DOKUZ EYLÜL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

SIRALI KÜME ÖRNEKLEMESİNE DAYALI MAKİNE ÖĞRENMESİ TEKNİKLERİ

Sena ASLAN

Nisan, 2022
İZMİR

SIRALI KÜME ÖRNEKLEMESİNE DAYALI MAKİNE ÖĞRENMESİ TEKNİKLERİ

Dokuz Eylül Üniversitesi Fen Bilimleri Enstitüsü
Yüksek Lisans Tezi
İstatistik Anabilim Dalı, Veri Bilimi Programı

Sena ASLAN

Nisan, 2022
İZMİR

YÜKSEK LİSANS TEZİ SINAV SONUÇ FORMU

SENA ASLAN, tarafından **DOÇ. DR. TUĞBA YILDIZ** yönetiminde hazırlanan **"SIRALI KÜME ÖRNEKLEMESİNE DAYALI MAKİNE ÖĞRENMESİ TEKNİKLERİ"** başlıklı tez tarafımızdan okunmuş, kapsamı ve niteliği açısından bir Yüksek Lisans tezi olarak kabul edilmiştir.

Doç. Dr. Tuğba YILDIZ

Yönetici

Dr. Öğr. Üyesi Özge ELMASTAŞ GÜLTEKİN

Jüri Üyesi

Doç. Dr. Neslihan DEMİREL

Jüri Üyesi

Prof. Dr. Okan FISTIKOĞLU

Müdür

Fen Bilimleri Enstitüsü

TEŞEKKÜR

Tez çalışmamın her aşamasında içten tavırlarıyla yanımda olan, bütün sorularımı sabırla dinleyen, bilgi ve tecrübelerini benimle paylaşarak yol gösteren değerli danışmanım Doç. Dr. Tuğba YILDIZ'a teşekkür ederim.

Yardımlarıyla ve fikirleriyle çalışmama katkıda bulunan arkadaşım Yusuf Can SEVİL'e teşekkür ederim.

Hayatım boyunca gösterdikleri sonsuz sevgi ile yanımda olan, her zaman beni dinleyen, hayallerimi destekleyen annem Meltem ASLAN ve babam Metin ASLAN'a teşekkür ederim.

Sena ASLAN

SIRALI KÜME ÖRNEKLEMESİNE DAYALI MAKİNE ÖĞRENMESİ TEKNİKLERİ

ÖZ

Son yıllardaki hızlı veri artışı ve bu verileri analiz etmenin giderek zorlaşmasıyla birlikte, günümüzde birçok çalışma alanında makine öğrenmesi kullanılmaktadır. Makine öğrenmesi, insan beyninin öğrenme mantığını esas alarak, çeşitli algoritma ve teknikler geliştirmeyi amaçlayan bilimsel çalışma alanıdır. Makine öğrenmesi algoritmaları, olayları inceler ve nasıl meydana geldiklerini anlamaya çalışır. Bu çabaları sonucunda, elde ettikleri sonuçlar ile genelleme yapma yeteneği kazanırlar. Makine öğrenmesi algoritmalarının bilgileri öğrenmeleri ve ne kadar iyi öğrendiklerinin değerlendirilmesi için; veri kümesi, eğitim ve test seti olarak ikiye ayrılır. Literatürde bu işlem, kullanıcının belirlediği bir oranda rastgele olarak yapılır. Bu tez çalışmasında; veri seti bölme işlemi, literatürde kullanılan yönteme ek olarak, Sıralı Küme Örneklemesi (SKÖ), Uç Değer SKÖ (USKÖ), Medyan SKÖ (MSKÖ) ve Yüzdelik SKÖ (YSKÖ) yöntemleriyle yapılarak, elde edilen sonuçların karşılaştırılması amaçlanmıştır. Farklı çalışma alanlarından seçilen gerçek hayat verileri, belirtilen yöntemler ile eğitim ve test setlerine ayrılmıştır. Eğitim setleri ile makine öğrenmesi algoritmaları eğitilerek, test setleri ile öğrenme başarıları sınanmıştır. Karşılaştırma kriteri olarak; regresyon algoritmalarında hata kareler ortalamasının karekökü (HKOK) değerleri, sınıflandırma algoritmalarında ise doğru sınıflandırma oranları kullanılmıştır.

