaçlar üretilmektedir. Dallanmış bu ağaçlar budanmazlar. Bu özellik diğer ağaç algoritmalarına göre RO algoritmasının en büyük avantajıdır. Bir ağaç oluşturmak için her düğümde kullanılan özelliklerin sayısı ve oluşturulan ağaçların sayısı RO sınıflayıcısı için kullanıcı tarafından belirlenen parametrelerdir. Her düğümde sadece seçilen özellikler için en iyi bölünme belirlenmektedir. Sonuç olarak RO sınıflayıcısı *N* sayıda ağaçtan meydana gelmektedir, *N* oluşturulması gereken ağaç sayısıdır ve bu sayı kullanıcı tarafından belirlenebilmektedir. Yeni bir veri setini sınıflayabilmek için veri kümelerinin her bir durumu *N* ağaca iletilir (Pal, 2005).

Karar ağaçlarında olduğu gibi RO algoritması da hem sınıflama hem de regresyon yapmak için kullanılmaktadır. RO çok sayıda farklı karar ağaçları oluşturarak bunlardan elde edilen yanıt değişkenlerini belirlemektedir. Esas yanıt oluşturulan bu karar ağaçlarından elde edilen yanıtlara göre belirlenmektedir. Sınıflama yapıldığında en çok tahmin edilen sınıf o nesne için atanan sınıftır. Regresyon için ise bir nesnenin sonuç değeri tüm tahminlerin ortalamasıdır (Horning, 2010).



Şekil 11. Rastgele Orman Algoritması (D'Souza, 2018'den uyarlanmıştır)

Şekil 11'de RO algoritması verilmiş olup, karar ağacının oluşumu ve çoğunluk oyuna göre belirlenişi gösterilmiştir. Ormandaki karar ağaçlarının rastgele oluşturulmasında iki aşama bulunmaktadır. Bu algoritmada öncelikle rastgele örnekler oluşturmak üzere bagging algoritması kullanılmaktadır. Her bir ağacın oluşturulması için verilen eğitim seti içerisinden rastgele (tekrar yerine konarak) seçim yapılarak oluşturulmaktadır. Her ağaç için verilen eğitim setinden bir alt küme kullanılarak bir

karar ağacı modeli oluşturulmaktadır. Eğitim verilerinin kalan üçte biri ise modelin doğruluğunun sınanması için kullanılmaktadır. Test için kullanılan örnek veriler genellikle "küme dışı" veri seti olarak adlandırılır. Küme dışı örneklem kullanılarak hatanın yansız tahmini belirlenmektedir. Veri setinden p sütun sayısı olmak üzere her düğüm için P <<p olacak şekilde sütun seçilmektedir. P sütunu rastgele seçilmektedir. P genelde sınıflama için karekök (p) olarak belirlenmektedir (Saraswat, 2018; Breiman, 2001).

İkinci adım ise ağaçtaki her düğümün ayrılma kriterlerin belirlenmesidir. Ağaçtaki her düğüm için ikili kural uygulanmak üzere tahmin ediciler için bir alt küme rastgele seçilmektedir. Rastgele seçilen tahmin ediciler kullanıcı tarafından ya da RO algoritması tarafından belirlenmektedir. Her düğümün ayrılması için tahmin edici değişkenlerin rastgele seçilmiş bir alt kümesinin kullanılması, ağaçlar arasında daha az korelasyona neden olur ve sonuç olarak daha düşük bir hata oranı elde edilir. Her bir ağaç için tüm değişkenler kullanılırsa, ağaçlar benzer şekilde olacaktır ve bunun sonucunda da daha yüksek bir hata oranı elde edilecektir (Breiman, 2001). Tahmin ediciler için küçük alt kümeleri elde edilirse ağaçlar arasında ilişki azalacaktır ve tahmin gücü, daha fazla tahmin edici kullanılan ağaçlar göre düşük olacaktır. Bu sebeple düşük ilişki veren ancak tahmin gücü yüksek olacak şekilde değişkenlerin sayısı belirlenmelidir (Horning, 2010). Tüm ağaçlar tam olarak oluşturulur. Son olarak birçok ağaç oluşturulur ve tahmin modeli en çok belirlenen sınıfa göre belirlenir. Karar ağaçları yüksek değişkenliğe sahiptirler. Bu durum ise test verisi üzerinde yüksek tahmin hatasına sebep olmaktadır. Eğitim için daha fazla veri kullanarak bu problem çözülebilmektedir. Ancak veri seti sınırlı olduğu için bagging gibi yeniden örnekleme teknikleri kullanılarak yeni veri üretilmektedir (Saraswat, 2018).

RO algoritmasını optimize edebilmek için öncelikle ayarlanması gereken parametre mtry'dir. Mtry düğüm noktasında kaç tane değişkenin seçilmesi gerektiğini göstermektedir. Bunun için varsayılan (default) değeri regresyon için p/3, sınıflama için karekök(p)'dir. Mtry için küçük değerler kullanılması önerilmemektedir böylece aşırı uyum engellenmiş olmaktadır (Scornet, 2018).

3.4.6. Super Learner

Araştırmacılar bir problemi analiz ederken istatistikte veya makine öğrenimi yöntemlerini kullandıklarında farklı algoritmaların olduğunu görmektedirler. Sürekli sayıları artan ve farklı özellikleri bulunan bu algoritmalar içinde veri setine uygun ve araştırma konusunu en iyi açıklayacak algoritmayı belirlemek önceden mümkün değildir. Genelde araştırmacılar bir veya birkaç algoritmayı keyfi olarak seçerek analizi gerçekleştirirler. Ancak elimizdeki problem için daha iyi performans gösteren bir algoritma mevcut olabilir (Bacak ve Kennedy, 2018). Birleştirilmiş kayıp-temelli tahmin sisteminin bu probleme çözümü yeni bir tahmin edici olan super learner'dir (Van der Laan ve Dudoit, 2003).

Van der Laan ve Dudoit (2003), parametre tahmini için genel bir çerçeve oluşturmuşlardır. Bilinmeyen P_0 dağılımına sahip $X_1, \dots, X_n$ değişkenler, aynı dağılımlı n veriden oluşsun. Amaç verileri kullanarak P_0 dağılımı için bir parametre ψ_0 tahmin etmektir. Burada ψ_0 parametresi P_0'ın bir fonksiyonudur. Yani $\hat{\psi}$ tahmin ediciyi, ψ_0 parametresine yakın olacak (risk açısından) şekilde tahmin etmektir. Kayıp temelli tahmin için genel plan öncelikle bir kayıp fonksiyonun belirlenmesidir. Performans değerlendirilmesi ve tahmin edici seçimi ise çapraz geçerlemeye dayanmaktadır (Sinisi ve ark, 2007).

Kayıp fonksiyon genel olarak gösterimi Eşitlik 3.46'da verilmiştir.

$$L: (X, \psi) \rightarrow L(x, \psi) \epsilon \, \mathcal{R} \tag{3.46}$$

Herhangi bir aday parametre değeri ψ ve gözlem X değeri ile bir reel sayıya eşleyen bir fonksiyondur. Bu değerin P_0 için beklenen değeri $\psi_0 = \Psi(P_0)$ parametresi için (riskin) minimum olduğu değerdir (Sinisi ve ark, 2007).

SL için gözlenen eğitim seti $X_i = (Y_i, W_i)$ olmak üzere, burada $i = 1, 2, \dots, n$ dir. Y ilgilenilen çıktı değişkeni ve W, p boyutlu ortak değişkenler kümesidir. Amaç $\psi_0(W) = E(Y \setminus W)$ fonksiyonunu tahmin etmektir. Burada beklenen kayıp fonksiyonun bağlı olarak risk (beklenen kayıp) hata kareler ortalamasına göre elde edilmektedir. Hata kareler ortalaması bu fonksiyonu minimize etmektedir $\psi_0(W) = E(Y \setminus W = w)$. Bu da Eşitlik 3.47'de verilmektedir (Polley ve Van der Laan, 2010).

$$\psi_0(W) = arg \min_{\psi} E[L(X, \psi(W))] \tag{3.47}$$

Burada kayıp fonksiyon L_2: $(Y - \psi(W))^2$ şeklinde verilmektedir.

SL bir takım aday öğreticileri gözlenen veriye uygular ve en optimal öğreticiyi ÇG riskine dayalı olarak seçmektedir. Verilen bir problem için tahmin algoritmaları kütüphanesi bulunmaktadır. Kütüphane toplu halde algoritmaların bulunduğu bir yerdir (Van der Laan ve ark., 2007). R programında yer alan birtakım aday öğreticilerin yer aldığı kütüphane Tablo 4'de verilmiştir (Kennedy, 2017).

Tablo 4. R paketinde bulunan bir takım aday öğreticiler (Kennedy, 2017'den uyarlanmıştır)

[1]	"SL.bartMachine"	"SL.bayesglm"	"SL.biglasso"
[4]	"SL.caret"	"SL.caret.rpart"	"SL.cforest"
[7]	"SL.earth"	"SL.extraTrees"	"SL.gam"
[10]	"SL.gbm"	"SL.glm"	"SL.glm.interaction"
[13]	"SL.glmnet"	"SL.ipredbagg"	"SL.kernelKnn"
[16]	"SL.knn"	"SL.ksvm"	"SL.lda"
[19]	"SL.leekasso"	"SL.lm"	"SL.loess"
[22]	"SL.logreg"	"SL.mean"	"SL.nnet"
[25]	"SL.nnls"	"SL.polymars"	"SL.qda"
[28]	"SL.randomForest"	"SL.ranger"	"SL.ridge"
[31]	"SL.rpart"	"SL.rpartPrune"	"SL.speedglm"
[34]	"SL.speedlm"	"SL.step"	"SL.step.forward"
[37]	"SL.step.interaction"	"SL.stepAIC"	"SL.svm"
[40]	"SL.template"	"SL.xgboost"	

Birçok algoritma mevcut paket programlarında bulunmaktadır. Bunlar SL.KNN (En yakın komşu algoritması), SL.bayesglm (bayes doğrusal fonksiyonu), SL.RO, SL.SVM, SL.rpart (Cart algoritmasının R paketindeki uygulamasıdır) şeklinde çoğaltılabilir. Tek bir algoritma kullanmak istediğimizde bunun yanında birçok analiz seçenekleri bulunmaktadır. Örneğin ayar parametrelerinin belirlenmesi, EYK için en yakın komşuluk sayısının belirlenmesi, DVM için hangi çekirdeğin kullanılması ve RO için kaç değişkenin bölünmesinin ve ağaç sayısı gerektiği gibi. Araştırmacılar için bu seçimleri en iyi şekilde yapmak kolay bir süreç değildir. Veri bu seçimleri nasıl yapabileceğimiz konusunda bilgi verebilir. Bunun için SL ve çapraz geçerliğin kullanılması gerekmektedir (Bacak ve Kennedy, 2018).

Yukarıda bahsedildiği gibi en optimal öğretici ÇG riskine dayalı olarak elde edilmektedir. ÇG veriyi parçalara ayırarak öğrenme algoritmalarını karşılaştırmak ve değerlendirmek için kullanılan istatistiksel bir yöntemdir. Bu parçalardan biri veriyi öğrenmek veya eğitmek için, diğeri ise doğrulamak için kullanır. Bu yöntemde eğitim ve geçerlilik setlerinin her veri noktasının onaylanma şansına sahip olacak şekilde ardışık turlarda çaprazlanması gerekmektedir. Temel yöntem k- katlı çapraz geçerliktir. K- katlı çapraz geçerlikte veriler ilk olarak eşit boyutlarda katlara bölünmektedir. K aşama olacak şekilde eğitim ve geçerlik setleri her aşamada verinin başka bir katı doğrulamak için dışarıda tutulurken geriye kalan *k-1* kat veri eğitim seti olarak kullanılmaktadır. Makine öğreniminde 10 (*k=10*) kat çapraz geçerlik en sık kullanılandır (Refaeilzadeh ve ark., 2009). ÇG tasarıları k- katlı ÇG, birini dışarıda bırakarak ÇG, Monte Carlo ÇG ve bootstrap ÇG'tir (Van der Laan ve Dudoit, 2003).

Birini dışarıda bırak ÇG metodu k-kat çapraz geçerlik yönteminin özel bir durumudur. Burada k değeri verideki örneklerin sayısını (k=n) göstermektedir. Yani her bir tekrarda bir tek veri hariç olmak üzere tüm veriler eğitim için kullanılır ve model bu tek gözlemde test edilir (Refaeilzadeh ve ark., 2009). Monte Carlo ÇG yöntemi (ya da tekrarlı dışarıda tutma yöntemi) dışarıda tutma yönteminin *k* kez farklı rastgele çekirdek kullanılarak tekrarlanması ve bu *k* sonucun ortalama performansıdır. Eşitlik 3.48 ve 3.49'daki gibi hesaplanmaktadır (Raschka, 2018).

$$ACC_{ort.} = \frac{1}{k}\sum_{j=1}^{k} ACC_j \tag{3.48}$$

$$ACC_j = 1 - \frac{1}{m}\sum_{i=1}^{m} L(\hat{y}_i, y_i) \tag{3.49}$$

Burada tahmin doğruluğu *ACC= 1- ERR*, tahmin hatası ise $ERR_s=\frac{1}{m}\sum_{i=1}^{m} L(\hat{y}_i, y_i)$ (m örnekli, S ise veri setini göstermektedir). Dışarıda tutma yönteminde veri seti eğitim ve test seti olarak ayrılmaktadır. Burada test setinde kullanılan veri eğitim setinin dışındaki verilerden oluşmaktadır (Balaban ve Kartal, 2015).

Bootstrap ÇG yönteminde örneklem büyüklüğü *S* olmak üzere yerine koyarak $S_b{}^*$kümeleri oluşturulmaktadır, $b = 1, \dots, B$. Her bootstrap örnekleminde her sınıftan en az üç farklı değerin yer alması gerekmektedir. Her $S_b{}^*$ kümesi için önceden belirlenen sınıflama kuralına göre (örneğin doğrusal ayrıştırma analizi) çapraz geçerlik yapılır ve hata tahmini r_B elde edilir. Bu aşama *B* kez tekrar edilir ve ortalama hata tahmin edilir ($r_{BCV} = B^{-1}\sum_{b=1}^{B} r_B$) (Fu ve ark, 2005).

SL için ÇG seçicisi geçerlilik seti için en iyi performansı veren öğreticiyi belirlemektedir. Bunun için *v* – katlamalı ÇG kullanılır. Her veri seti ve tamamlayıcısı v kat bölünerek sırasıyla geçerlilik ve eğitim setini oluşturmaktadır. Her *v* bölünme için tahmin edici eğitim setine uygulanarak tamamlayıcı olan geçerlik setine göre riskleri tahmin edilmektedir. Her tahmin edici/öğretici için elde edilen risklerin ortalaması alınarak ÇG riski elde edilir. En küçük ÇG riskine sahip tahmin edici seçilmektedir (Sinisi ve ark, 2007).

Kütüphanede ($\mathcal{L}$'de) bulunan her algoritma tüm veri seti üzerinde yani $X = \{X_i : i = 1, \dots, n\}$'ne göre eşlenir ve $\widehat{\Psi_k}(W)$ tahmin edilir ($k = 1, \dots, K(n)$).
Veri seti *X* eğitim ve geçerlik seti olarak yukarıda bahsedildiği şekilde bölünür (yani *v* katlamalı çapraz geçerliğe göre). Burada *v*. grup geçerlilik seti ve geri kalan setler eğitim setidir. Böylece $T(v)$ v. eğitim verisi ve V(v)de tamamlayıcısı olan geçerlilik setidir. $T(v) = X \backslash V(v)$'dir. v. kat için her algoritma kütüphanede ($\mathcal{L}$'de) $T(v)$ göre eşlenir ve tahminler ilişkili geçerlik setinde yer alır, Eşitlik 3.50'de gösterildiği gibidir (Polley ve Van der Laan, 2010);

$$\widehat{\Psi}_{k,T(v)}(W_i), \qquad X_i \in V(v), \qquad v = 1, \dots, V \tag{3.50}$$

Her algoritmadan elde edilen tahminler, *K* matrisini oluşturmak üzere n kez yığılır, Eşitlik 3.51'da *K* matrisi verilmiştir (Polley ve Van der Laan, 2010);

$$Z = \left\{\widehat{\Psi}_{k,T(v)}\left(W_{V(v)}\right), v = 1, \dots, V \;\&\; k = 1, \dots, K\right\} \tag{3.51}$$

$W_{V(v)} = (W_i : X_i \in V(v))$ $V(v)$ geçerlilik kümesi için ortak vektörlerdir.

Aday öğreticilerden ağırlıklandırılmış bir takım kombinasyonlar oluşturulur ve bu α (ağırlıklandırma) vektör ile gösterilmektedir Eşitlik 3.52'de verilmiştir (Polley ve Van der Laan, 2010);

$$m(z/\alpha) = \sum_{k=1}^{K} \alpha_k \widehat{\Psi}_{k,T(v)}\left(W_{V(v)}\right) \qquad \alpha_k \geq 0 \; \forall k, \; \sum_{k=1}^{K}, \; \alpha_k = 1 \tag{3.52}$$

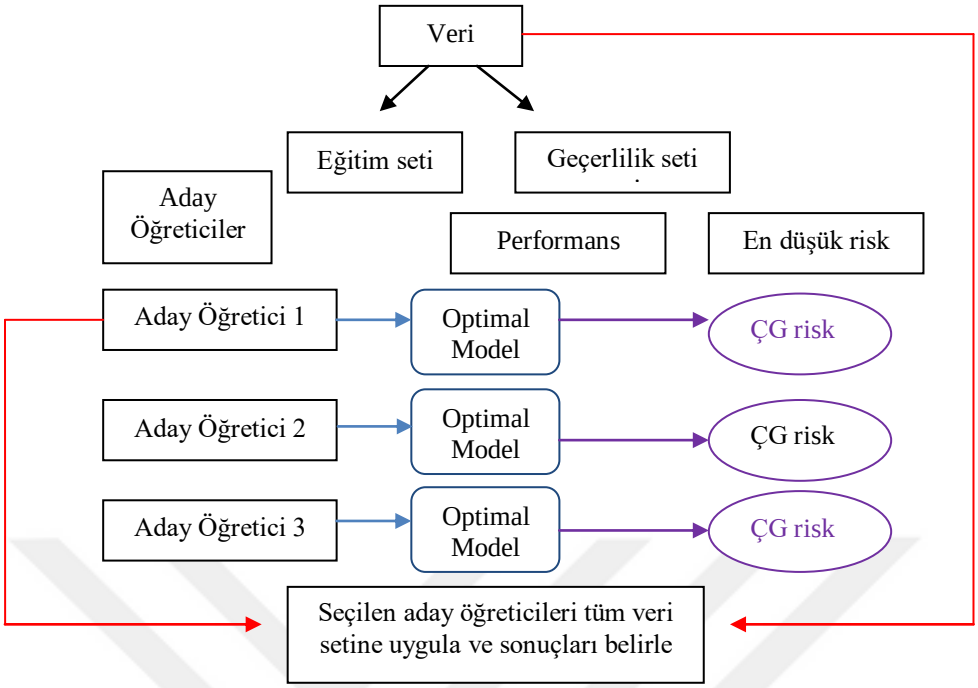
Yukarıda bahsedilen ÇG riskini minimize eden α belirlemek için aday öğreticiler $\sum_{k=1}^{K} \alpha_k \widehat{\Psi}_k$ tüm α kombinasyonları için Eşitlik 3.53'de verilmiştir (Polley ve Van der Laan, 2010);

$$\hat{\alpha} = arg \min_{\alpha} \sum_{i=1}^{n} (Y_i - m(z_i/\alpha))^2 \tag{3.53}$$

Elde edilen $\hat{\alpha}$, tüm kombinasyonları içeren $m(z/\alpha)$ göre $\widehat{\Psi}_k(W), k = 1, \ldots, K$ ile birleştirildiği takdirde SL algoritması Eşitlik 3.54'deki gibi elde edilmektedir (Polley ve Van der Laan , 2010);

$$\widehat{\Psi}_{SL}(W) = \sum_{k=1}^{K} \hat{\alpha}_k \widehat{\Psi}_k(W) \tag{3.54}$$

Aday öğreticiler arasında ağırlıklandırılmış model SL algoritmasının elde edilmesini sağlamaktadır. Eğer tek bir model ile sonuç elde edilirse bu discrete SL algoritmasını vermektedir. Aday öğreticiler için herhangi bir sınırlandırma bulunmamaktadır. Şekil 12'de SL tahmin algoritmasının işleyişi verilmiştir. Birçok optimal öğreticiyi bir arada kullanarak en iyi performansı elde etmesi ve farklı algoritmaları kullanması en önemli avantajıdır (Van der Laan ve ark, 2007). Bu algoritmada öncelikle veri seti, eğitim seti ve geçerlilik seti olmak üzere ikiye ayrılmaktadır. Belirlenen aday öğreticiler eğitim setindeki örneklemlerde eğitilerek optimal model elde edilmektedir. Aday öğreticilerin performansları geçerlilik setinde karşılaştırılarak ortalama değerleri elde edilmektedir. Bunlar içerisinde en küçük riske sahip olan model seçilmektedir (Sinisi ve ark, 2007).



Şekil 12. Super learner algoritması (Sinisi ve ark, 2007'den uyarlanmıştır)

4. BULGULAR

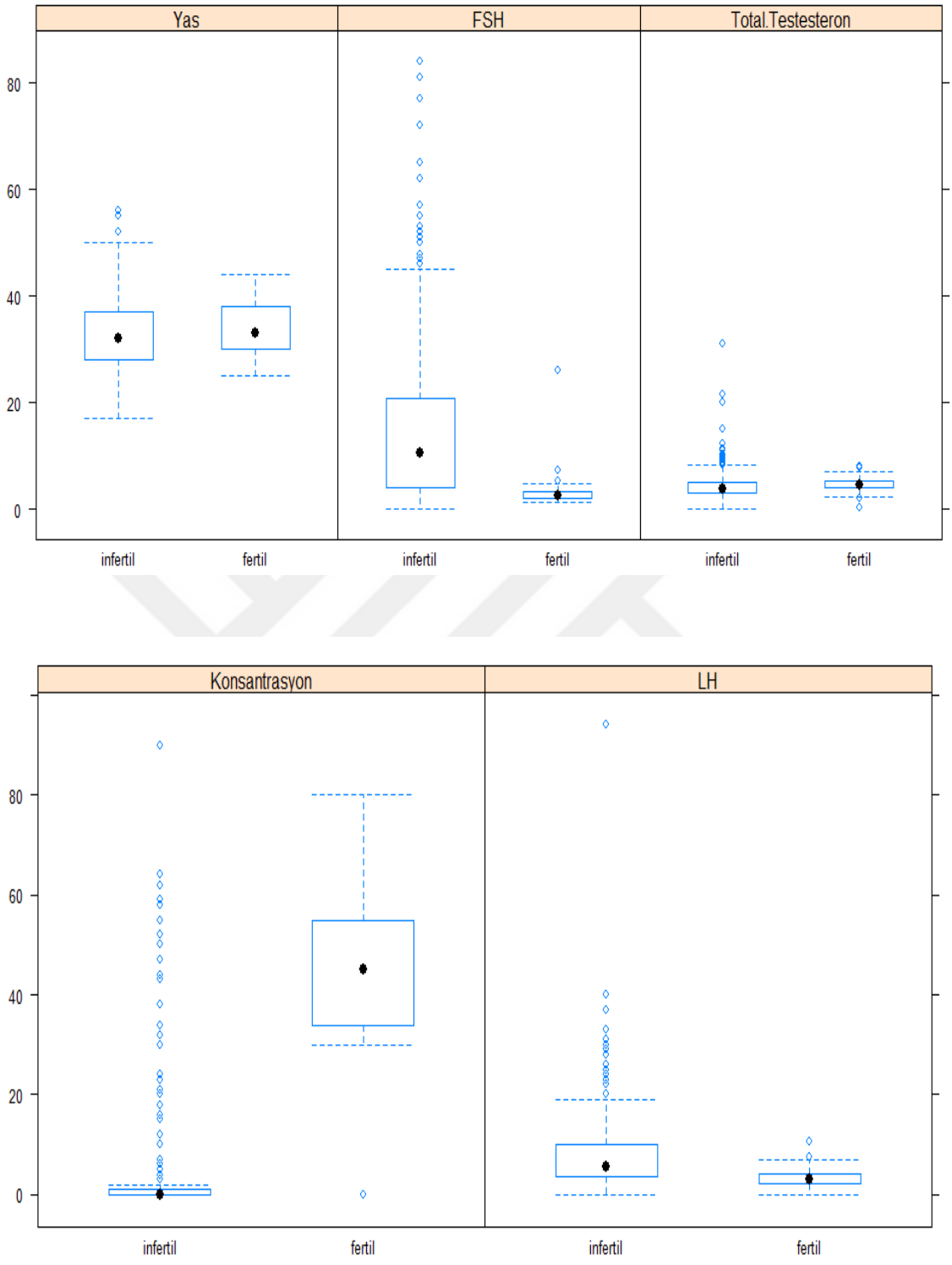
4.1. İnfertilite Verisi için

İnfertilite veri seti başlangıçta 530 infertil ve 57 fertil erkek olmak üzere toplam 587 gözlemden; beş sayısal, beş kategorik ve bir hedef değişkenden (fertil ve infertil) oluşmaktaydı. Sıfıra yakın varyans bulunması sebebi ile gr/gr+b2/b3 (polimorfizm geni) değişkeni ve eksik gözlemler analizden çıkarıldı. Bu işlemlerden sonra analizler 329 (%85,50) infertil, 56 (%15,50) fertil erkek, toplam 385 gözlem ve 10 değişken üzerinde gerçekleştirildi. Değişkenler için özet istatistikler Tablo 5'te verildi.

Tablo 5. İnfertilite verisi için kategorik ve sayısal değişkenlere ait özet istatistikler **Min:** Minimum değer, **AO:** Aritmetik ortalama, **Mak:** Maksimum değer, **Ç1:** I. Çeyrek, **Ç3:** III. Çeyrek

Polimorfizmi	**Sonuç**					
	İnfertil			**Fertil**		
	n (%)			n (%)		
gr/gr	334(86,75)			51(13,25)		
sy1191	330(85,71)			55(14,29)		
b2/b3	330(85,71)			55(14,29)		
sy1291	334(86,75)			51(13,25)		
	Min.	**Ç1**	**Ortanca**	**AO**	**Ç3**	**Mak.**
Yaş	17,00	28,00	32,00	32,85	37,00	56,00
FSH düzeyi	0,10	3,00	6,90	12,28	18,00	84,00
Total Testesteron düzeyi	0,10	3,10	4,00	4,33	5,10	31,00
Sperm Konsantrasyonu	0,00	0,00	0,00	10,34	6,00	90,00
LH düzeyi	0,10	3,00	4,90	7,00	9,20	94,00

Sayısal değişkenlere ait grafikler Şekil 13' te verildi.



Şekil 13. İnfertilite verisine ait sayısal değişkenlerin kutu-çizgi grafikleri

Şekil 13'te sayısal değişkenler için elde edilen grafikte uç noktaların olduğu görülmektedir. Verideki değişkenler için uygun dönüşümler, ardından ön-işleme adımı

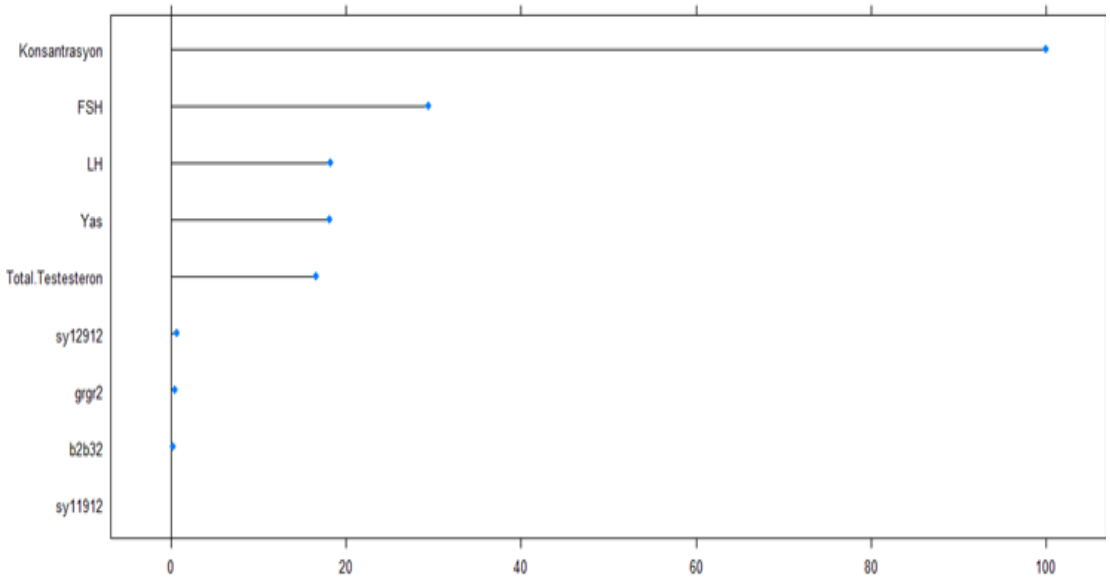
gerçekleştirildi. Eğitim setleri sırasıyla %80, %70 ve %60; test setleri de sırasıyla %20, %30 ve %40 olarak ayrıldı. Analiz öncesi aynı sonuçları elde etmek amacıyla her seferinde set.seed (1234) komutu kullanıldı. Rastgelelik içeren işlemlerde herkesin aynı sonucu bulabilmesi için set.seed kullanıldı. Makine Öğrenimi algoritmaları C4.5, EYK, NB, DVM ve RO ile elde edilen sonuçlar Tablo 6' da verildi.

Tablo 6**.** İnfertilite verisine ait algoritma sonuçları

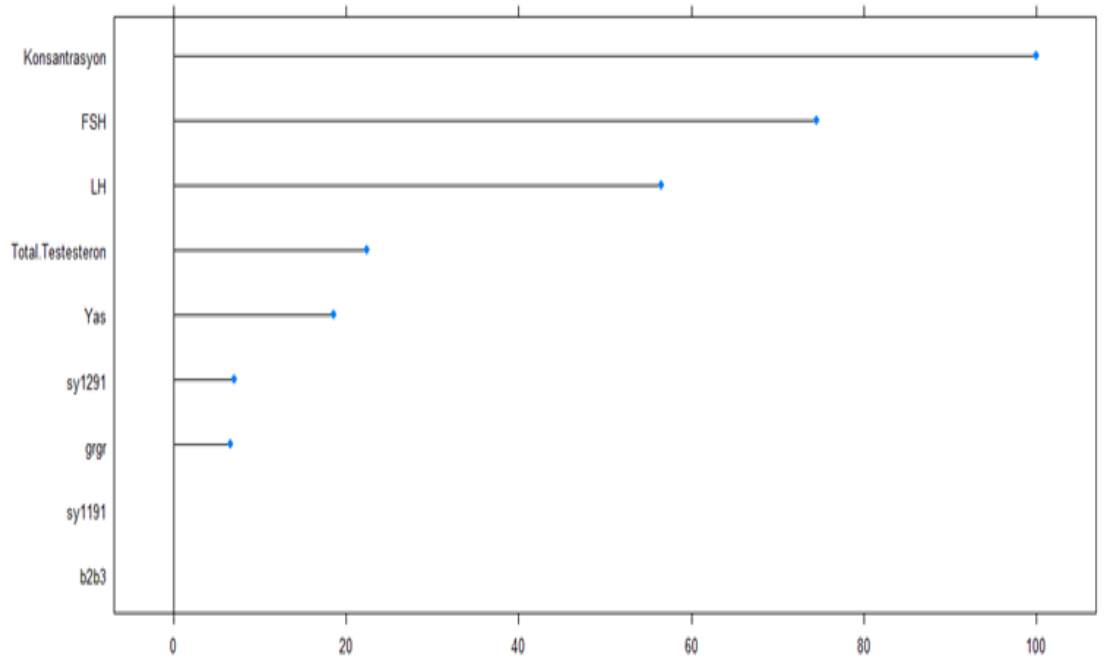
	Performans Ölçüleri	**C4.5**	**EYK**	**NB**	**DVM**	**RO**
%80 -%20	**Doğruluk Oranı**	0,9342	0,9079	0,8947	0,9342	**0,9605**
	Duyarlılık	0,9385	0,9385	0,8923	0,9692	0,9846
	Seçicilik	0,9091	0,7273	0,9091	0,7273	0,8182
	Pozitif Kestirim Değeri	0,9839	0,9531	0,9831	0,9545	0,9697
	Negatif Kestirim Değeri	0,7143	0,6667	0,5882	0,8000	0,9000
	EAA	0,9244	0,9069	0,8727	**0,9594**	0,9209
%70-%30	**Doğruluk Oranı**	0,9123	0,9035	0,8596	**0,9561**	0,9386
	Duyarlılık	0,9082	0,9388	0,8469	0,9694	0,9694
	Seçicilik	0,9375	0,6875	0,9375	0,8750	0,7500
	Pozitif Kestirim Değeri	0,9889	0,9485	0,9881	0,9794	0,9596
	Negatif Kestirim Değeri	0,6250	0,6471	0,5000	0,8235	0,8000
	EAA	0,9237	0,9435	0,8903	**0,9534**	0,9298
%60- %40	**Doğruluk Oranı**	0,8954	0,8824	0,8889	0,8889	**0,9346**
	Duyarlılık	0,8855	0,9313	0,8779	0,9618	0,9695
	Seçicilik	0,9545	0,5909	0,9545	0,4545	0,7273
	Pozitif Kestirim Değeri	0,9915	0,9313	0,9914	0,9130	0,9549
	Negatif Kestirim Değeri	0,5833	0,5909	0,5676	0,6667	0,8000
	EAA	0,9200	0,9221	0,9302	0,9323	**0,9458**

Veri seti %80- %20 olarak ayrıldığında doğruluk oranı, DVM ve C4.5'de %93,42; EYK'de %90,79 ve NB'de %89,47 olarak elde edildi. En iyi performans gösteren algoritma olan RO'da doğruluk oranı %96,05 şeklindeydi. Veri seti %70- %30 olarak ayrıldığında doğruluk oranı, C4.5'de %91,23; EYK'de %90,35; NB'de %85,96 ve RO'da %93,86 şeklinde iken, en iyi performans gösteren algoritma DVM'de %95,61 olarak elde edildi. Veri seti %60- %40 olarak ayrıldığında doğruluk oranı, C4.5'de %89,54; EYK'de %88,24; DVM ile NB'de %88,89 şeklinde saptandı. En iyi performans gösteren algoritma RO'da doğruluk oranı %93,46' di.

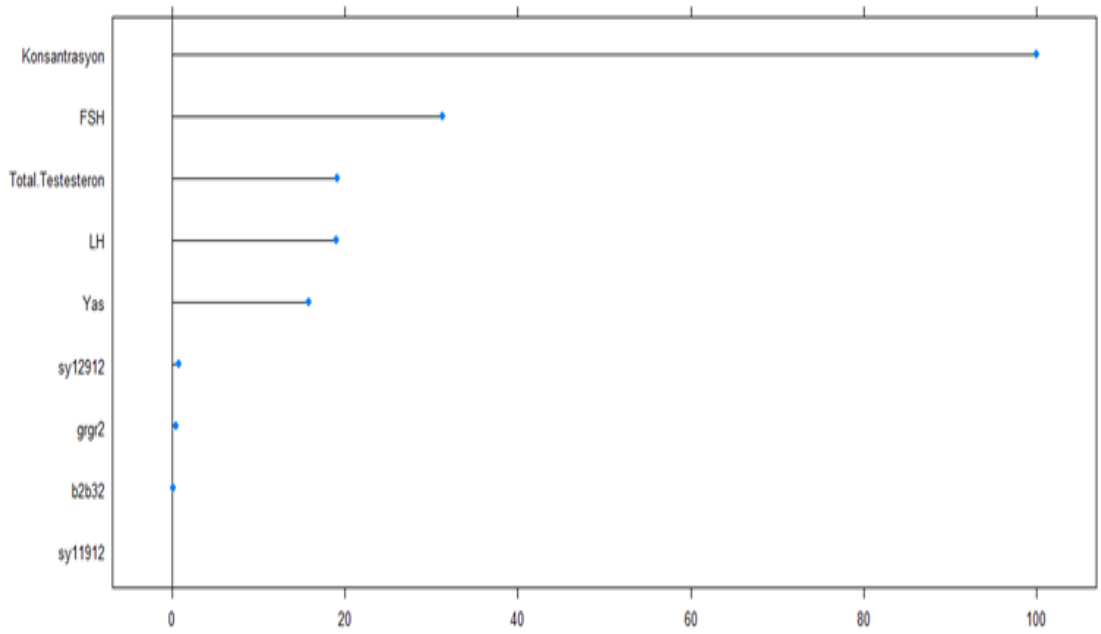
En iyi performans gösteren algoritmalara ait farklı eğitim ve test bölmelerine göre değişkenlerin önemliliği aşağıdaki şekillerde (Şekil 14-16) verildi.