Anahtar kelimeler: Makine öğrenmesi, sıralı küme örneklemesi, uç değer sıralı küme örneklemesi, medyan sıralı küme örneklemesi, yüzdelik sıralı küme örneklemesi

MACHINE LEARNING TECHNIQUES BASED ON RANKED SET SAMPLING

ABSTRACT

With the rapid increase in data in recent years and the increasing difficulty of analyzing this data, machine learning is used in many fields of study today. Machine learning is a scientific field of study that aims to develop a variety of algorithms and techniques based on the learning logic of the human brain. Machine learning algorithms study events and try to understand how they occur. As a result of these efforts, they gain the ability to generalize with obtained results. For machine learning algorithms to learn information and to evaluate how well they learn; the data set is split into a training and test set. In the literature, this process is performed randomly at a rate determined by the user. In this thesis study, in addition to the method used in the literature, data set splitting is done by Ranked Set Sampling (RSS), Extreme RSS (ERSS), Median RSS (MRSS), Percentile RSS (PRSS) methods. It is aimed to compare the results obtained. Real life data sets selected from different workspaces are split into training and test sets with the specified methods. Machine learning algorithms were trained with training sets and learning achievements were tested with test sets. As a comparison criteria; root mean square error (RMSE) values were used in the regression algorithms, and the accurate classification rate values were used in the classification algorithms.

Keywords: Machine learning, ranked set sampling, extreme ranked set sampling, median ranked set sampling, percentile ranked set sampling

İÇİNDEKİLER

Sayfa

ŞEKİLLER LİSTESİ

Sayfa

TABLOLAR LİSTESİ

Sayfa

KISALTMALAR

BRÖ	: Basit Rastgele Örnekleme
SKÖ	: Sıralı Küme Örneklemesi
USKÖ	: Uç Değer Sıralı Küme Örneklemesi
MSKÖ	: Medyan Sıralı Küme Örneklemesi
YSKÖ	: Yüzdelik Sıralı Küme Örneklemesi
HKOK	: Hata Kareler Ortalamasının Karekökü
HKO	: Hata Kareler Ortalaması
OMH	: Ortalama Mutlak Hata
DP	: Doğru Pozitif
YN	: Yanlış Negatif
YP	: Yanlış Pozitif
DN	: Doğru Negatif
ÇDR	: Çoklu Doğrusal Regresyon
EKK	: En Küçük Kareler
TBR	: Temel Bileşenler Regresyonu
KEKKR	: Kısmi En Küçük Kareler Regresyonu
RR	: Ridge Regresyon
LR	: Lasso Regresyon
ENR	: Elastik Net Regresyon
KNN	: K-En Yakın Komşuluk
DVM	: Destek Vektör Makineleri
DVR	: Destek Vektör Regresyonu
YSA	: Yapay Sinir Ağları
SRA	: Sınıflandırma ve Regresyon Ağaçları
TA	: Torbalama Algoritması
RO	: Rastgele Orman
GYM	: Gradyan Yükseltme Makineleri
AGY	: Aşırı Gradyan Yükseltme Makineleri
LRA	: Lojistik Regresyon Analizi