Şekil 14. Veri seti %80-%20 bölündüğünde DVM için değişkenlerin önemlilik göstergeleri



Şekil 15. Veri seti %70-%30 bölündüğünde DVM için değişkenlerin önemlilik göstergeleri



Şekil 16. Veri seti %60-%40 bölündüğünde RO için değişkenlerin önemlilik göstergeleri

Farklı eğitim ve test setlerine göre önemlilik incelendiğinde, üç farklı bölme oranı için ilk ve ikinci sırada yer alan değişkenler, konsantrasyon ile FSH iken, %80-%20 ile %70-%30 bölünmede üçüncü önemli değişken LH ve %60-%40 bölünmede total testosterondur. Polimorfizm genleri incelendiğinde en etkili olan sy1291 olup, bunu gr/gr ve b2/b3 genleri takip etmektedir.

Verideki infertil, fertil bireylerin oranı sırasıyla %85,50 ve %14,50 olduğu için bu durum dengesiz veri seti problemini çağrıştırmaktadır. Dengesiz veri setini dengelemek için eğitim ve test seti literatürde önerilen yöntemlere göre analiz edildi. Veri seti %80- %20 bölündüğünde 264 (%85,40) infertil ve 45 (%14,60) fertil erkek bulunmaktadır. Tablo 7'de dengesiz veri seti için önerilen farklı örnekleme sonuçları bulunmaktadır.

Tablo 7. İnfertilite verisi %80 - %20 bölündüğünde sınıf dengesizliği için gözlem sayıları

	İnfertil	**Fertil**
Aşağı örnekleme	45	45
Yukarı örnekleme	264	264
SMOTE	180	135
ROSE	155	154

Verilen örnekleme yöntemlerine göre elde edilen eğitim seti analiz sonuçları Tablo 8'de verildi.

Tablo 8. İnfertilite verisi %80 - %20 bölündüğünde sınıf dengesizliği eğitim performansı için EAA sonuçları **Min:** Minimum değer, **AO:** Aritmetik ortalama, **Mak:** Maksimum değer, **Ç1:** I. Çeyrek, **Ç3:** III. Çeyrek

		Min.	Ç1	Ortanca	AO	Ç3	Mak.
%80 -%20	**Orijinal - C4.5**	0,8222	0,9337	0,9629	0,9571	0,9842	1
	Orijinal - EYK	0,8269	0,9448	0,9615	0,9555	0,9769	1
	Orijinal - NB	0,8666	0,9394	0,9692	**0,9586**	0,9813	1
	Orijinal- DVM	0,8692	0,9307	0,9615	0,9570	0,9975	1
	Orijinal - RO	0,8461	0,9337	0,9615	0,9558	0,9807	1
	AÖ - C4.5	0,8800	1	1	**0,9908**	1	1
	AÖ - EYK	0,8125	1	1	0,9842	1	1
	AÖ -NB	0,8500	1	1	0,9900	1	1
	AÖ - DVM	0,8500	1	1	0,9851	1	1
	AÖ - RO	0,8400	1	1	0,9884	1	1
	YÖ - C4.5	0,9700	0,9970	1	0,9962	1	1
	YÖ - EYK	0,9691	0,9933	1	0,9956	1	1
	YÖ -NB	0,9729	0,9943	0,9985	0,9954	1	1
	YÖ - DVM	0,9672	0,9971	1	**0,9963**	1	1
	YÖ - RO	0,9672	0,9960	0,9985	0,9952	1	1
	ROSE – C4.5	0,9375	0,9926	0,9980	**0,9903**	1	1
	ROSE - EYK	0,9291	0,9825	0,9959	0,9880	1	1
	ROSE -NB	0,9250	0,9833	0,9916	0,9877	1	1
	ROSE- DVM	0,9335	0,9866	0,9956	0,9887	1	1
	ROSE - RO	0,9700	0,9849	0,9938	0,9879	1	1
	SMOTE – C4.5	0,9337	0,9851	1	0,9877	1	1
	SMOTE - EYK	0,9401	0,9841	0,9980	0,9884	1	1
	SMOTE -NB	0,9273	0,9841	0,9978	0,9877	1	1
	SMOTE - DVM	0,9444	0,9848	0,9960	**0,9891**	1	1
	SMOTE - RO	0,9380	0,9841	1	0,9883	1	1

Eğitim için en iyi performanslar incelendiğinde veri seti %80-%20 olarak ayrıldığında orijinal veri seti için NB'de % 95,86 EAA elde edildi. AÖ- C4.5'de % 99,08 EAA değeri, YÖ-DVM'de %99,63, ROSE- C4.5'de %99,03 ve SMOTE- DVM'de %98,91 elde edildi. Burada YÖ'nün genel olarak diğerlerinden çok daha iyi performans gösterdiği saptandı.

Veri seti %70- %30 bölündüğünde hedef değişkeni 231 (%85,20) infertil ve 40 (%14,70) fertil erkekten oluşmaktaydı. Aşağıda verilen Tablo 9'da dengesiz veri için önerilen farklı örnekleme sonuçları bulunmaktadır.

Tablo 9. İnfertilite verisi %70 - %30 bölündüğünde sınıf dengesizliği gözlem sayıları

	İnfertil	**Fertil**
Aşağı örnekleme	40	40
Yukarı örnekleme	231	231
SMOTE	160	120
ROSE	138	133

Buna göre elde edilen eğitim seti analiz sonuçları Tablo 10'da verildi.

Tablo 10. İnfertilite verisi %70 - %30 bölündüğünde sınıf dengesizliği eğitim performansı için EAA sonuçları **Min:** Minimum değer, **AO:** Aritmetik ortalama, **Mak:** Maksimum değer, **Ç1:** I. Çeyrek, **Ç3:** III. Çeyrek

		Min.	Ç1	Ortanca	AO	Ç3	Mak.
%70- %30	**Orijinal - C4.5**	0,8804	0,9592	0,9841	0,9685	1	1
	Orijinal - EYK	0,8369	0,9673	0,9945	0,9706	1	1
	Orijinal - NB	0,9293	0,9673	0,9732	0,9723	0,9891	0,9891
	Orijinal - DVM	0,9130	0,9711	0,9836	**0,9729**	0,9972	1
	Orijinal - RO	0,8152	0,9497	1	0,9657	1	1
	AÖ - C4.5	0,9687	1	1	0,9968	1	1
	AÖ - EYK	0,9375	1	1	0,9937	1	1
	AÖ -NB	1	1	1	**1**	1	1
	AÖ - DVM	0,9375	1	1	0,9937	1	1
	AÖ - RO	1	1	1	**1**	1	1
	YÖ - C4.5	0,9735	0,9985	1	0,9966	1	1
	YÖ - EYK	0,9820	0,9945	1	0,9967	1	1
	YÖ -NB	0,9905	0,9962	0,9990	**0,9977**	1	1
	YÖ - DVM	0,9706	1	1	0,9970	1	1
	YÖ - RO	0,9682	0,9985	1	0,9970	1	1
	ROSE – C4.5	0,9693	0,9941	1	0,9946	1	1
	ROSE - EYK	0,9780	1	1	0,9972	1	1
	ROSE -NB	0,9890	0,9945	1	0,9972	1	1
	ROSE - DVM	0,9890	0,9958	1	0,9972	1	1
	ROSE - RO	0,9881	1	1	**0,9977**	1	1
	SMOTE – C4.5	0,9505	0,9908	1	0,9903	1	1
	SMOTE - EYK	0,9739	0,9960	1	**0,9963**	1	1
	SMOTE -NB	0,9479	0,9869	1	0,9911	1	1
	SMOTE - DVM	0,9635	1	1	0,9953	1	1
	SMOTE - RF	0,9505	0,9947	1	0,9934	1	1

Veri setinin %70-%30 bölünmesinde, eğitim için en iyi performanslar incelendiğinde, EAA orijinal veri seti için DVM'de %97,29; AÖ-NB ve AÖ- RO'da %100; YÖ-NB'de %99,77; ROSE- RO'da %99,77 ve SMOTE- EYK'de %99,63 elde edildi. Burada AÖ'nün yaklaşık olarak en iyi performansı gösterdiği saptandı.

Veri seti %60- %40 bölündüğünde hedef değişkenimizde 198 (%85,30) infertil ve 34 (%14,60) fertil erkek bulunmaktadır. Aşağıdaki Tablo 11'de dengesiz veri için önerilen farklı örnekleme sonuçları bulunmaktadır.

Tablo 11. İnfertilite verisi %60 - %40 bölündüğünde sınıf dengesizliği gözlem sayıları

	İnfertil	**Fertil**
Aşağı örnekleme	34	34
Yukarı örnekleme	198	198
SMOTE	136	102
ROSE	118	114

Buna göre elde edilen eğitim seti analiz sonuçları Tablo 12'de verildi.

Tablo 12. İnfertilite verisi %60 - %40 bölündüğünde sınıf dengesizliği eğitim performansı için EAA sonuçları **Min:** Minimum değer, **AO:** Aritmetik ortalama, **Mak:** Maksimum değer, **Ç1:** I. Çeyrek, **Ç3:** III. Çeyrek

		Min.	Ç1	Ortanca	AO	Ç3	Mak.
%60- %40	**Orijinal - C4.5**	0,8333	0,9656	0,9916	0,9648	1	1
	Orijinal - EYK	0,7916	0,9826	0,9875	0,9644	1	1
	Orijinal - NB	0,7368	0,9750	1	**0,9653**	1	1
	Orijinal - DVM	0,8245	0,9520	0,9645	0,9599	1	1
	Orijinal - RO	0,7937	0,9541	0,9934	0,9647	1	1
	AÖ - C4.5	0,8333	1	1	0,9833	1	1
	AÖ - EYK	0,8333	1	1	0,9833	1	1
	AÖ -NB	0,8750	1	1	**0,9875**	1	1
	AÖ - DVM	0,8333	1	1	0,9833	1	1
	AÖ - RO	0,8333	1	1	0,9833	1	1
	YÖ - C4,5	0,9868	0,9981	1	**0,9981**	1	1
	YÖ - EYK	0,9775	1	1	0,9974	1	1
	YÖ -NB	0,9850	0,9981	1	0,9979	1	1
	YÖ - DVM	0,9875	0,9981	1	0,9979	1	1
	YÖ - RO	0,9712	1	1	0,9968	1	1
	ROSE– C4.5	0,9375	0,9851	0,9982	0,9874	1	1
	ROSE - EYK	0,9583	0,9851	0,9900	**0,9886**	1	1
	ROSE -NB	0,9507	09696	0,9762	0,9807	1	1
	ROSE - DVM	0,9583	0,9734	0,9892	0,9851	0,9982	1
	ROSE - RO	0,9375	0,9788	0,9848	0,9811	0,9982	1
	SMOTE – C4.5	0,9076	1	1	0,9892	1	1
	SMOTE - EYK	0,9535	1	1	**0,9932**	1	1
	SMOTE -NB	0,8951	1	1	0,9881	1	1
	SMOTE- DVM	0,9500	0,9892	1	0,9897	1	1
	SMOTE - RO	0,9307	0,9946	1	0,9904	1	1

Veri setinin %60-%40 bölünmesinde, eğitim için en iyi performanslar incelendiğinde, EAA orijinal veri için NB'de %96,53; AÖ - NB'de %98,75; YÖ- C4.5'de %99,81; ROSE- EYK'de %98,86 ve SMOTE- EYK'de %99,32 şeklinde elde edildi. Burada YÖ'nün genel olarak en iyi performansı gösterdiği görüldü.

Test performansına ait sonuçlar %80- %20 olarak ayrıldığında EAA için güven aralığı alt ve üst değeri Tablo 13'de verildi.

Tablo 13. İnfertilite verisi %80- %20 bölündüğünde test performansı için EAA sonuçları

	Alt	EAA	Üst
Orijinal- C4.5	0,7557	0,9125	1,000
Orijinal - EYK	0,7619	0,9146	1,000
Orijinal - NB	0,8049	**0,9321**	1,000
Orijinal - DVM	0,7619	0,9146	1,000
Orijinal - RO	0,8077	0,9293	1,000
AÖ - C4.5	0,7810	0,9111	1,000
AÖ - EYK	0,7919	0,9153	1,000
AÖ -NB	0,7518	0,9013	1,000
AÖ - DVM	0,7951	0,9160	1,000
AÖ - RO	0,8063	**0,9188**	1,000
YÖ - C4.5	0,8157	0,9349	1,000
YÖ - EYK	0,8242	0,9377	1,000
YÖ -NB	0,8151	0,9328	1,000
YÖ - DVM	0,8338	**0,9419**	1,000
YÖ - RO	0,7733	0,9195	1,000
ROSE – C4.5	0,7828	**0,9237**	1,000
ROSE - EYK	0,7830	0,9209	1,000
ROSE -NB	0,7254	0,9027	1,000
ROSE - DVM	0,7400	0,9062	1,000
ROSE - RO	0,7700	0,9174	1,000
SMOTE– C4.5	0,7804	0,9209	1,000
SMOTE - EYK	0,7900	**0,9251**	1,000
SMOTE -NB	0,7887	0,9237	1,000
SMOTE - DVM	0,7796	0,9216	1,000
SMOTE- RO	0,7873	0,9237	1,000

Test için en iyi performanslar incelendiğinde veri seti %80-%20 olarak bölündüğünde EAA orijinal veri seti için NB'de %93,21; AÖ- RO'da %91,88; YÖ- DVM'de %94,19; ROSE- C4.5'de %92,37 ve SMOTE- EYK'de %92,51 olarak elde edildi. Burada en iyi yöntemin YÖ olduğu görüldü. Sonuç olarak eğitim ve test performansları birlikte değerlendirildiğinde EAA için YÖ-DVM'de en iyi performans yaklaşık %94 olarak saptandı.

Test performansına ait sonuçlar %70- %30 olarak ayrıldığında EAA için güven aralığı alt ve üst değeri Tablo 14'de verildi.

Tablo 14. İnfertilite verisi %70- %30 bölündüğünde test performansı için EAA sonuçları

	Alt	EAA	Üst
Orijinal - C4.5	0,8436	**0,9346**	1,000
Orijinal - EYK	0,8157	0,9250	1,000
Orijinal - NB	0,8389	0,9330	1,000
Orijinal - DVM	0,8416	0,9333	1,000
Orijinal - RO	0,8154	0,9247	1,000
AÖ - C4.5	0,8463	0,9381	1,000
AÖ - EYK	0,8647	0,9336	1,000
AÖ -NB	0,8554	0,9292	1,000
AÖ - DVM	0,8639	0,9330	1,000
AÖ - RO	0,8724	**0,9384**	1,000
YÖ - C4.5	0,8511	0,9368	1,000
YÖ - EYK	0,8514	0,9378	1,000
YÖ -NB	0,8529	0,9375	1,000
YÖ - DVM	0,8532	0,9371	1,000
YÖ - RO	0,8536	**0,9381**	1,000
ROSE – C4.5	0,8582	0,9323	1,000
ROSE - EYK	0,8629	0,9339	1,000
ROSE -NB	0,8711	0,9381	1,000
ROSE - DVM	0,8638	0,9359	1,000
ROSE - RO	0,8722	**0,9390**	1,000
SMOTE – C4.5	0,8527	0,9362	1,000
SMOTE - EYK	0,8680	0,9400	1,000
SMOTE -NB	0,8698	**0,9432**	1,000
SMOTE - DVM	0,8701	0,9403	1,000
SMOTE - RO	0,8581	0,9381	1,000

Test için en iyi performanslar incelendiğinde veri seti %70-%30 olarak bölündüğünde EAA orijinal veri seti için C4.5'de %93,46; AÖ - RO'da %93,84; YÖ- RO'da %93,81; ROSE - RO'da %93,90 ve SMOTE - NB'de % 94,32 olarak elde edildi. Burada en iyi yöntemin SMOTE olduğu görüldü. Sonuç olarak eğitim ve test performansları birlikte değerlendirildiğinde EAA için ROSE-RO'da en iyi performans yaklaşık %94 olarak saptandı.

Test performansına ait sonuçlar veri %60- %40 olarak bölündüğünde EAA için güven aralığı alt ve üst değeri Tablo 15'de verildi.

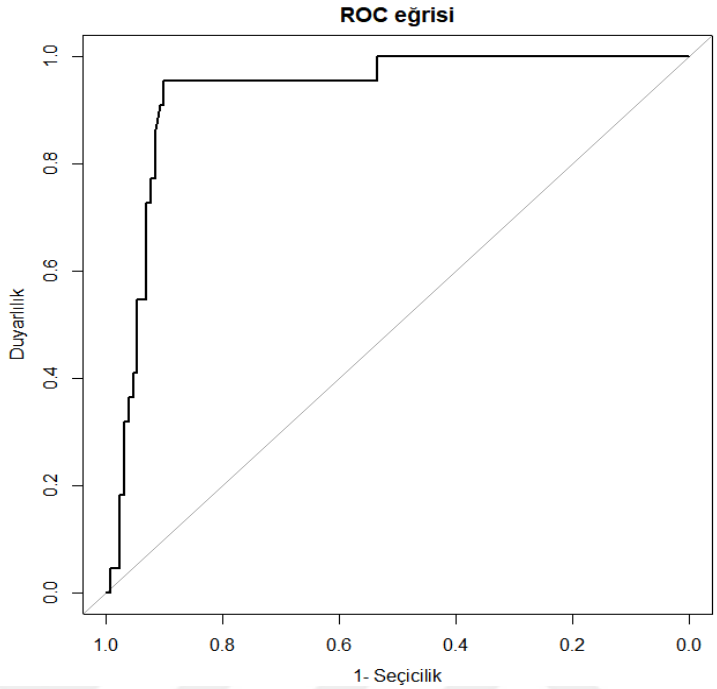
Tablo 15. İnfertilite verisi %60- %40 bölündüğünde test performansı için EAA sonuçları

	Alt	EAA	Üst
Orijinal - C4.5	0,8797	0,9470	1,000
Orijinal - EYK	0,8822	0,9479	1,000
Orijinal - NB	0,8845	0,9482	1,000
Orijinal - DVM	0,8881	**0,9507**	1,000
Orijinal - RO	0,8805	0,9479	1,000
AÖ - C4.5	0,8837	0,9463	1,000
AÖ - EYK	0,8571	0,9371	1,000
AÖ -NB	0,8923	**0,9495**	1,000
AÖ - DVM	0,8735	0,9439	1,000
AÖ - RO	0,8625	0,9392	1,000
YÖ - C4.5	0,8820	0,9453	1,000
YÖ - EYK	0,8812	0,9437	1,000
YÖ -NB	0,8825	0,9441	1,000
YÖ - DVM	0,8823	0,9444	1,000
YÖ - RO	0,8864	**0,9463**	1,000
ROSE – C4.5	0,8515	0,9292	1,000
ROSE- EYK	0,8592	0,9330	1,000
ROSE -NB	0,8542	0,9297	1,000
ROSE- DVM	0,8291	0,9188	1,000
ROSE - RO	0,8835	**0,9411**	0,9988
SMOTE– C4.5	0,8962	0,9547	1,000
SMOTE - EYK	0,9026	**0,9580**	1,000
SMOTE -NB	0,9005	0,9573	1,000
SMOTE - DVM	0,8997	0,9557	1,000
SMOTE - RO	0,9049	**0,9580**	1,000

Test için en iyi performanslar incelendiğinde veri seti %60-%40 olarak bölündüğünde EAA, orijinal veri seti için DVM'de %95,07; AÖ - NB'de %94,95; YÖ- RO'da %94,63; ROSE - RO'da %94,11; SMOTE -EYK ve RO'da %95,80 şeklinde elde edildi. Burada en iyi yöntemin SMOTE olduğu saptandı. Sonuç olarak eğitim ve test performansları birlikte değerlendirildiğinde EAA için SMOTE-EYK'de en iyi performans yaklaşık %96 olarak saptandı. En iyi performanslara ait ROC eğrileri Şekil 17'de verildi.