SEMBOLLER LİSTESİ

m	: Küme sayısı
r	: Tekrar sayısı
$X_{i(j:m)}$	: i. kümede j. sıralı birim
$X_{i(j:m)n}$	: n. tekrarda i. kümede j. sıralı birim
$X_{i[n]}$	: n. tekrarda i. sıralı birim
$\bar{X}_{SK\ddot{O}}$	: SKÖ ile elde edilen kitle ortalaması tahmini
$V(\bar{X}_{SK\ddot{O}})$	: SKÖ ile elde edilen kitle ortalamasının varyans tahmini
$\bar{X}_{USK\ddot{O}}$	: USKÖ ile elde edilen kitle ortalaması tahmini
$V(\bar{X}_{USK\ddot{O}})$	: USKÖ ile elde edilen kitle ortalamasının varyans tahmini
$\bar{X}_{MSK\ddot{O}}$	: MSKÖ ile elde edilen kitle ortalaması tahmini
$V(\bar{X}_{MSK\ddot{O}})$	: MSKÖ ile elde edilen kitle ortalamasının varyans tahmini
p	: Yüzdelik değer
$\bar{X}_{YSK\ddot{O}}$	: YSKÖ ile elde edilen kitle ortalaması tahmini
$V(\bar{X}_{YSK\ddot{O}})$	: YSKÖ ile elde edilen kitle ortalamasının varyans tahmini
k	: K-katlı çapraz doğrulama yöntemindeki kat sayısı
e_i	: Gözlem ve tahmin değerleri arasındaki fark (artık değeri)
β_0	: ÇDR denklemindeki sabit değer
β_j	: Regresyon katsayıları
Y	: Yanıt değişkeni vektörü
X	: Bağımsız değişkenler matrisi
β	: Regresyon katsayıları vektörü
ε	: Hata terimleri vektörü
$\hat{y}_i$	: i. gözleme ait bağımlı değişken tahmini
$\hat{\beta}_0$	: Sabit değer tahmini
$x_i{}'$	: Bağımsız değişkenlerin tahminleri matrisinin transpozu
$\hat{\beta}$	: β'nın tahmin edicisi
y_i	: Gözlem değerleri
X'	: Bağımsız değişkenler matrisinin transpozu
σ^2	: Kitle varyansı

$E(\hat{\beta})$	: $\hat{\beta}$'nın beklenen değeri
$V(\hat{\beta})$	: $\hat{\beta}$'nın varyansı
P	: $X'X$'in öz vektör matrisi
P'	: $X'X$'in öz vektör matrisinin transpozu
D	: $X'X$'in öz değerlerinin matrisi
W	: Temel bileşenler matrisi
W'	: Temel bileşenler matrisinin transpozu
γ	: Temel bileşenler regresyonu katsayı tahmini
T, U	: Skor matrisleri
P, Q	: Yükler matrisleri
E, F	: Artık matrisleri
w, c	: Ağırlık değerleri
$cov(t, u)$	: t ve u skor matrisleri arasındaki kovaryans
λ_R	: Ridge regresyondaki ayarlama parametresi
$\hat{\beta}_{RR}$	: $\hat{\beta}$'nın ridge regresyon tahmin edicisi
I	: Birim matris
$V(\hat{\beta}_{RR})$	: $\hat{\beta}$'nın ridge regresyon tahmin edicisinin varyansı
$\hat{\beta}_{Lasso}$	: $\hat{\beta}$'nın lasso regresyon tahmin edicisi
λ_L	: Lasso regresyondaki ayarlama parametresi
$\hat{\beta}_{EN}$	: $\hat{\beta}$'nın elastik net regresyon tahmin edicisi
k	: KNN algoritmasındaki en yakın komşu sayısı
x_q	: Sınıflandırılacak gözlem değeri
$x_1, \dots, x_k$	: En yakın komşu değerleri
$\hat{f}(x_q)$	: Sınıf belirleme fonksiyonu
x_i, y_i	: Nokta değerleri
p	: Sabit sayı
x_i	: Gelecek vektörü
y_i	: Hedef çıktısı
$f(x)$	: DVR'daki tahmin değerleri
w	: Model parametre vektörü
φ	: Doğrusal olmayan haritalama fonksiyonu

b	: Sapma miktarı
w^T	: Model parametre vektörünün transpozu
C	: Ceza, karmaşıklık parametresi
$\pm\varepsilon$	: Kabul edilebilir hata miktarı
ξ_i, ξ_i^*	: Kabul edilebilir hata miktarı üzerine eklenecek değerler
λ_i, λ_i^*	: Lagrange çarpanları
γ	: Kernel boyutu
ξ	: Pozitif bir değişken
L_D	: Lagrange çarpanları kullanılarak elde edilen eşitlik
α_i	: Sıfır ile ceza parametresi aralığındaki Lagrange çarpanı
$K(x_i, x_j)$	: İki noktanın çekirdek fonksiyonu
$x_1, \dots, x_n$	: Girdi değerleri
$W_1, \dots, W_n$	: Ağırlık değerleri
t_P	: Üst düğüm
t_l	: Üst düğümün bölünmesiyle elde edilen sol alt düğüm
t_r	: Üst düğümün bölünmesiyle elde edilen sağ alt düğüm
x_j	: j. bağımsız değişken
M	: Değişken sayısı
N	: Gözlem sayısı
K	: Toplam sınıf sayısı
x_j^R	: x_j'nin en iyi bölme değeri
$i(t)$	: Safsızlık fonksiyonu
$\Delta i(t)$	: Üst düğüm safsızlığı ile alt düğüm beklenen değer farkı
$i(t_P)$	: Üst düğümün safsızlık fonksiyonu
$i(t_c)$	: Sol ve sağ alt düğümleri
$E[i(t_c)]$	: Sol ve sağ alt düğümlerin beklenen değeri
P_l	: Sol alt düğüm olasılığı
P_r	: Sağ alt düğüm olasılığı
$V(Y_l)$	: Bağımlı değişkenin sol alt düğümünün varyansı
$V(Y_r)$	: Bağımsız değişkenin sol alt düğümünün varyansı
n	: Örneklem sayısı

m	: Değişken sayısı
k	: RO algoritmasındaki oluşturulacak ağaç sayısı
d	: Ağaç derinliği
K	: İterasyon sayısı
α	: Öğrenme oranı
η	: Ağaç oluşumunda kullanılacak minimum gözlem sayısı
$L(y, f(x))$	: Kayıp fonksiyonu
r^m	: Artık değerleri
τ_k	: Ağaçların yaprak değerleri
$\Omega(f_i)$	: Düzenlileştirme terimi
C	: Sabit terim
g_i	: Kayıp fonksiyonunun birinci dereceden türevi
h_i	: Kayıp fonksiyonunun ikinci dereceden türevi
T	: Ağaçlardaki yaprak sayısı
γ, α	: Yaprak ve ağırlık değerlerini kontrol eden parametreler
λ	: modelin hatalarını kontrol eden ayarlama parametresi
$E(Y\|X)$	: Koşullu ortalama değeri
P_i	: Kategorik değişkende istenilen olayın olması olasılığı
T_i	: Lojistik regresyon modeli
e^{T_i}	: Odds oranı
$g(x)$	: Logit dönüşüm fonksiyonu
lne^{T_i}	: Odds oranının logaritması
D	: Katsayıların önem kontrolü için kullanılan istatistik
G	: 1 serbestlik derecesi ile ki-kare dağılan istatistik