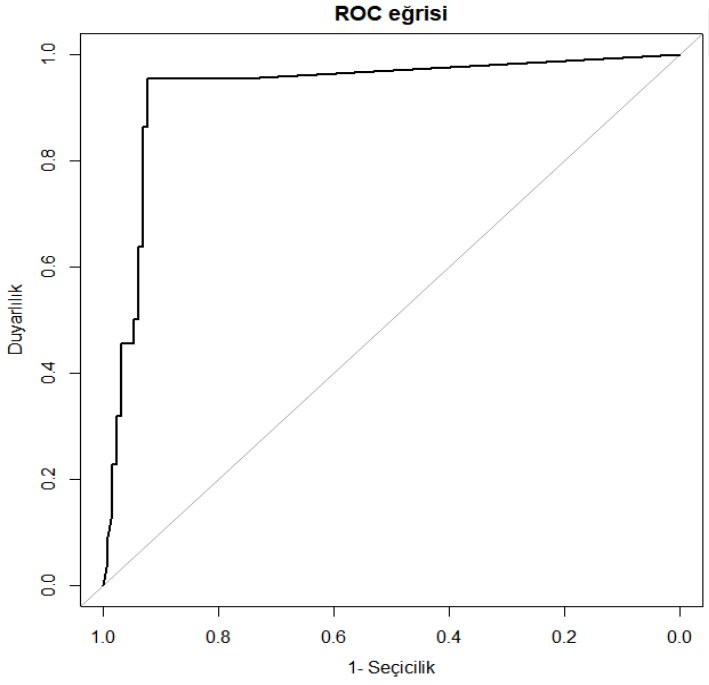
(a)

EAA: 0.9932



(b)

EAA: 0.9580



Şekil 17. (a) Orijinal veri seti ile EYK algoritması ile elde edilen ROC eğrisi 17 (b) SMOTE ile elde edilen EYK ve RO algoritmasına ait ROC eğrisi

Bir diğer algoritma SL için analiz yapılmadan önce eksik gözlemler analizden çıkarıldı. Veri özellikleri dikkate alınarak faktör değişkenler belirlendi. Buna göre kütüphaneden RO, EYK, bayes genelleştirilmiş lineer modeller (bayesglm) ve karar ağaçları (rpart) seçilerek analiz yapıldı. Aşağıdaki Tablo 16'da elde edilen risk ve katsayı sonuçları verildi.

Tablo 16. İnfertilite verisine ait super learner ile elde edilen risk ve katsayı sonuçları

	Risk	**Katsayı**
SL.randomForest_All	**0,0589**	**0,5810**
SL.knn_All	0,0625	0,2327
SL.bayesglm_All	0,0788	0,0000
SL.rpart_All	0,0613	0,1862

Tabloda modelde en fazla ağırlığın RO algoritmasına verildiği görülmekte olup, minimum risk 0,0589 ve ağırlık 0,5810 şeklindeydi. Bunu takip eden değerler, rpart ve EYK algoritması ile elde edilmiştir. Kurulacak olan modelde en fazla ağırlık RO algoritmasına verilecek olup bunu rpart ve EYK takip etmektedir. Bayes algoritması modele katkı sağlamayacaktır.

Tablo 17. İnfertilite verisine ait çapraz geçerleme ile super learner için EAA analiz sonuçları

Algoritmalar	**Minimum**	**Ortalama EAA**	**Maksimum**
Super Learner (SL)	0,8888	0,9653	1
Discrete SL	0,8333	0,9597	1
SL.randomForest_All	0,8333	0,9597	1
SL.knn_All	0,7222	0,9321	1
SL.bayesglm_All	0,8461	0,9465	1
SL.rpart_All	0,7222	0,9304	1

Tablo 17'de ÇG sonuçları yer almaktadır. ÇG kullanarak elde edilen modelde en yüksek başarı SL ile elde edildi, EAA %96,53 idi. Bunu sırası ile Discrete SL ve RO'da %95,97 EAA değeri, bayesglm'de %94,65 ve EYK ile rpart'da sırasıyla %93,21 ve %93,04 EAA değeri takip etti.

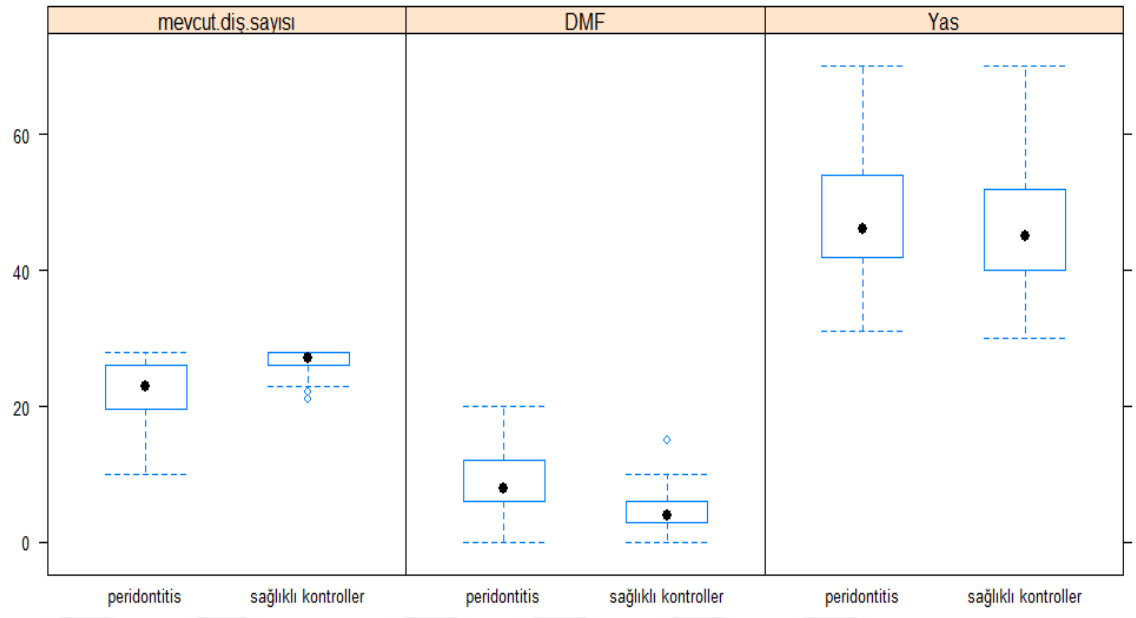
4.2. Peridontitis Verisi için

Peridontitis veri setinde, başlangıçta 72 peridontitis hastası ve 102 sağlıklı kontrol olmak üzere toplam 174 gözlem ile üç sayısal, yedi kategorik ve bir hedef değişken (peridontitis ve sağlıklı kontrol) bulunmakta idi. Sıfıra yakın varyans bulunması sebebi ile çalışmadan Taq2 değişkeni ve eksik olan gözlemler çıkarılarak çalışma 71 (%44,94) peridontitis hastası ve 87 (%55,06) sağlıklı kontroller üzerinde, toplam dokuz değişken için yapıldı. Değişkenler için özet istatistikler Tablo 18'de verildi.

Tablo 18. Peridontitis verisi için kategorik ve sayısal değişkenlere ait özet istatistikler **Min:** Minimum değer, **AO:** Aritmetik ortalama, **Mak:** Maksimum değer, **Ç1:** I. Çeyrek, **Ç3:** III. Çeyrek

Polimorfizmi	**Sonuç**					
	Peridontitis için Allel Sayısı			**Sağlıklı Kontroller için Allel Sayısı**		
	n (%)			**n (%)**		
Bsm1	83(52,53)			75(47,47)		
Bsm2	92(58,23)			66(41,77)		
Apa1	98(62,03)			60(37,97)		
Apa2	126(79,75)			32(20,25)		
Taq1	90(56,96)			68(43,04)		
	Erkek			**Kadın**		
Cinsiyet	85(53,80)			73(46,20)		
	Min.	**Ç1**	**Ortanca**	**AO**	**Ç3**	**Mak.**
Mevcut Diş Sayısı	10,00	23,00	26,00	24,61	27,75	28,00
Çürük Diş İndeksi (DMF)	0,00	3,00	6,00	6,46	8,75	20,00
Yaş	30,00	40,25	45,50	47,01	52,00	70,00

Sayısal değişkenlere ait grafikler Şekil 18' de verildi.



Şekil 18. Peridontitis verisine ait sayısal değişkenlere ilişkin kutu- çizgi grafikleri

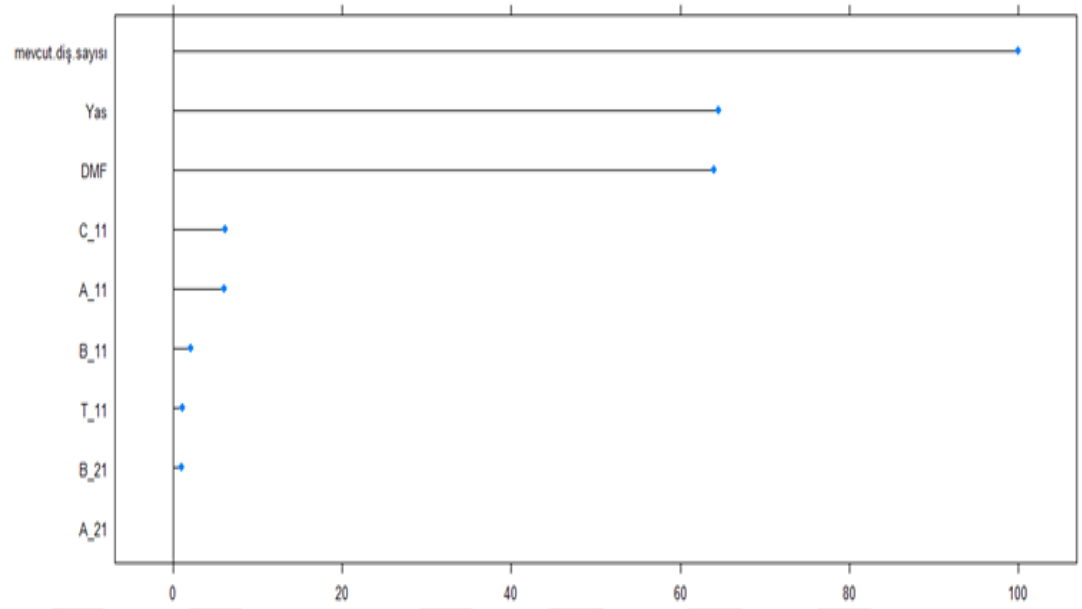
Sayısal değişkenler için ön-işleme adımı yapıldıktan sonra eğitim setleri sırasıyla %80, %70 ve %60; test setleri de sırasıyla %20, %30 ve %40 olarak ayrıldı. Analiz öncesi aynı sonuçları elde etmek amacıyla her seferinde set.seed (1234) komutu kullanıldı. Makine öğrenimi algoritmaları C4.5, EYK, NB, DVM ve RO ile elde edilen sonuçlar Tablo 19' da verildi.

Tablo 19. Peridontitis verisine ait algoritma sonuçları

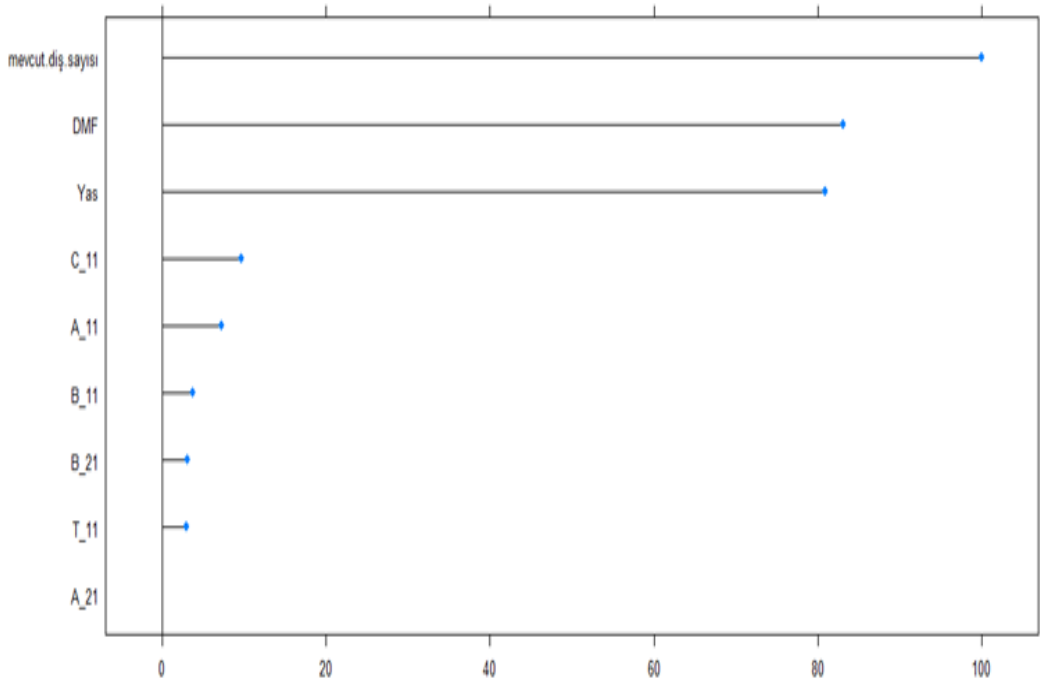
	Performans Ölçüleri	C4.5	EYK	NB	DVM	RO
%80 -%20	**Doğruluk Oranı**	0,6774	0,6129	**0,7419**	0,7097	0,7097
	Duyarlılık	0,5714	0,3571	0,5714	0,5714	0,6429
	Seçicilik	0,7647	0,8235	0,8824	0,8235	0,7647
	Pozitif Kestirim Değeri	0,6667	0,6250	0,8000	0,7273	0,6923
	Negatif Kestirim Değeri	0,6842	0,6087	0,7143	0,7000	0,7222
	EAA	0,6890	0,6932	0,7310	0,7016	**0,8172**
%70-%30	**Doğruluk Oranı**	0,7021	0,7021	**0,7234**	0,7021	0,6809
	Duyarlılık	0,5238	0,3333	0,4762	0,5238	0,5238
	Seçicilik	0,8462	1,0000	0,9231	0,8462	0,8077
	Pozitif Kestirim Değeri	0,7333	1,0000	0,8333	0,7333	0,6875
	Negatif Kestirim Değeri	0,6875	0,6500	0,6857	0,6875	0,6774
	EAA	0,6785	0,6602	0,7673	0,7472	**0,8278**
%60- %40	**Doğruluk Oranı**	0,7419	0,6774	0,7419	**0,8065**	0,7742
	Duyarlılık	0,5714	0,3571	0,4643	0,6071	0,5714
	Seçicilik	0,8824	0,9412	0,9706	0,9706	0,9412
	Pozitif Kestirim Değeri	0,8000	0,8333	0,9286	0,9444	0,8889
	Negatif Kestirim Değeri	0,7143	0,6400	0,6875	0,7500	0,7273
	EAA	0,7731	0,6827	0,8182	0,7636	**0,8466**

Veri seti %80 - %20 olarak ayrıldığında doğruluk oranı, C4.5'de %67,74; EYK'de %61,29 ve DVM ile RO'da %70,97 olarak elde edildi. En iyi performans gösteren algoritma olan NB'de doğruluk oranı %74,19 şeklindeydi. Veri seti %70- %30 olarak ayrıldığında doğruluk oranı, C4.5, EYK ve DVM'de %70,21; RO'da %68,09 şeklinde iken, en iyi performans gösteren algoritma NB'de %72,34 olarak elde edildi. Veri seti %60- %40 olarak ayrıldığında doğruluk oranı, NB ve C4.5'de %74,19; EYK'de %67,74 ve RO'da %77,42 şeklinde saptandı. En iyi performans gösteren algoritma DVM'de doğruluk oranı %80,65'di.

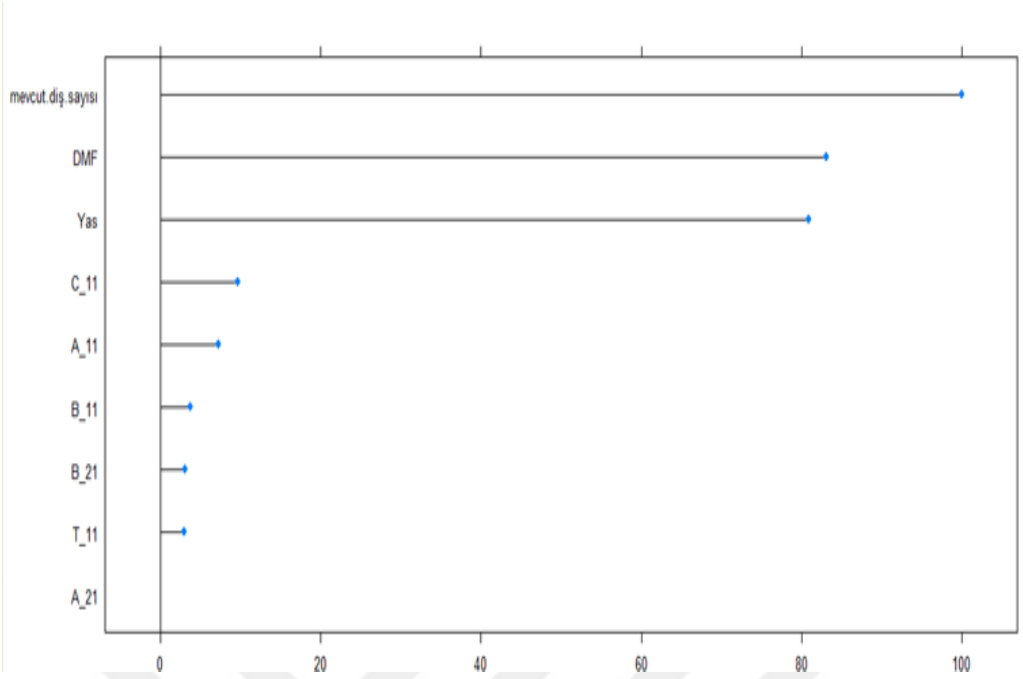
En iyi performans gösteren algoritmalara ait farklı eğitim ve test bölmelerine göre değişkenlerin önemliliği aşağıdaki şekillerde (Şekil 19-21) verildi.