BÖLÜM 1
GİRİŞ

Yapılan araştırma ya da çalışma alanıyla ilgili birimlerin tamamı kitleyi oluşturur (Demirhan ve Hamurkaroğlu, 2015). Ancak, araştırma sürecinde kitle ile çalışmak maliyet, zaman ve emek bakımından zorlayıcı olmaktadır. Bu sebeple, kitleyi temsil edebilecek daha az miktarda birim seçilir. Kitleden seçilen bu birime örneklem denir. Yapılan seçim işlemi örnekleme, seçim sürecinde kullanılan yöntemler ise örnekleme yöntemleri olarak adlandırılır. Araştırmalarda, kitlenin yapısına uygun örnekleme yöntemi belirlenir ve seçilen yönteme göre örneklem büyüklüğü hesaplanır. Bilinen en eski örnekleme yöntemi basit rastgele örneklemedir (BRÖ). BRÖ'de, ilgilenilen kitleden rastgele olarak n tane birim seçilir. Seçilen n tane birim örneklemi oluşturur. Bu yöntemin avantajı, kitle hakkında çok fazla bilginin olmasını gerektirmemesidir. Ancak, bu yöntemle örneklem seçmek çok zaman almakta ve yüksek maliyet gerektirmektedir. Sıralı küme örneklemesi (SKÖ), örneklem birimlerinin ölçümünün zor ve maliyetli, ilgilenilen değişken bakımından ise görsel olarak sıralanmasının kolay olduğu durumlarda, BRÖ yöntemine alternatif olarak McIntyre (1952) tarafından önerilmiştir (Cihantimur Özkan, 2014). McIntyre (1952) yaptığı çalışmada meraların verimini tahmin ederek, SKÖ'de sıralamada yapılabilecek hataları ve korelasyon problemlerini göstermiştir. Daha sonra Halls ve Dell (1966), Doğu Teksas'ta bir çam ormanındaki yem verimini tahmin etmek için SKÖ'yü kullanmışlar ve çalışmaları sonucunda SKÖ'nün, BRÖ'den daha verimli olduğunu öne sürmüşlerdir. Takahasi ve Wakimoto (1968), sıralama çok iyi yapıldığında örneklem ortalamasının kitle ortalaması için yansız bir tahmin edici olduğunu göstermiş ve SKÖ ile edilen ortalama varyansının, BRÖ ile edilen ortalama varyansından küçük olduğunu belirtmişlerdir. Ayrıca Dell ve Clutter (1972) da, mükemmel sıralama olmadığı durumu inceleyerek aynı sonuca ulaşmışlardır. Dell ve Clutter (1972) ile David ve Levin (1972), sıralama hatalarına teorik çözümler öneren ilk kişilerdir. Stokes (1977), SKÖ'de yardımcı değişken kullanımını incelemiştir. Birkaç yıl sonra Stokes (1980), sıralama hatası olduğu durumlarda kitle varyansı tahmini üzerine çalışmıştır. Stokes ve Sager (1988), SKÖ'nün dağılım fonksiyonunu tahmin etmeye yönelik bir çalışma yapmışlardır. SKÖ ile regresyon tahmini üzerine Patil, Sinha ve

Taillie (1993), Yu ve Lam (1997), Chen (2001) çalışmışlardır. Bohn ve Wolfe (1994) ve Hettmansperger (1995); işaret testi, sıralı işaret testi ve Mann-Whitney-Wilcoxon testi gibi yöntemleri SKÖ'ye uygulamışlardır. Samawi ve Muttlak (1996) çalışmalarında SKÖ'yü kullanarak kitle oranı tahmini yapmışlardır. SKÖ'ye dayalı U-istatistiği Presnell ve Bohn (1999) tarafından geliştirilmiştir. Chen (1999) sıralı küme örneklemesi verilerini kullanarak, yoğunluk fonksiyonu tahmini üzerine çalışmıştır. Parametrik ve parametrik olmayan SKÖ yöntemleri ile ilgili, Bai ve Chen (2003) tarafından genel bir teori ileri sürülmüştür. Chen ve Shen (2003), çoklu yardımcı değişken varlığında SKÖ prosedürü üzerine çalışmışlardır. SKÖ yöntemi için yakın zamanlarda yapılan çalışmalar bulunmaktadır. Wolfe (2012), SKÖ'nün çıkarsamalı istatistik üzerindeki etkisini incelemiştir. Zamanzade ve Vock (2015), SKÖ'de yardımcı değişken varlığında varyans tahmini yapmışlardır. Z. Zhang, Liu ve Zhang (2016), sıralı küme örnekleriyle; parametrik yöntemlerin sağlaması gereken varsayımları dikkate almadan, kitle ortalaması ve iki kitle ortalaması arası fark için aralık tahmini ve hipotez testine, parametrik olmayan bir yaklaşım önermişlerdir. Öztürk (2018), sonlu bir kitleden sıralı küme örneğine dayalı oran tahmini ile ilgili bir çalışma yapmıştır. SKÖ hakkında daha fazla bilgi edinmek için Patil, Sinha ve Taillie (1999), Chen, Bai ve Sinha (2003) çalışmaları incelenebilir.

Literatürde, klasik SKÖ prosedüründen yola çıkılarak; sıralamadaki hataları azaltmak ve kitle parametrelerini daha doğru tahmin edebilmek için geliştirilen birçok yöntem vardır. İlk olarak; Samawi, Ahmed ve Abu-Dayyeh (1996) sadece maksimum ve minimum birimlerin kullanıldığı uç değer sıralı küme örneklemesi (USKÖ) yöntemini önermişlerdir. Daha sonra, Muttlak (1997) sıralama hatalarını azaltmak için kümelerdeki ortanca birimlerin kullanıldığı medyan sıralı küme örneklemesini (MSKÖ) geliştirmiştir. Muttlak (2003) diğer bir çalışmasında, kitle ortalamasını tahmin etmek için yüzdeliklerin kullanıldığı yüzdelik sıralı küme örneklemesi (YSKÖ) yöntemini önermiştir.

Makine öğrenmesi, günümüzde hemen hemen her alanda kullanılır hale gelmiştir. Temeli, insanların öğrenme mantığına dayanarak oluşturulmuştur. İnsanlar yaşadıkları olaylardan tecrübe edinirler ve edindikleri tecrübelere göre kararlarını ya da

davranışlarını değiştirirler. Makine öğrenmesinde de benzer şekilde, bilgisayar kendisine gösterilen örnekleri ya da olayları inceleyerek öğrenir ve yeni meydana gelecek olaylar için tahmin yapma yeteneği kazanır.

Makine öğrenmesi ilk olarak, araştırmacıların yapay zeka arayışından ortaya çıkmıştır. Makine öğrenmesi kavramı bir öngörü olarak, Turing (1950) tarafından "*Makineler düşünebilir mi?"* sorusuyla ortaya atılmıştır (s. 433). Bu sorudan yola çıkarak Turing testi geliştirilmiştir. Bu testte, bir kişi klavye sistemiyle karşısında kim olduğunu bilmeden sorular sormuştur. Aldığı cevaplara göre karşısındakinin bilgisayar mı yoksa insan mı olduğuna karar vermiştir. Kişi bilgisayarın, insan olduğuna karar verirse makine Turing testini geçmiş sayılmaktadır (Akman ve Blackburn, 2000). Daha sonra Samuel (1959) dama oyunu için bir program geliştirmiş ve makine öğrenmesi tabirini ilk defa kullanmıştır. Bu programla, her dama oyununda bir önceki oyunun hatalarını göz önünde bulundurarak oyunu kazanmak için daha iyi stratejiler üretmek amaçlanmaktadır. Bu gelişmelerden 1980'lere kadar, soyut zihin ve bilgi tabanlı sistemler üzerine çalışmalar yapılmıştır. 1990'larda, verilerin hızlı artışı ve bu verilerin analizlerinin zorlaşması sebebiyle makine öğrenmesi çalışmaları hız kazanmıştır. Günümüzde, birçok çalışma alanında makine öğrenmesi kullanılmaktadır (Çelik ve Altunaydın, 2018). İkinci el satış sitelerinde fotoğraftan ürünün tanınması, akıllı telefonların klavyelerindeki otomatik cümle tamamlama özelliği, e-ticaret sitelerinin tavsiye sistemleri, bankaların kredi başvuru değerlendirmeleri makine öğrenmesine örnek verilebilir.