Şekil 19. Veri seti %80-%20 bölündüğünde RO için değişkenlerin önemlilik göstergeleri



Şekil 20. Veri seti %70-%30 bölündüğünde RO için değişkenlerin önemlilik göstergeleri



Şekil 21. Veri seti %60-%40 bölündüğünde RO için değişkenlerin önemlilik göstergeleri

Farklı eğitim ve test setlerine göre önemlilik incelendiğinde, üç farklı bölme oranı için ilk üç sıradaki değişkenler, mevcut diş sayısı, yaş ve çürük diş indeksidir. Polimorfizmleri incelendiğinde en etkili Taq1 polimorfizmi olup, bunu Apa1 ve Bsm1 polimorfizmleri takip etmektedir.

SL analizi için analiz yapılmadan önce eksik gözlemler çıkarıldı. Veri özellikleri dikkate alınarak faktör değişkenler belirlendi. Kütüphaneden RO, EYK, bayesglm, DVM ve karar ağaçları seçilerek analiz yapıldı. Aşağıdaki Tablo 20'de elde edilen risk ve katsayı sonuçları verildi.

Tablo 20. Peridontitis verisine ait super learner ile elde edilen risk ve katsayı sonuçları

	Risk	**Katsayı**
SL.randomForest_All	**0,1757**	**0,2220**
SL.knn_All	0,1959	0,0000
SL.bayesglm_All	0,1870	0,0339
SL.svm_All	0,1888	0,2742
SL.rpart_All	0,1772	0,4698

Tabloda modele en fazla ağırlığın RO algoritmasına verildiği görülmekte olup, minimum risk 0,1757 ve ağırlık 0,2220 şeklindeydi. Bunu takip eden değerler, rpart ve bayesglm algoritmasıdır. Kurulacak olan modelde en fazla ağırlık RO algoritmasına verilecek olup EYK algoritması modele katkı sağlamayacaktır.

Tablo 21. Peridontitis verisine ait ÇG ile super learner için ortalama EAA analiz sonuçları

Algoritmalar	Minimum	Ortalama Değer	Maksimum
Super Learner (SL)	0,5416	0,8483	1,0000
Discrete SL	0,5200	0,7687	1,0000
SL.randomForest_All	0,5200	0,8124	1,0000
SL.knn_All	0,5714	0,8027	1,0000
SL.bayesglm_All	0,5833	0,8148	0,9583
SL.svm_All	0,5238	0,7688	0,9375
SL.rpart_All	0,5416	0,8142	0,9800

Tablo 21'de ÇG sonuçları yer almaktadır. ÇG kullanarak elde edilen modelde en yüksek başarı SL ile elde edilmiş olup EAA değeri %84,83 şeklindeydi. EAA değeri, RO'da % 81,24; bayesglm'de % 81,48 ve rpart'da % 81,42; EYK'de %80,27; discrete SL'de % 76,87 ve DVM'de %76,88 şeklinde idi.

4.3. İlk Benzetim Verisi

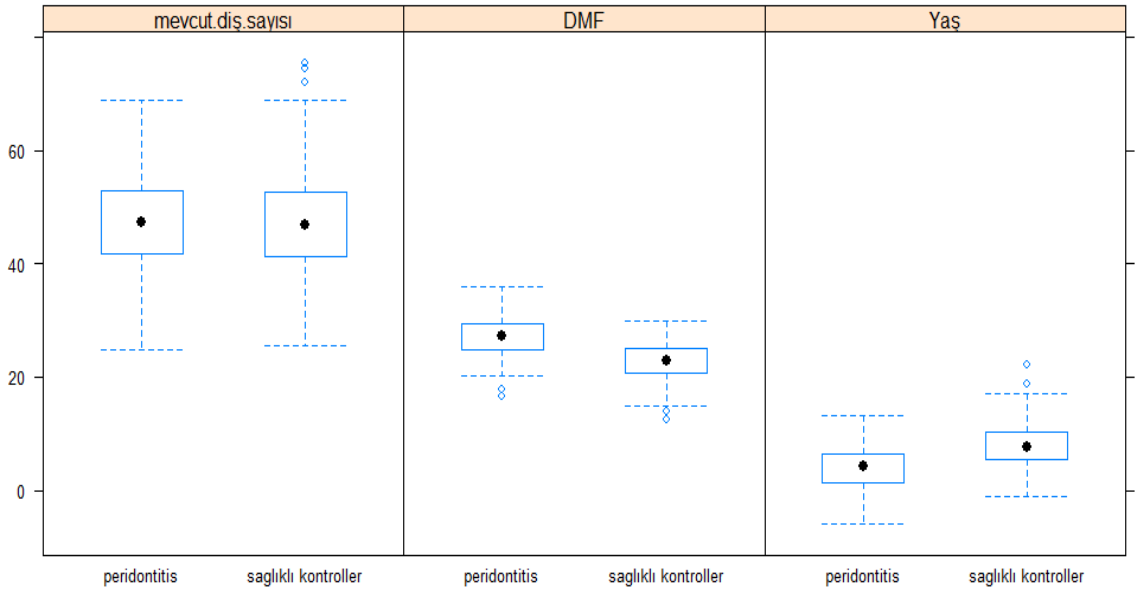
İlk benzetim verisi peridontitis verisinden n= 500 için oluşturulmuştur. 199 (%39,80) peridontitis hastası ve 301 (%60,20) sağlıklı birey bulunmaktadır. Verideki değişkenler için uygun dönüşümler, ardından ön-işleme gerçekleştirildi. Değişkenler için özet istatistikler Tablo 22'de verildi.

Tablo 22. İlk Benzetim verisine ait kategorik ve sayısal değişkenlere ait özet istatistikler **Min:** Minimum değer, **AO:** Aritmetik ortalama, **Mak:** Maksimum değer, **Ç1:** I. Çeyrek, **Ç3:** III. Çeyrek

Polimorfizmi	**Sonuç**	
	Peridontitis için Allel Sayısı	**Sağlıklı Kontroller için Allel Sayısı**
	n (%)	**n (%)**
Bsm1	264(52,80)	236 (47,20)
Bsm2	254(50,80)	246(49,20)
Apa1	300(60)	200 (40)
Apa2	315(63)	185(37)
Taq1	403(80,60)	97(19,40)
	Erkek	**Kadın**
Cinsiyet	260(52)	240(48)

	Min.	**Ç1**	**Ortanca**	**AO**	**Ç3**	**Mak.**
Mevcut Diş Sayısı	12,59	21,75	24,47	24,50	27,23	36,07
Çürük Diş İndeksi (DMF)	0,00	3,82	6,46	6,41	9,19	22,06
Yaş	24,88	41,51	47,24	47,05	52,70	75,30

Sayısal değişkenlerin dağılımları Şekil 22'de verildi.



Şekil 22**.** İlk benzetim verisindeki sayısal değişkenlere ait kutu- çizgi grafiği

İlk benzetim verisine ait analiz sonuçları Tablo 23' de verildi.

Tablo 23**.** İlk benzetim verisine ait algoritma sonuçları

	Performans Ölçüleri	**C4.5**	**EYK**	**NB**	**DVM**	**RO**
%80 -%20	**Doğruluk Oranı**	0,6869	0,6970	0,6970	**0,7273**	0,7172
	Duyarlılık	0,4103	0,3846	0,4359	0,4359	0,4359
	Seçicilik	0,8667	0,9000	0,8667	0,9167	0,9000
	Pozitif Kestirim Değeri	0,6667	0,7143	0,6800	0,7727	0,7391
	Negatif Kestirim Değeri	0,6933	0,6923	0,7027	0,7143	0,7105
	EAA	0,7138	0,6884	**0,7222**	0,7029	0,7021
%70-%30	**Doğruluk Oranı**	0,6376	0,6577	**0,6779**	**0,6779**	**0,6846**
	Duyarlılık	0,5424	0,4407	0,5593	0,5085	0,4746
	Seçicilik	0,7000	0,8000	0,7556	0,7889	0,8222
	Pozitif Kestirim Değeri	0,5424	0,5909	0,6000	0,6122	0,6364
	Negatif Kestirim Değeri	0,7000	0,6857	0,7234	0,7100	0,7048
	EAA	0,6038	0,7136	**0,7419**	**0,7397**	0,7145
%60- %40	**Doğruluk Oranı**	0,7186	**0,7286**	0,6884	**0,7286**	0,6834
	Duyarlılık	0,5316	0,5190	0,5696	0,5823	0,4557
	Seçicilik	0,8417	0,8667	0,7667	0,8250	0,8333
	Pozitif Kestirim Değeri	0,6885	0,7193	0,6164	0,6866	0,6429
	Negatif Kestirim Değeri	0,7319	0,7324	0,7302	0,7500	0,6993
	EAA	0,7196	0,7331	**0,7821**	**0,7791**	0,7436

Veri seti %80 - %20 olarak ayrıldığında doğruluk oranı, C4.5'de %68,69; EYK ve NB'de %69,70 ve RO'da %71,72 olarak elde edildi. En iyi performans gösteren algoritma olan DVM'de doğruluk oranı %72,73 şeklindeydi. Veri seti %70- %30 olarak ayrıldığında doğruluk oranı, C4.5'de %63,76; EYK'de %65,77 şeklinde iken; en iyi performans gösteren algoritmalar NB'de, DVM'de %67,79 ve RO'da %68,46 elde edildi. Veri seti %60- %40 olarak ayrıldığında doğruluk oranı, C4.5'de %71,86, NB'de %68,84 ve RO'da %68,34 şeklinde saptandı. En iyi performans gösteren algoritma DVM ve EYK'de %72,86'di.

Super learner analizi için faktör değişkenler belirlendi. Kütüphaneden RO, EYK, bayesglm, DVM ve karar ağaçları (rpart) seçilerek analiz yapıldı. Tablo 24'de elde edilen risk ve katsayı analiz sonuçları verildi.

Tablo 24. İlk benzetim verisine ait super learner risk ve katsayı sonuçları

	Risk	**Katsayı**
SL.randomForest_All	0,1754	0,0000
SL.knn_All	0,1974	0,0000
SL.bayesglm_All	**0,1708**	**0,5472**
SL.svm_ALL	0,1875	0,0000
SL.rpart_All	0,1755	0,4527

Modelde en fazla ağırlığın bayesglm algoritmasına verildiği görülmekte olup, minimum risk 0,1708 ve ağırlık 0,5472 şeklindeydi. Bunu takip eden rpart'dır.

Tablo 25**.** İlk benzetim verisine ait ÇG super learner EAA analiz sonuçları

Algoritmalar	**Minimum**	**Ortalama EAA**	**Maksimum**
Super Learner (SL)	0,6666	0,8099	0,9047
Discrete SL	0,6666	0,8071	0,8928
SL.bayesglm_All	0,6666	0,8071	0,8929
SL.rpart_All	0,5297	0,7524	0,8809

Tablo 25'de ÇG sonuçları yer almaktadır. ÇG kullanarak elde edilen modelde en yüksek başarı SL ile elde edilmiş olup EAA değeri %80,99 şeklindeydi. EAA değeri, Discrete SL ve bayesglm %80,71 elde edilirken Rpart'da %75,24 şeklindeydi.

4.4. İkinci Benzetim Verisi

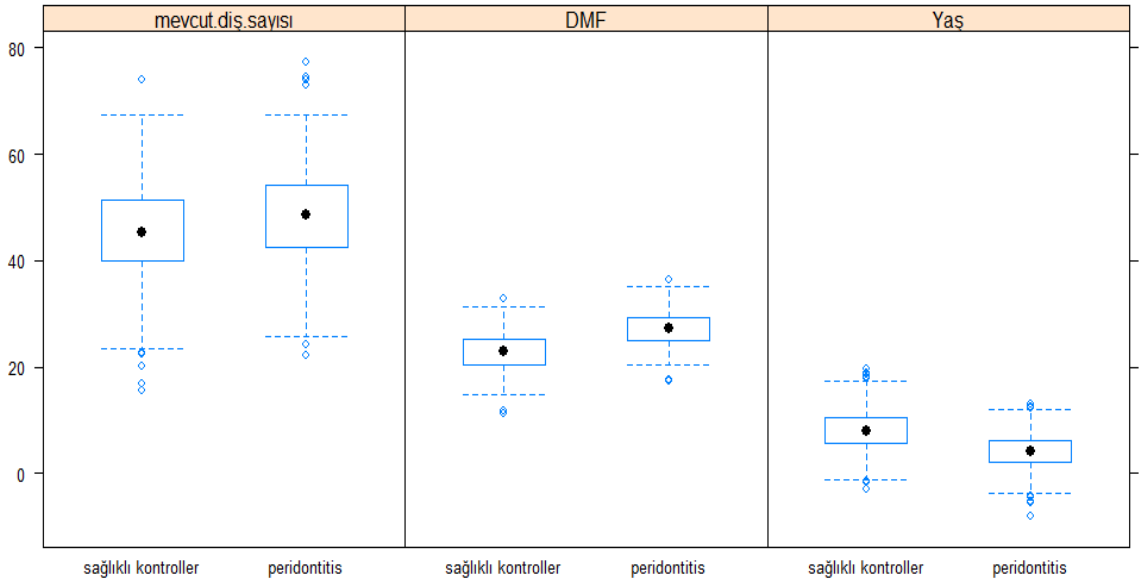
İkinci benzetim verisi periodontitis verisinden n=1000 için oluşturulmuştur. 389 (%38,90) peridontitis hastası ve 611 (%61,10) sağlıklı birey bulunmaktadır. Ön-işleme adımına göre sayısal veriler normalleştirilme yapıldı. Değişkenlerin özet istatistikleri Tablo 26'da verildi.

Tablo 26. İkinci benzetim verisine ait ikili ve sayısal değişkenlere ait özet istatistikler **Min:** Minimum değer, **AO:** Aritmetik ortalama, **Mak:** Maksimum değer, **Ç1:** I. Çeyrek, **Ç3:** III. Çeyrek

Polimorfizmi	**Sonuç**	
	Peridontitis için Allel Sayısı	**Sağlıklı Kontroller için Allel Sayısı**
	n (%)	**n (%)**
Bsm1	510 (51,00)	490 (49,00)
Bsm2	506 (50,60)	494 (49,40)
Apa1	580 (58,00)	420 (42,00)
Apa2	618 (61,80)	382 (38,20)
Taq1	781 (78,10)	219 (21,90)
	Erkek	**Kadın**
Cinsiyet	561 (56,10)	439 (43,90)

	Min.	**Ç1**	**Ortanca**	**AO**	**Ç3**	**Mak.**
Mevcut Diş Sayısı	11,23	21,82	24,54	24,50	27,09	36,44
Çürük Diş İndeksi (DMF)	0,00	3,83	6,49	6,43	8,98	19,65
Yaş	15,57	40,75	46,47	46,50	52,33	77,25

Sayısal değişkenlerin grafikleri Şekil 23'de verildi.



Şekil 23. İkinci benzetim verisindeki sayısal niteliklere ait kutu- çizgi grafiği

Eğitim ve test setlerinin farklı bölünme durumlarına ilişkin analiz sonuçları Tablo 27'de verildi.

Tablo 27. İkinci benzetim verisine ait algoritma sonuçları

	Performans Ölçüleri	C4.5	EYK	NB	DVM	RO
%80 -%20	**Doğruluk Oranı**	0,7337	0,7085	**0,7588**	**0,7638**	0,7487
	Duyarlılık	0,6234	0,5195	0,7403	0,6753	0,6234
	Seçicilik	0,8033	0,8279	0,7705	0,8179	0,8279
	Pozitif Kestirim Değeri	0,6667	0,6557	0,6706	0,7027	0,6957
	Negatif Kestirim Değeri	0,7717	0,7319	0,8246	0,8000	0,7769
	EAA	0,7962	0,7769	**0,8441**	0,8236	0,8217
%70-%30	**Doğruluk Oranı**	0,7492	0,7324	0,7559	**0,7726**	**0,7692**
	Duyarlılık	0,7069	0,5345	0,6897	0,6379	0,6552
	Seçicilik	0,7760	0,8579	0,7978	0,8579	0,8415
	Pozitif Kestirim Değeri	0,6667	0,7045	0,6838	0,7400	0,7238
	Negatif Kestirim Değeri	0,8068	0,7441	0,8022	0,7889	0,7938
	EAA	0,7801	0,8062	**0,8459**	0,8387	0,8363
%60- %40	**Doğruluk Oranı**	0,7544	0,7143	0,7669	**0,7769**	0,7619
	Duyarlılık	0,6323	0,5226	0,7226	0,6710	0,6387
	Seçicilik	0,8320	0,8361	0,7951	0,8443	0,8402
	Pozitif Kestirim Değeri	0,7050	0,6694	0,6914	0,7324	0,7174
	Negatif Kestirim Değeri	0,7808	0,7338	0,8186	0,8016	0,7854
	EAA	0,7889	0,7659	**0,8450**	0,8301	0,8223

Veri seti %80 - %20 olarak ayrıldığında doğruluk oranı, C4.5'de %73,37; EYK'de %70,85 ve RO'da %74,87 olarak elde edildi. En iyi performans gösteren algoritmalar olan DVM doğruluk oranı %76,38 ve NB'de %75,88 şeklindeydi. Veri seti %70- %30 olarak ayrıldığında doğruluk oranı, C4.5'de %74,92; EYK'de %73,24 ve NB'de %75,59 iken en iyi performans gösteren algoritmalar DVM'de %77,26 ve RO'de %76,92 olarak elde edildi. Veri seti %60- %40 olarak ayrıldığında doğruluk oranı, C4.5'de %75,44; EYK'de %71,43; NB'de %76,69 ve RO'da %76,19 şeklinde saptandı. En iyi performans gösteren algoritma DVM'de doğruluk oranı %77,69'di.

SL analizi için faktör değişkenler belirlendi. Kütüphaneden RO, EYK, DVM, bayesglm ve karar ağaçları (rpart) seçilerek analiz yapıldı. Aşağıdaki Tablo 28'de elde edilen risk ve katsayı sonuçları verildi.

Tablo 28. İkinci benzetim verisine ait super learner ağırlıklandırılmış algoritma sonuçları

	Risk	**Katsayı**
SL.randomForest_All	0,1628	0,0000
SL.knn_All	0,1647	0,1804
SL.svm_All	0,1603	0,0000
SL.bayesglm_All	**0,1514**	**0,8169**
SL.rpart_All	0,1777	0,0025

Modele en fazla ağırlığın bayesglm algoritmasına verildiği görülmekte olup, minimum risk 0,1514 ve ağırlık 0,8169 şeklindeydi. Modelde RO ve DVM ağırlığı sıfırdır.

Tablo 29. İkinci benzetim verisi için ÇG super learner algoritmasına ait EAA analiz sonuçları

Algoritmalar	**Minimum**	**Ortalama**	**Maksimum**
Super Learner (SL)	0,7519	0,8590	0,9437
Discrete SL	0,7363	0,8640	0,9616
SL.knn_All	0,7358	0,8332	0,9113
SL.bayesglm_All	0,7363	0,8640	0,9616
SL.rpart_All	0,6457	0,7832	0,9206

Tablo 29'da ÇG sonuçları yer almaktadır. ÇG kullanarak elde edilen modelde en yüksek başarı discrete SL ve bayesglm ile elde edilmiş olup EAA değeri %86,40 şeklindeydi. EAA değeri, SL'de %85,90; EYK'de %83,32 ve rpart'da %78,32 şeklinde idi.

5. TARTIŞMA

Denetimli öğrenme yöntemleri bir dizi etiketlenmiş örneğe sahiptir ve daha önce görmediği noktalar için tahminlerde bulunur (Mohri ve ark., 2012). Bu yöntemler içinde sıklıkla kullanılan karar ağaçları, EYK, NB, DVM algoritmalarının yanı sıra topluluk yöntemleri içerisinde yer alan RO algoritması ile sonuçlar elde edildi. DVM algoritması için çekirdek fonksiyon olarak RTF kullanıldı. EYK algoritması hesaplanırken de uzaklık ölçüsü olarak öklit kullanıldı. Makine öğrenimi algoritmalarının analizinde karşılaşılan en büyük problemler arasında, hangi algoritmanın hangi veri seti için kullanılacağı ve kullanılan veri setinin hangi oranlarda eğitim-test verisi olarak ayrılması gerektiği şeklinde sıralanabilir. Çalışmada genetik bilgileri de içeren veri setleri için bu konular üzerinde çalışılarak, farklı büyüklükte veri setleri için performanslar saptandı. Ayrıca, bu yöntemler bir takım aday öğreticilerin uygulandığı ve topluluk yöntemleri içerisinde yer alan SL algoritması sonuçları ile karşılaştırıldı.

Çalışmada ilk olarak infertilite verisi üzerinde analizler yapıldı. Bu veri seti hedef değişken bakımından ön işleme yöntemleri ile değerlendirildiğinde bir sınıf çok sayıda örnek ile temsil edilirken, diğer sınıfta örnek sayısı daha azdı. Bu sebeple, dengesiz veri problemi için önerilen farklı yöntemler (AÖ, YÖ, ROSE ve SMOTE) kullanılarak analizler yapıldı.

Eğitim ve test setlerinin farklı bölünmelerine göre infertilite veri seti %80-%20 olarak ayrıldığında doğruluk oranı için en iyi sonuç RO'da elde edildi. Takip eden algoritmalar doğruluk oranı için DVM ile C4.5 algoritmalarıdır. Veri seti %70-%30 olarak ayrıldığında doğruluk oranı için en iyi sonuç DVM'de elde edildi. Bunları takip eden algoritmalar RO ve C4.5'dir. Veri seti %60-%40 olarak ayrıldığında doğruluk oranı için RO'da en iyi sonuç elde edildi. Performans kriteri doğruluk oranı olduğunda ikinci iyi performans C4.5'dir. Sonuçlara göre RO ve DVM algoritmaları ile en iyi performanslar elde edildi. Bu elde edilen sonuçlar literatürdeki çalışmaları desteklemektedir. Nitekim Noi ve Kappas (2017), çalışmalarında dengeli ve dengesiz veri setleri için farklı bölünme oranları kullanarak farklı veri büyüklükleri için analizler yapıldığında, %90-%95 doğruluk elde edilmiştir. En yüksek performans, %60 bölünen eğitim veri seti için DVM ile elde edilmiş ve bunu RO ile EYK takip etmiştir (Noi ve Kappas, 2017). Nitze ve ark. (2012), çalışmalarında ürün sınıflandırma için performans ölçüsü olarak doğruluk oranı ve DVM'nin özellikle RTF çekirdek fonksiyonu

kullanıldığında, RO ve YSA algoritmalarına göre daha yüksek performans gösterdiğini belirtilmiştir. Burada kullanılan görüntü sayısı arttıkça performansın arttığı gözlenmiştir (Nitze ve ark, 2012). İnfertilite veri seti de RTF çekirdek fonksiyonu kullanarak en yüksek performans DVM ve RO ile elde edildi. Farklı bölünme oranları farklı algoritmaların performansı üzerinde etki ettiği elde edildi.

Dengesiz veri problemini çözmek için literatürde önerilen yaklaşımlara göre farklı bölünmelere göre analizler infertilite veri seti için yapıldı. Eğitim ve test seti performansları birlikte değerlendirildiğinde veri seti %80- %20 olarak ayrıldığında EAA için en yüksek performans YÖ- DVM'de %94 elde edildi. Veri seti %70-%30 olarak ayrıldığında en yüksek EAA için performans ROSE-RO'da yaklaşık %94 ve veri seti %60-%40 olarak ayrıldığında EAA için en yüksek performans SMOTE-EYK'de ve RO'da yaklaşık %96 şeklinde saptandı. Hordri ve ark. (2018), kredi kartı dolandırıcılığı konusundaki çalışmalarında farklı bölünme oranları için performans ölçüsü olarak EAA kullanmışlardır. Kullanılan algoritmalar ise NB, doğrusal regresyon, RO ve çok katmanlı ağ yöntemleridir. Sonuçlara göre doğrusal regresyon, RO ve çok katmanlı ağ yöntemleri için SMOTE yönteminin %99 başarı sağladığı YÖ –RO'de tam başarı elde edilmiştir (Hordri ve ark., 2018). İnfertilite için farklı örnekleme yöntemleri kullanıldığında bölünme oranları dikkate alındığında birbirine yakın ve yüksek performans elde edildi. Bölünme %60-%40 olduğunda SMOTE yönteminin daha başarılı olduğu elde edildi.

İnfertilite verisi SL algoritması kullanarak analiz sonucunda en düşük risk değeri RO ile elde edildi. Kurulan SL modelinde en büyük katkıyı sırasıyla RO, EYK ve rpart sağladı. Analiz sonucunda SL'de EAA, yaklaşık %97 elde edildi. SL ile elde edilen performansın klasik yöntemler (DVM-%96) ile elde edilen sonuçtan daha iyi olduğu görüldü. Nitekim Van der Laan ve ark. (2007), diyabet hastalığına ilişkin veri setinde aday öğreticiler arasında RO, En küçük kareler yöntemi, En Küçük Açı Regresyonu [Least Angle Regression (LARS)] ve Silme/Değişiklik/Ekleme [(Deletion/Substitution/Addition (D/S/A)] yöntemleri kullanmıştır. En küçük risk D/S/A ve SL algoritmaları kullanarak elde etmişlerdir. SL algoritması farklı makine öğrenimi algoritmaları için ÇG kullanarak performansı arttırmaktadır (Van der Laan ve ark, 2007). İnfertilite veri seti için farklı aday öğreticiler kullanılarak daha iyi performans elde edildi. Ayrıca en küçük risk RO algoritması için elde edildi.

İnfertilite veri seti genetik varyasyonları, hormanlar ve semen parametreleri içermektedir. Veri seti %80-%20 olarak ayrıldığında en yüksek performans gösteren EAA RO'da %96'dı ve SL'de %97 elde edildi. Palechor ve ark. (2016), çalışmalarında karar ağaçları, DVM, Bayes ağları ve EYK algoritmalarını kullanarak infertiliteyi değerlendirmişlerdir. UCI makine öğrenimi deposundan alınan "fertilite" veri setinde kişinin yaşı, çocukken hastalık geçirmesi, kaza ya da travma durumu, ameliyat durumu, son yıllarda gözlenen yüksek ateş, alkol tüketimi, sigara kullanımı ve oturarak geçirilen süre değişkenlerinden oluşmaktadır. Toplam gözlem sayısı 100 kişidir ve değişkenler kategorik hale getirilmiştir. Sonuçta hastalık tanısında bu algoritmalarla tüm tanı testleri için yüksek değerler (%100) elde etmişler ve infertilite tanısı için kullanılması önermişlerdir (Palechor ve ark, 2016). Sonuçlar karşılaştırıldığında literatürdeki çalışmada tanı testlerin tam sonuç vermesi bir probleminin olabileceğini göstermektedir. Osaseri ve Agharese (2016) çalışmasında aynı veri seti için YSABM Yapay Sinir Ağları ve Bulanık Mantık (Artificial Neural Networks and Fuzzy Logic) sınıflayıcısı ile elde edilen sonuçları değerlendirmişlerdir. Eğitim ve test seti %70-%30 olarak bölünmüştür. Sonuçlara göre bu yöntemle %90 doğruluk oranı elde edilmiştir (Osaseri ve Agharese, 2016). Bu sonuçlar infertilite verisi sonuçları ile karşılaştırıldığında veri setinde kullanılan genetik faktörlerin ve SL algoritmasının literatüre ek katkı sağladığı görüldü.

İnfertilite verisi değişkenleri için yapılan analizlerde sperm konsantrasyon, FSH ve LH değerlerinin önemli değişkenler olduğu saptandı. En etkili değişken semen konsantrasyonu'dur. Polimorfizm genleri önem sırası da sy1291, gr/gr, b2/b3 şeklinde elde edildi. Nitekim Kumar ve Singh (2015), çalışmalarında infertilite için en büyük etkenin normal sınırlarda olmayan semen parametrelerinden kaynaklandığını belirtmişlerdir. İnfertilite verisi analizi sonucunda elde edilen değişkenlerin önemliliği ile ilgili bilgiler literatür ile elde edilen sonuçları desteklemektedir.

Çalışmada analiz yapılan ikinci veri seti peridontitis verisidir. En optimal bölünme doğruluk oranı açısından değerlendirildiğinde %60-%40 olduğunda elde edildi. Sonuçlardan da görüldüğü gibi farklı bölünmeler kullanıldığında performans değişmektedir. Bölünme %80-%20 olduğunda doğruluk oranı NB için yaklaşık %74, EAA RO için yaklaşık %82 iken, bölünme %60-%40 olduğunda ise DVM için yaklaşık %81, EAA RO için yaklaşık %85 yükselmektedir. Huang ve ark. (2003), çalışmalarında farklı veri setleri için doğruluk oranı performans ölçütü olarak kullanıldığında

sonuçların NB, DVM, C4.4 ve C4.5 için benzer olduğu ancak DVM'nin daha başarılı sonuçlar verdiği elde edilmiştir. Performans ölçütü olarak EAA kullanıldığında ise NB, DVM ve C4.4 benzer olduğu ancak C4.5 daha başarısız sonuçlar elde edilmiştir (Huang ve ark., 2003). Peridontitis verisi ile elde edilen doğruluk oranı sonuçlarına göre algoritmaların yakın sonuçlar verdiği ve DVM algoritmasının daha başarılı olduğu saptandı. EAA ile elde edilen sonuçlara bakıldığında EYK ve C4.5 algoritmalarının diğer algoritmalara göre daha başarısız sonuçlar elde edildi.

Peridontitis verisi SL algoritması kullanarak analiz edildiğinde en düşük risk değeri RO ile elde edildi. Kurulan SL modelinde en büyük katkıyı sırayla RO, SVM ve rpart sağladı. Analiz sonucunda SL'de EAA yaklaşık %85 elde edildi. Bölünme %60-%40 olduğunda RO ile elde edilen sonuç ile aynıdır. Bu sonuç bize SL algoritmasının analiz aşamasında algoritma seçme için yol gösterici olabileceğini göstermektedir. Polley ve Van der Laan (2010), çalışmalarında küçük veri setleri için de SL algoritmasının iyi performans elde edileceğini göstermişlerdir. Ağırlıklandırılmış SL ile elde edilen sonuç ya en iyi performansı gösterir ya da en iyiye en yakın performansın elde edilmesini sağlamaktadır (Polley ve Van der Laan, 2010). Peridontitis verisi ile elde edilen sonuç literatür sonucunda elde edilen sonuçlarla eşleşmektedir.

Peridontitis verisi 174 gözlemden oluşmaktadır. Bu veri setindeki değişkenler arasındaki ilişki yapısı kullanılarak örnek büyüklüğü 500 olan veri seti elde edildi. Burada en yüksek performans %60-%40 olduğunda EAA için NB ve DVM'de %78 elde edildi. Duyarlılık ve seçicilik açısından da incelendiğinde DVM ve EYK algoritmaları ile daha iyi sonuç elde edilmiştir. SL algoritması ile hesaplanan model için EAA değeri %81 olarak elde edildi. Bayesglm ve discrete SL ile benzer sonuçlar elde edildi. Ağırlıklı ortalamanın hesaplanması için kullanılan en küçük risk bayesglm için elde edildi. Bacak ve Kennedy (2018), çalışmalarında farklı öğreticiler kullanarak SL performansını hata oranına ve EAA performans ölçütleri göre değerlendirilmiştir. Birinci benzetim seti verisi ile elde edilen sonuçlar literatür sonuçları ile benzerlik göstermektedir.

İkinci benzetim veri seti peridontitis veri setindeki değişkenler arasındaki ilişki yapısı kullanılarak örnek büyüklüğü 1000 olan veri seti elde edildi. Veri seti için farklı bölünmeler kullanarak sonuçlar incelendiğinde en yüksek performans EAA için NB'de %85 olarak elde edildi. Örnek büyüklüğü arttığında eğitim ve test seti bölünmelerinin

etkisi kalmamıştır. SL ile hesaplanan EAA yaklaşık %86, discrete SL ile bayesglm'de ile aynıdır. ÇG ile hesaplanan SL için kullanılan en küçük risk bayesglm için elde edildi. Bu sonuçlarda literatür sonuçları ile benzerlik göstermektedir.

Peridontitis verisi değişkenleri için yapılan analizlerde mevcut diş sayısının, yaş ve çürük diş indeksi değerlerinin önemli değişkenler olduğu saptandı. En etkili değişken mevcut diş sayısıdır. Polimorfizm genleri önem sırası da Taq1, Apa1 ve Bsm1 şeklinde saptandı. Van Dyke ve Dave (2006), çalışmalarında bakteri enfeksiyonun etkili olduğunu elde etmişlerdir. Güneş ve ark. (2008), çalışmalarında polimorfizm genlerinin etkilerini sağlıklı kontrollerde ve kronik peridontitis hastaları için değerlendirmişlerdir. Peridontitis verisi analizi sonucunda elde edilen değişkenlerin önemliliği ile ilgili bilgiler literatür ile elde edilen sonuçları desteklemektedir.

İnfertilite veri seti için farklı bölünme oranları kullanıldığında performans değerlerinin farklılık gösterdiği elde edildi. Peridontitis veri setinde ise %60- %40 bölünme ile daha iyi performans değerleri elde edildi. İlk benzetim veri setinde %60-%40 olarak ayrıldığında en yüksek performanslar elde edildi. İkinci benzetim veri setinde ise bölünmenin herhangi bir etkisi olmadığı görüldü. Eğitim ve test bölünme oranları literatürde ki çalışmalar göz önüne alındığında verinin yapısı ile ilişkilidir. Raschka (2015) çalışmasında çok büyük veri setlerinde %90-%10 bölme kullanılabileceği belirtilmiştir. Chakraborty (2019) çalışmasında farklı bölünme yüzdeleri kullanarak tıbbi verilerin analizinde sınıflama modellerin yanında topluluk öğrenme yöntemleri ve k-katlı ÇG kullanılarak analizler gerçekleştirilmiştir. K-katlı ÇG için 10 ve 20 kat kullanılmıştır. Veri seti 303 gözlem, 6 sayısal değişken ve 8 kategorik değişkenden oluşmuştur. Analiz sonucunda en iyi bölünme %80-%20 olduğunda, RO algoritması ile elde edilmiştir. Bunu takip eden algoritmalar NB ve C4.5 algoritmasıdır (Chakraborty, 2019). Bu çalışmada kullanılan farklı büyüklükteki veriler için bölünme oranlarının etkileri gösterildi. Kullanılan algoritmaların bölme oranlarına göre değiştiği elde edildi.

6. SONUÇ ve ÖNERİLER

Çalışma sonuçları eğitim ve test seti bölünme oranlarının performans üzerinde etkili olduğunu göstermektedir. Bölünme oranları kullanılan algoritmayı da etkilemektedir. İnfertil veri seti peridontitis veri setine göre daha büyüktür. İnfertil veri setinin %80-%20 ve %70-%30 oranında bölünmelerin performansı arttırdığı görüldü. Bu veri setini dengesiz veri seti olarak işlemlere tabi tuttuğumuzda %60-%40 bölünme oranında daha iyi sonuçları elde edildi. Peridontitis veri seti orijinal hali ile en iyi bölünmenin bize %60-%40 olduğunda performansı arttırdığını göstermektedir. Orijinal veri özellikleri dikkate alınarak gözlem sayısını arttırdığımızda performans küçük oranda artmıştır.

SL algoritması aday öğreticilerden oluşan ve ÇG riskine göre hesaplanan bir ağırlaklındırılmış algoritmadır. Bu tez çalışmasından elde edilen sonuçlara göre minimum risk ile en yüksek performans gösteren algoritma bağlantılıdır. Ancak infertilite veri seti dengesiz olduğu için SL algoritması ile minimum riske ilişkin algoritma benzer elde edilmemiştir. Bunun nedenin, algoritmaların çoğunluğunun çoğunluk sınıfına göre işlem yapması sebebiyle olduğu söylenebilir.

Bir araştırıcı model kuracağı zaman literatürde yer alan farklı algoritmaları deneyerek sonuçları değerlendirmektedir. Ancak literatürde çok fazla algoritma yer almaktadır. Uygun algoritmayı seçmek zaman ve uzmanlık gerektirmektedir. SL bu aşamada, daha kısa zamanda araştırıcıya yol gösteren ve yüksek performans elde etmede önemli bir araçtır ve önerilmektedir. Bu çalışmada sınırlı sayıda algoritma kullanarak model elde edilmiştir. Ancak farklı algoritmalar değerlendirerek farklı modeller kurulabilir.

Bu çalışmada özellikle genetik faktörlerin yer aldığı veri setleri üzerine çalışmalar yapılmıştır. Böylece genetik hastalıkla ilişkilendirilen sonuçların mevcut veriler ile yüksek tahmin performansı elde edilmiştir. Böylece hastalıklarda etkin olan faktörler belirlenmeye çalışılmıştır. Gelecek çalışmalarda etiyolojisi açıklanamayan diğer hastalıkların tahmin performansının değerlendirilmesinde bu sonuçlar kullanılmak üzere değerlendirilmesi planlanmaktadır.

Sonraki çalışmalarda dengeli veri seti ve gözlem sayısı daha yüksek olan veriler farklı algoritmaların yer aldığı kombinasyonlar denenerek çalışmaların yapılması planlanmaktadır.

KAYNAKLAR

Abdallh, M.M.A, Bilal, K.H., Babiker, A. Machine learning (pattern recognition) review. International Journal of Engineering, Applied and Management Sciences Paradigms, 2016; 36 (1):10-16.

Acharya, U. R., Yu, W., Editorial data mining techniques in medical informatics. The Open Medical Informatics Journal, 2010; 4: 21-22.

Akbaş, B. Machine learning 101-herkes için ML. https://www.linkedin.com/pulse/machine-learning-101-herkes-için-ml-burç-akbaş, 2017. Erişim tarihi: 03.12.2019

Akosa, J. S. Predictive accuracy: A misleading performance measure for highly imbalanced data. Oklahoma State University, 2017; 942: 1-11.

Al-Aidaros K. M, Bakar A. A, Othman Z. Medical data classification with naive bayes approach. Information Technology Journal, 2012; 11(9): 1166-1174.

Aličković, E. Subaşı A. Data mining techniques for medical data set. The International Arab Conference on Information Technology, Jordan, 2011; 243-246.

Alpar, R. Uygulamalı İstatistik ve Geçerlilik-Güvenirlik, 4. Baskı, Ankara, Detay Yayıncılık. 2016; 80-81.

Alpaydın, E. Introduction to Machine Learning 2nd Edition. Massachusetts Institute of Technology. 2010; 1-323.

Amazon. Collecting labeled data. http://docs.aws.amazon.com/machine-learning/latest/dg/collecting-labeled-data.html, 2017. Erişim tarihi: 12.10.2019

Amazon. Splitting the data into training and Evaluation data. https://docs.aws.amazon.com/machine-learning/latest/dg/splitting-the-data-into-training-and-evaluation-data.html, 2017. Erişim tarihi: 12.10.2019

Amit Y., Geman, D. Shape quantization and recognition with randomized trees. Neural Computation, 1997; 9(7): 1545–1588.

Awad M., Khanna R. Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers, New York, Springer Verlag. 2015;1-2.

Babar, V. S., Ade, R. A review on imbalanced learning methods. International Journal of Computer Applications, 2015; 23-27.

Bacak, V. Kennedy, E. H. Principled machine learning using the superlearner: An application to predicting prison violence. Sociological Methods Research DOI: 10.1 177/0049 1241 17747301, 2018; 1-24.

Badr, W. Having an imbalanced dataset? Here is how you can fix it. https://towardsdatascience.com/having-an-imbalanced-dataset-here-is-how-you-can-solve-it-1640568947eb, 2019. Erişim Tarihi: 03.12.2019

Balaban M. E. Kartal E. Veri Madenciliği ve Makine Öğrenmesi Temel Algoritmaları ve R Dili ile Uygulamaları. Çağlayan kitabevi, İstanbul, Net Kırtasiye Tanıtım ve Matbaa San. 2015; 45-112.

Barnett, A. Santokhi, J. Simpson, M. Smart N. P. Stainton-Bygrave, C. Vivek, S. Waller, A. Image classification using non- linear support vector machines on encrypted data. IACR Cryptology Print Archive, 2017; 857.

Batchelor, B.G. Pattern Recognition: Ideas in Practice. Plenum Press, Heidelberg, 1978; 68-72.

Bayer, H., Çoban, T. Web istatistiklerinde makine öğrenmesi algoritmaları ile kritik parametre tespiti. Electronic Journal of Vocational Colleges- Special Issue: The Latest Trends in Engineering, 2015; 23-41.

Ben-Gal I. Outlier detection. Data Mining and Knowledge Discovery Handbook: A Complete Guide for Practitioners and Researchers, Kluwer Academic Publishers. 2005; 1-2.

Berzal, F. Cubero, J. C.,Marin N., Sanchez D. Building multi-way decision trees with numerical attributes. Information Sciences, 2004; 165:73-90.

Bhatia N. Vandana. Survey of nearest neighbor techniques. International Journal of Computer Science and Information Security, 2010; 8(2): 302-305.

Bishop, C. M. PatternRecognition and Machine Learning. Springer Science Business Media, Cambridge, LLC. 2006; 291-292.

Boyle, T. Dealing with imbalanced data. https://towardsdatascience.com/methods-for-dealing-with-imbalanced-data-5b761be45a18,2019. Erişim tarihi: 12.10.2019

Breiman, L., Friedman, J. H., Olshen, R. A., Stone, C. J. Classification and regression trees: book review. Cytometry, 1987; 8: 534-535.

Breiman, L. Stacked regression. Machine Learning, 1996; 24: 49-64.

Breiman, L.Random forests. Machine learning, 2001; 45(1): 5–32.

Brownlee J. How to prepare data for machine learning. https://machinelearningmastery.com/how-to-prepare-data-for-machine-learning/, 2013. Erişim tarihi: 09.10.2019

Brownlee J. Supervised and unsupervised machine learning algorithms. https://machinelearningmastery.com/how-to-prepare-data-for-machine-learning/, 2016. Erişim tarihi: 09.10.2019

Brownlee, L. Naive bayes for machine learning. https://machinelearningmastery.com/naive-bayes-for-machine-learning/, 2016. Erişim tarihi: 09.10.2019

Bulut, F. Sınıflandırıcı topluluklarının dengesiz veri kümeleri üzerindeki performans analizleri. Bilişim Teknolojileri Dergisi, 2016; 9(2): 153-159.

Camastra F, Vinciarelli A. Machine Learning for Audio, Image and Video Anlaysis. Springer, Spin, 2007; 86.

Chakraborty, Chinmay. Advanced Classification Techniques for Healthcare Analysis. IGI Global, Mesra, 2019; 3-12.

Chawla, N. V., Bowyer, K. W., Kegelmeyer, W. P. SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 2002; 16: 321-357.

Cortes, C., Vapnik, V. Support vector networks. Machine Learning, 1995; 20: 273-297.

Coşgun E., Karaağaoğlu E. Veri madenciliği yöntemleriyle mikrodizilim gen ifade analizi. Hacettepe Tıp Dergisi, 2011; 42: 180-189.

Cover, T.M., Hart, P.E. Nearest neighbor pattern classification. IEEE Trans, Inform, Theory, 1967; 13(1): 21–27.

Cunningham, P., Delany S. J. K nearest neighbour classifiers. Technical Report UCD-CSI, 2007; 1-17.

Das, S., Dey A., Pal, A., Roy, N. Applications of artificial intelligence in machine learning: review and prospect. International Journal of Computer Applications, 2015; 115(9): 31-41.

Data Flair. Kernel functions-introduction to SVM kernel and examples. https://data-flair.training/blogs/svm-kernel-functions/, 2018. Erişim tarihi: 06.09.2019

DataPreparation. https://paginas.fe.up.pt/~ec/files_1112/week_03_Data_Preparation.pdf, 2017. Erişim Tarihi: 09.10.2019.

Demirci M. Bayes teoremi ve işletme bölümünde uygulamaları. International Journal of Social Science, 2016; 43: 439-462.

Deo, R. C. Machine learning in medicine. PMC, 2018; 132(20): 1920–1930.

Deza, E.,Deza, M. M. Encyclopedia of distances. Springer, 2009; 3-4.

Dhriti, K. M., Kaur, M. K-nearest neighbor classification approach for face and fingerprint at feature level fusion. International Journal of Computer Applications 2012; 60(14): 13-17.

Diri B., Makine öğrenmesine giriş, machine learning. https://docplayer.biz.tr/6389888-Makine-ogrenmesine-giris-machine-learning-ml.html, 2014. Erişim tarihi: 09.10.2019

Domingos P. Pazzani M. On the optimality of the simple bayesian classifier under zero-one loss. Machine Learning, 1997; 29: 103–130.

D'Souza, J. A trip to random forest. https://medium.com/greyatom/a-trip-to-random-forest-5c30d8250d6a, 2018. Erişim tarihi: 03.12.2019.

Elitedatascience. Modern machine learning algorithms: strengths and weeknesses. https://elitedatascience.com/machine-learning-algorithms, 2019. Erişim tarihi: 13.10.2019

Fawagreh, K., Gaber, M.M., Elyan, E. Random forests: from early developments to recent advancements. Systems Science & Control Engineering: An Open Access Journal, 2014; 2(1): 602-609.

Fix, E., Hodges, J.L. Discriminatory analysis, nonparametric discrimination: consistency properties. Technical Report 4, USAF School of Aviation Medicine, Texas, Randolph Field, 1951.

Foote, K. A brief history of machine learning. https://www.dataversity.net/a-brief-history-of-machine-learning/ , 2019. Erişim tarihi: 11.10.2019.

Fournier D. Cr´emilleux B. A Quality index for decision tree pruning. Elsevier, 2002; 15(2): 37-43.

Fu, W. J. Carroll, R. J. Wang, S. Estimating mislassification error with small samples via bootstrap cross- validation. Bioinformatics, 2005; 21(9): 1979–1986.

Fumo, D. Types of machine learning algorithms you should know. https://towardsdatascience.com/types-of-machine-learning-algorithms-you-should-know-953a08248861, 2017. Erşim tarihi: 09.12.2019.

Ghosh, A. K. On optimum choice of k in nearest neighbor classification. Computational Statistics and Data Analysis, 2006; 50: 3113–3123.

Goncalves, L., Subtil, A.,Oliveira, M. R., Bermudez, P. D. Z. ROC curve estimation: an overview. Statistical Journal, 2014; 12(1): 1–20.

Gümüşçü A., Taşaltın R., Aydilek İ. B. C4.5 Karar ağaçlarında genetik algoritma ile budama. Dicle Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 2016; 5(2): 77-80.

Güneş, S., Sümer, A. P., Keles, G. C., Kara, N., Köprülü, H., Bağcı, H., Bek, Y. Analysis of vitamin D receptor gene polymorphisms in patient with chronic peridontitis. Indian Journal Medical Research, 2008; 127: 58-64.

Han, J., Kamber, M., Pei, J. Data Mining Concepts and Techniques. Third edition, Morgan Kaufman Publications, USA, Elsevier,. 2012; 5-609.

Harrington, P. Machine Learning in Action. Manning Publications, Island, 2012; 7-10.

Hasan, A. M. Xu, S. Kabir, J. Ahmad, S. Performance evaluation of different kernels for support vector machine used in intrusion detection system. International Journal of Computer Networks and Communications, 2016; 8 (6): 39-53.

Hassanat A. B., Abbadi M. A., Altarawneh G. A., Alhasanat A. A. Solving the problem of the K parameter in the KNN classifier using an ensemble learning approach. International Journal of Computer Science and Information Security, 2014; 12(8): 33-39.

Hekim, N., Gure, M.A., Mahmutoğlu, M.,Güneş, S., Ascı, R., Henkel, R. SNP's in xenobiotic metabolism and male infertility. Xenobiotica. doi: 10.1080/00498254.2019.1616850, 2019; 1-8.

Hertzmann, A., Fleet, D. Machine Learning and Data Mining Lecture Notes. Computer Science Department University of Toronto, CSC 411 / CSC D11, 2010; 42.

Ho, T. K. Random decision forests. In document analysis and recognition. Proceedings of the third international conference, Quebec 1995; 1: 278–282.

Ho, T. K. The random subspace method for constructing decision forests. Intelligence, Transactionson Pattern Analysis and Machine, 1998; 20(8): 832–844.

Hofmann, M. Support vector machines - kernels and the kernel trick. Uni Bamberg, 2006; 3.

Hordri, N. F., Yuhaniz, S.S., Azmi, N. F. M., Shamsuddin, S. M. Handling class imbalance in credit card fraud using resampling methods. International Journal of Advanced Computer Science and Applications, 2018; 9 (11), 390-396.

Horning, N. Random Forests : An algorithm for image classification and generation of continuous fields data sets. International Conference on Geoinformatics for Spatial Infrastructure Development in Earth and Allied Sciences; 2010.

Howley, T., Madden, M. G. The genetic evolution of kernels for support vector machine classifiers. 15th Irish Conference on Artificial Intelligence, Cognitive Science; 2004.

Hssina B. Merbouha A.,Ezzikouri H.,Erritali M. A comparative study of decision tree ID3 and C4.5. International Journal of Advanced Computer Science and Applications, Special Issue on Advances in Vehicular Ad Hoc Networking and Applications, 2014; 13-19.

Hu, L.Y., Huang, M.W., Ke, S. W.,Tsai C. F. The distance function effect on k-nearest neighbor classification for medical datasets. Hu et al, Springer Plus, 2016; 5(1304): 2-9.

Huang J., Lu J., Ling C.X. Comparing naive bayes, decision trees, and SVM with AUC and Accuracy. Third IEEE International Conference on Data Mining, 2003; 553–556.

IBM Corp. Released. IBM SPSS Statistics for Windows, Version 21.0. Armonk, NY: IBM Corp; 2012.

Japkowicz, N. Learning from imbalanced data sets: a comparison of various strategies, Technical Report, 2000; 00(05): 10- 15.

Jeni, L., Cohn, J. F., Torre, F. Facing imbalanced data recommendations for the use of performance metrics. DOI: 10.1109/ACII.2013.47.

John H. G., Langley P. Estimating continous distributions in bayesian classifiers. Proceeding of the 11.th Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers, San Mateo, 1995.

Karakoyun, M. Hacıbeyoğlu, M. Biyomedikal veri kümeleri ile makine öğrenmesi sınıflandırma algoritmalarının istatistiksel olarak karşılaştırılması. Dokuz Eylül Üniversitesi, Mühendislik Fakültesi Mühendislik Bilimleri Dergisi, 2014; 16(48): 30-41.

Karatzoglou, A. Meyer, D. Hornik, K. Support vector machines in R. Journal of Statisical Software, 2006; 15(9): 1-25.

Karlık, B, Yibre, M A., Koçer, B. Comparising feature selection and classifier methods with SMOTE for prediction of male infertility. International Journal of Fuzzy Systems and Advanced Applications, 2016; 3: 1-6.

Kataria, A., Singh, M. D. A Review of data classification using K nearest neighbour algorithm. International Journal of Emerging Technology and Advanced Engineering, 2013; 3 (6): 354–360.

Kaur, P., Gosain, A. Comparing the behavior of oversampling and undersampling approach of class imbalance learning by combining class imbalance problem with noise. Advances in Intelligent Systems and Computing 2018; 653: 23-30.

Kavzoğlu T. Çölkesen İ. Karar ağaçları ile uydu görüntülerinin sınıflandırılması: Kocaeli örneği. Harita Teknolojileri Elektronik Dergisi, 2010; 2(1): 36-45.

Kaynar O, Yıldız M, Görmez Y, Albayrak A. Makine öğrenmesi yöntemleri ile duygu analizi. International Artificial Intelligence and Data Processing Symposium, Malatya; 2016, 234-241.

Kayri, M. Kayri, İ. The comparison of gini and twoing algorithms in Terms of predictive ability and misclassification cost in data mining: an empirical study. International Journal of Computer Trends and Technology, 2015; 27 (1): 21-30.

Kecman, V. Support vector machines- an introduction. Study Fuzzy, 2005; 177: 1–47.

Kennedy, C. Guide to superlearner. https://cran.r-project.org/web/packages/SuperLearner/vignettes/Guide-to-SuperLearner.html , 2017. Erişim Tarihi: 18.11.2019.

Kılınç, D., Borandağ, E., Yücealar, F., Tunalı, V., Şimşek, M., Özçivit, A. KNN algoritması ve R dili ile metin madenciliği kullanılarak bilimsel makale tasnifi. Marmara Fen Bilimleri Dergisi, 2016; 3: 89-94.

Kırçiçek, O. Ensemble learning. https://medium.com/@oguzkircicek/ensemble-learning-7ec8c9a7227f , 2019. Erişim tarihi: 12.10.2019

Korkmaz S. Caret ile sınıflandırma algoritmaları ve uygulamaları. XIX. Ulusal ve II. Uluslar Arası Biyoistatistik Kongresi- Uygulamaları, 2017; 1-19.

Kotsiantis, S. B. Supervised machine learning: a review of classification techniques. Informatica, 2007; 31: 249-268.

Kourou, K., Exarchos, T. P. , Exarchos K. P. , Karamouzis, M. V. , Fotiadis, D. I. Machine learning applications in cancer prognosis and prediction. Computational and Structural Biotechnology Journal, 2015; 13: 8–17.

Kowalczyk, B. Support Vector Machines. Succinctly e- book Morrisville, 2017; 48-53.

Kuhn M, Johnson K. Applied Predictive Modelling. Springer, New York, 2013;159-265.

Kumar, N., Singh, A. K. Trends of male factor infertility, an important cause of infertility: a review of literature. Journal Human Reprod Science, 2015; 8(4): 191–196.

Lantz, B. Machine Learning with R. Second Edition. Packt publishing, 2015; 19-331.

Liu B. Web data mining: exploring hyperlinks, contents, and usage data. Data-Centric Systems and Applications, DOI 10.1007/978-3-642-19460-3(3), Springer-Verlag Berlin Heidelberg, 2011; 63-79.

Louppe, G. Understanding random forest from theory to practice. University of Liège Faculty of Applied Sciences Department of Electrical Engineering and Computer Science, Liege, 2015; 61: 71-72.

Lunardon, N. ,Menardi, G. Torelli, N. ROSE: A package for binary imbalanced learning. The R Journal, 2014; 6(1): 79-89.

Mademir, Naive bayes sınıflandırma algoritması. http://www.mademir.com/2011/01/naive-bayes-snflandrma-algoritmas.html, 2011. Erişim Tarihi: 09.10.2019

Mahmood, A. M. Imran, M. Satuluri, N. Kuppa, M. R. Rajesh, V. An improved CART decision tree for datasets with irrelevant feature. Springer-Verlag Berlin, Heidelberg, 2011; 539-549.

Mail B. Bayes sınıflandırıcılar. http://mail.baskent.edu.tr/~20410964/DM_9.pdf, 2018. Erişim tarihi: 09.10.2019

Maimon, O., Rokach L. Data Mining for Imbalanced Data Sets: An Overview. Data Mining and Knowledge Discovery Handbook, Chapter 4, 2005; 860.

Makhabel B, Learning Data Mining with R. Packt publishing, Birmingham, Mumbai, 2015; 82-89.

Marin-Reyes, P. A, Lorenzo-Navarro, J., Castrillon-Santana, M. Comparative study of histogram distance measures for re-identification. arXiv:1611.08134v1, 2016.

Marr, B. A short history of machine learning—Every manager should read. https://www.forbes.com/sites/bernardmarr/2016/02/19/a-short-history-of-machine-learning-every-manager-should-read/#310dfcca15e7, 2016. Erişim Tarihi: 09.10.2019.

Marsland S, Machine Learning An Algorithmic Perspective. CRC Press, Boca Raton, 2015; 24-257.

Menardi, G. Torelli, N. Training and assessing classification rules with unbalanced data. Working Paper Series, 2010; 2: 5-12.

Menzel K. Ekonomide istatistiksel yöntemlere giriş ders notları 4. https://docplayer.biz.tr/9831255-Mit-opencourseware-http-ocw-mit-edu-14-30-ekonomide-istatistiksel-yontemlere-giris-bahar-2009.html , 2009. Erişim Tarihi: 15.10.2019

Merih, K. Machine learning I- genel tanımlar. https://datalabtr.com/index.php/2017/03/27/5142/, 2017. Erişim Tarihi: 09.10.2019

Michalski R.S, Stepp R.E, Diday E. A recent advance in data analysis: clustering objects into classes characterized by conjunctive concepts. A Progress in pattern recognition, Amsterdam, 1981; 33–56.

Michie D, Spiegelhalter D. J, Taylor J.J, Machine Learning, Neural and Statistical Classification. Cambridge, 1994; 1-217.

Mitchell T. M. Machine Learning. McGraw Hill Science Publisher, 1997; 1-231.

Mohri, M. Rostamizadeh, A., Talwalker, A. Foundations of Machine Learning. The MIT Press, 2012; 7-8.

Mountassir, A., Bennrahim, H., Berrada, I. Addressing the problem of unbalanced data sets in sentiment analysis. Proceedings of the International Conference on Knowledge Discovery and Information Retrieval, 2012; 306-311.

Murphy, K. P. Machine Learning A Probabilistic Perspective. The MIT Press Cambridge, London, 2012; 3-9.

Namal, D. Makineler düşünebilir mi? https://medium.com/türkiye/turing-testi-2b87097ae6f0, 2018. ErişimTarihi: 09.10.2019

Nath, D. S. 9 Baby steps to start with machine learning. https://medium.com/@deepusnath/9-baby-steps-to-start-with-machine-learning-fe3f31b83fe, 2016. Erişim Tarihi: 13.10. 2019.

Navlani, A. Understanding random forests classifiers in python. https://www.datacamp.com/community/tutorials/random-forests-classifier-python, 2018. ErişimTarihi: 09.06.2019.

Nilson N. J. The Quest For Artificial Intelligence A History of Ideas and Achievements. Cambridge University Press, 2009; 503-506.

Nitze, I., Schulthess, U., Asche, U. Comparison of machine learning algorithms random forest, artificial neural network and support vector machine to maximum likelihood for supervised crop type classification. Proceedings of the 4th GEOBIA, Brazil, 2012; 35-40.

Nizam H, Akın S. S. Sosyal medyada makine öğrenmesi ile duygu analizinde dengeli ve dengesiz veri setlerinin performanslarının karşılaştırılması. https://docplayer.biz.tr/7310009-Sosyal-medyada-makine-ogrenmesi-ile-duygu-analizinde-dengeli-ve-dengesiz-veri-setlerinin-performanslarinin-karsilastirilmasi.html , 2014. Erişim Tarihi: 10.12.2019

Noi, P. T., Kappas, M. Comparison of random forest, K-nearest neighbor and support vector machine classifiers for land cover classification using sentinel-2 imagery. sensors 18, doi:10.3390/s18010018, 2017; 2-20.

Oğuzlar, A. Veri ön işleme. Erciyes Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, 2003; 21: 67-76.

Olgun M. O. Özdemir G. İstatistiksel özellik temelli bayes sınıflandırıcı kullanarak kontrol grafiklerinde örüntü tanıma. Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi, 2012; 27 (2): 303-311.

Oluk,E. Makine öğrenimi kavramı ve kullanım alanları. https://www.gazetebilkent.com/2019/04/13/makine-ogrenimi-kavrami-ve-kullanim-alanlari-nursen-erginsoy/ ,2019. ErişimTarihi: 09.10.2019

Orhan U. Makine öğrenmesi ders notları. http://bmb.cu.edu.tr/uorhan/DersNotu/Ders04.pdf, 2012. Erişim Tarihi: 15.10.2019.

Osaseri, R. Agharese, U. R. Predicting male fertility using soft computing approach. Proceedings of Academics World International Conference, Dublin, 2016; 1-3.

Ovla H. D. Taşdelen B. Aykırı değer yönetimi. Mersin Üniversitesi Sağlık Bilimleri Dergisi, 2012; 5(3): 1-8.

Paginas, Classification. https://paginas.fe.up.pt/~ec/files_1011/week%2008%20-%20Decision%20Trees.pdf, 2008. Erişim tarihi: 09.10. 2019

Pal, M. Random forest classifier for remote sensing classification**.** International Journal of Remote Sensing, 2005; 26 (1): 217-222.

Palechor, M., Fabio, E., Colpas A. Paola, P.,Ojeda, S., Jorge, A., Manotas, A., Melo, P.M. Fertility analysis method based on supervised and unsupervised data mining techniques. International Journal of Applied Engineering Research, 2016; 11(21): 10374-10379.

Patel,S. Chapter2: SVM support vector machine theory. https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory-f0812effc72, 2017. Erişim Tarihi: 15.10.2019.

Patil, D. D, Wadhai V. M, Gokhale, J. A. Evaluation of decision tree pruning algorithms for complexity and classification accuracy. International Journal of Computer Applications, 2010; 11(2): 23-30.

Peng R. D. R Programing for Data Science. Lean Publishing, 2015; 28.

Peterson, L. E. K- nearest neighbor. Scholarpedia, 4(2), doi:10.4249/scholarpedia.1883 2009.

Polamuri S. Difference between classification and regression in machine learning. http://dataaspirant.com/2014/09/27/classification-and-prediction/, 2014. Erişim tarihi: 13.10.2019.

Polley, E. C., Van der Laan, M. J. Super learner in prediction. U.C. Berkeley Division of Biostatistics Working Paper Series, 2010; 266: 1-19.

Pradhan, A. Support vector machines: a survey. International Journal of Emerging Technology and Advanced Engineering, 2012; 2(8): 82- 85.

Prasad, D.D., Rao, K. N. An improved approach on class imbalance data using within-class minority oversampling technique. International Journal of Latest Trends in Engineering and Technology, 2017; 7(4): 156-164.

Prasath V. B. S., Alfeilat H. A. A, Lasassmeh O. Hassanat A. B. A. Distance and Similarity Measures Effect on the Performance of K-Nearest Neighbor Classifier – A Review. arXiv:1708.04321v1, 2017; 1-50.

Priya, N. Improve machine learning results for seman analysis using ensemble meta classification. International Journal of Advanced Research in Computer Science, 2017; 8(8): 692-694.

Provalis, R. A brief history of machine learning. https://provalisresearch.com/blog/brief-history-machine-learning/ , 2017. Erişim Tarihi: 11.10.2019.

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

Quinlan, J. R. Book review: C4.5 programs for machine learning. Machine Learning, 1993; 16: 235-240.

Raschka, S. Python Machine Learning. Packt publishing, Birmingham-Mumbia, 2015; 3-109.

Raschka, S. Model evaluation, model selection and algorithm selection in machine learning, arXiv:1811.12808v2, 2018; 8-20.

Raza K, Hasan A. N. A comprehensive evaluation of machine learning techniques for cancer class prediction based on microarray data. Cornell University Library, 2013; 1-8.

Refaeilzadeh, P. Tang, L. Liu, H. Cross- validation. Encyclopedia of database systems, 2009; 532-538.

Rosenblatt F. The perceptron: a probabilistic model for information storage and organization in the brain. Psychological Review, 1958; 65(6): 386-408.

Sajda, P. Machine learning for detection and diagnosis of disease. Annual Review Biomed, 2006; 8:8.1–8.29.

Samuel, A. L. Some studies in machine learning using the game of checkers. IBM Journal of Research and Development, 1959; 3(3): 210-229.

Santoyo. S. A brief overview of outlier detection techniques. https://towardsdatascience.com/a-brief-overview-of-outlier-detection-techniques-1e0b2c19e561, 2017. Erişim tarihi: 12.10.2019.

Saraswat, M. Practical tutorial on random forest and parameter tuning in R. https://www.hackerearth.com/practice/machine-learning/machine-learning-algorithms/tutorial-random-forest-parameter-tuning-r/tutorial/, 2018. Erişim tarihi: 09.06.2019

Sayad, S. Support vector machine. http://www.saedsayad.com/support_vector_machine.htm, 2018. Erişim Tarihi: 15.10.2019

Schapire, R. E. The stength of weak learnability. Machine Learning, 1990; 5: 197-227.

Schapire, R. COS 511: Theoretical machine learning. http://www.cs.princeton.edu/courses/archive/spr03/cs511/scribe_notes/0204.pdf, 2008. Erişim tarihi: 08.10.2019.

Scornet, E. Biau, G. Vert J. P. Consistency of random forest. The Annals of Statistics, 2015; 43(4): 1716-1741.

Scornet, E. Tuning parameters in random forest. Proceedings and Surveys, 2018; 60: 144-162.

Sennaar, Kumba. Machine learning in genomics current efforts and future applications. https://emerj.com/ai-sector-overviews/machine-learning-in-genomics-applications/, 2019. Erişim tarihi: 08.10.2019

Shannon, C. E. A mathematical theory of communication. The Bell System Technical Journal, 1948; 27: 1-55.

Sharma, Rahul. 6 awesome application powered by machine learning. http://techgenix.com/machine-learning-medical-applications/, 2019. Erişim tarihi: 08.10.2019.

Sheth A. History of machine learning. https://medium.com/bloombench/history-of-machine-learning-7c9dc67857a5, 2017. Erişim tarihi: 09.10.2019.

Shutz, Machine learning: the power and promise of computers that learn by example. The Royal Society, Des, 2017; 4702: 22.

Shwartz, S. Ben-David, S. Understanding machine learning: from theory to algorithms. Library of Congress Cataloging in Publication Data, 2014; 7: 19-25.

Singh, S. Giri M. Comparative study ID3, Cart And C4.5 decision tree algorithm: a survey. International Journal of Advanced Information Science and Technology 2014; 3 (7): 47-52.

Sinisi, S. E. Polley, E. C. Petersen, M. L. Rhee, S. Y. Van der Laan, M. Super learning: an application to the Prediction of HIV-1 drug resistance. Statistical Application Genetic Molecular Biology, 2007; 6 (7): 1-25.

Smola, A., Vishwanathan, S.V.N. Introduction to Machine Learning. Cambridge University Press, 2008; 3-7.

Software, Naive bayes classification algorithm. http://software.ucv.ro/~cmihaescu/ro/teaching/AIR/docs/Lab4-NaiveBayes.pdf, 2017. Erişim Tarihi: 15.10.2019

Sokolova M. Assessing invariance properties of evaluation measures. https://www.researchgate.net/publication/228983741, 2006. Erişim tarihi: 10.10.2019

Song, Y., Huang J., Zhou D., Zha H., Giles C. L. Informative K-nearest neighbor pattern classification. Springer-Verlag Berlin, Heidelberg, Kok et al., 2007; 248-264.

Statsoft. Support vector machines. http://www.statsoft.com/textbook/support-vector-machines, 2018. ErişimTarihi: 15.10.2019

Sunasra, M. Performance metrics for classification problems in machine learning-Part I. https://medium.com/greyatom/performance-metrics-for-classification-problems-in-machine-learning-part-i-b085d432082b, 2017. Erişim tarihi: 13.10.2019.

Tan, P. N., Steinbach M., Karpatne, A., Kumar. V. Introduction to data mining: 2nd Edition. https://www-users.cs.umn.edu/~kumar001/dmbook/ch4.pdf , 2004; 148-149.

Tantithamthavorn, C., Hassan, E. A., Matsumoto, K. The impact of class rebalancing techniques on the performance and interpretation of defect prediction models. arXiv:1801.10269v1, 2018; 1-20.

Tripathi, H. What is balanced and unbalanced data set. https://medium.com/analytics-vidhya/what-is-balance-and-imbalance-dataset-89e8d7f46bc5,2019. Erişim tarihi: 03.12.2019

Turing, A. Computing machinery and intelligence. Mind, 1950; 49: 433-460.

Turiplatform. Turi machine learning platform user guide. Regression. https://turi.com/learn/userguide/supervised-learning/regression.html, 2017. Erişim tarihi: 13.10.2019

Turiplatform. Turi machine learning platform user guide. Classification metrics. https://turi.com/learn/userguide/evaluation/classification.html#receiver-operating-characteristic-roc-curve--, 2018. Erişim tarihi: 13.10.2019.

Van der Laan, M. J., Dudoit, S. Unified cross-validation methodology for selection among estimators and a general cross-validated adaptive epsilon-net estimator: finite sample oracle inequalities and Examples. The Berkeley Electronic Press, 2003; 130; 1-103.

Van der Laan, M.J., Dudoit S., Van der Waart, A. W. The cross- validated adaptive epsilon-net estimator. Statistics and Decisions, 2006; 24 (3): 373-395

Van der Laan, M. J. ,Polley, E. C. Hubbard, A. E. Superlearner. U.C. Berkeley Division of Biostatistics Working Paper Series, 2007; 222: 1-20.

Van Dyke, T. Dave, S. Risk factors for peridontitis. Journal of International Academy of Peridontitis, 2006; 7(1): 3–7.

Van Rijmenam, M. How to prepare for an automated future: 7 steps to machine learning. https://vanrijmenam.nl/prepare-for-automated-future-7-steps-machine-learning/,2019. ErişimTarihi: 09.10.2019.

Vijaykumar, B., Vikramkumar, T. Bayes and naive- bayes classifier. Computer Science and Engineering, arXiv:1404.0933; 2014.

Wang, H., Xu, Q., Zhou, L. Seminal quality prediction using clustering-based decision forests. Algorithms 2014; 7: 405-417.

Welling, M. A first encounter with machine learning. Donald Bren School of Information and Computer Science University of California Irvine, 2011; 6-7.

Wettschereck, D.,Aha, D.W., Mohri, T. A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms. Artificial Intelligence Review 1997; 11: 273–314.

Wikibooks, Support vector machines. https://en.wikibooks.org/wiki/Support_Vector_Machines, 2018. Erişim tarihi: 14.11.2019

Witten I. H., Frank E. Data mining practical machine learning tools and techniques. Morgan Kaufmann Publishers, Elsevier, San Francisco, 2005; 92-93.

Wolpert, D.H. Stacked generalization. Complex Systems Group, Theoretical Division, and Center for Non-linear Studies, Los Alamos, 1992; 505: 1-57.

Yang X.S, Deb S, Loomes M, Karamanoglu M. A framework for self-tuning optimization algorithm. Neural Computing and Applications, 2013; 23(7-8): 2051-2057.

Yufeng G. The 7 Steps in machine learning.

https://towardsdatascience.com/the-7-steps-of-machine-learning-2877d7e5548e, 2017. Erişim tarihi: 13.10.2019.

Zhang Y. New advances in machine learning. INTech Open, DOI: 10.5772/225, 2010; 19-20.

Zhang, C., Ma Y. Ensemble machine learning methods and applications. Springer Verlag, London, 2012; 1-2.

Zheng, A. Evaluating machine learning models. O' Reilly Media, 2015; 3,7.

Zhou, Z. H. Ensemble methods foundations and algorithms. Taylor and Francis Group, New York, 2012; 15-17.

Zprzydatek, M. ID3 algorithm- decision trees- machine learning. https://mariuszprzydatek.com/2014/11/11/iterative-dichotomiser-3-id3-algorithm-decision-trees-machine-learning/, 2014. Erişim tarihi: 13.10.2019.

EKLER

EK-1 Etik Kurul Raporu

T.C.
ONDOKUZ MAYIS ÜNİVERSİTESİ
KLİNİK ARAŞTIRMALAR ETİK KURULU

Sayı: B.30.2.ODM.0.20.08/943-1011 22.06.2017

Sayın Yrd. Doç. Dr. Leman Tomak

Etik Kurulumuza sunmuş olduğunuz **Genetik Alanında Elde Edilen Verilerin Makine Öğrenimi Algoritmaları Yardımıyla Karşılaştırılarak En Etkin Yöntemin Belirlenmesi** başlıklı OMÜ KAEK 2017/208 Karar nolu İstatistiksel Yöntem Değerlendirilmesi nitelikli araştırma projeniz amaç, gerekçe, yaklaşım ve yöntemle ilgili açıklamaları açısından Klinik Araştırmalar Etik Kurulu yönergesine göre incelenmiş ve etik açıdan bir sakınca olmadığına, çalışmanın süresi 6 ayı geçerse 6 aylık bildirimlerinin yapılmasına, çalışma tamamlandıktan sonra sonucunun tarafımıza en geç üç(3) ay içerisinde bildirilmesine 11.05.2017 tarihli Etik kurulumuzda oy birliği ile karar verilmiştir.

Bilgilerinize arz/rica ederim.

Prof.Dr.Dursun AYGÜN
Klinik Araştırmalar Etik Kurulu Başkanı

Ondokuz mayıs Üniversitesi Tıp Fak. Klinik Araştırmalar Etik Kurulu Tel:(0362)3121919/2782 -4576007 Omutaek@gmail.com
Hastane içi 1.Kat (Özel servis karsısı) Atakum/SAMSUN

ÖZ GEÇMİŞ

Adı Soyadı **:** Senem Koç

Doğum Yeri **:** Kassel (Almanya)

Doğum Tarihi **:** 12.09.1979

Medeni Hali **:** Bekar

Bildiği Yabancı Diller **:** Almanca, İngilizce, İtalyanca

Eğitim Durumu (Kurum ve Yıl): Ondokuz Mayıs Ünivsersitesi, Fen- Edebiyat Fakültesi, İstatistik-2002, Lisans.

: Ondokuz Mayıs Ünivsersitesi, Fen- Bilimleri Enstitüsü, İstatistik-2008, Yüksek Lisans.

Çalıştığı Kurum/Kurumlar ve Yıl : Samsun İl Özel İdaresi Ar-Ge Daire Başkanlığı-(2010-2014)

: Samsun Büyükşehir Belediyesi, Ondokuz Mayıs İlçesi, SASKİ (2014-2015)

E-posta **:** kocsenem79@hotmail.com