Makine öğrenmesi algoritmalarına bilginin öğretilmesi ve ne kadar iyi öğrendiğinin test edilmesi için veri seti bölme stratejisi kullanılır. Literatürde, veri setini bölme işlemi; kullanıcının belirlediği bir oranda, veri setinden rastgele gözlemler seçilerek yapılır. Örneğin, veri setinin %80'i rastgele seçilerek eğitim seti, kalan %20'si ise test seti olarak tanımlanabilir.

Bu tez çalışmasında, literatürde kullanılan yönteme ek olarak; SKÖ, USKÖ, MSKÖ ve YSKÖ yöntemleri ile veri setleri bölünerek elde edilen sonuçlar değerlendirilmektedir. Bu amaçla, gerçek hayat veri setleri; rastgele olarak ve SKÖ,

USKÖ, MSKÖ, YSKÖ yöntemleri kullanılarak eğitim ve test setlerine ayrılmıştır. Eğitim setleri ile makine öğrenmesi algoritmaları eğitilerek, test setleri ile ne kadar iyi öğrendikleri sınanmıştır. Performans ölçütü olarak; regresyon algoritmalarında, hata kareler ortalamasının karekökü (HKOK) değerleri, sınıflandırma algoritmalarında ise, doğru sınıflandırma oranları kullanılmıştır. Çalışmanın ikinci bölümünde, SKÖ ve bazı modifikasyonlarının çalışma prensipleri açıklanmıştır. Makine öğrenmesi ve terminolojisi üçüncü bölümde anlatılmıştır. Dördüncü bölümde, tez çalışmasında kullanılan makine öğrenmesi algoritmaları açıklanmıştır. Beşinci bölümde ise, veri seti bölme yöntemleri ve makine öğrenmesi algoritmalarının gerçek hayat verileri üzerinde uygulama sonuçlarına yer verilmiştir. Elde edilen sonuçların değerlendirilmesi altıncı bölümde yapılmıştır.

BÖLÜM 2
SIRALI KÜME ÖRNEKLEMESİ YÖNTEMİ VE BAZI MODİFİKASYONLARI

2.1 Sıralı Küme Örneklemesi

SKÖ, örneklem birimlerinin ölçümünün zor ve maliyetli olduğu durumlarda kitle parametrelerini daha doğru bir şekilde elde edebilmek için McIntyre (1952) tarafından geliştirilmiştir. Çevre, ekoloji, tarım, tıp, biyoloji gibi ilgilenilen değişkenin ölçümünün yapılması zor ve maliyetli, görsel olarak sıralanmasının kolay olduğu çalışma alanlarında SKÖ kullanımı tercih edilir (İskit, 2018).

SKÖ'de iki önemli parametre vardır. Bu parametrelerden biri küme sayısı (m), diğeri ise tekrar sayısı (r) olarak adlandırılır. İki parametre de kullanıcı tarafından belirlenir. Küme sayısı genellikle, 2, 3, 4 veya 5 değerlerini alır (McIntyre, 1952; Cihantimur Özkan, 2014). Daha büyük bir küme sayısı seçilmesi, işlemleri zorlaştırarak birimlerin sıralanmasında hatalar meydana getirebilir.

SKÖ, iki aşamalı olarak düşünülebilir. İlk aşamada, kitleden m büyüklüğünde m tane rastgele örneklem seçilir. Bu örneklemlerin her biri küme olarak adlandırılır. Her küme için seçilen örnekler, gerçek bir ölçüm olmadan, ilgilenilen değişkene göre küçükten büyüğe doğru sıralanır. m birim ölçülene kadar bu işlem devam eder. Diğer aşamada, birinci kümeden en küçük birim, ikinci kümeden en küçük ikinci birim, m. kümeden en küçük m. birim seçilir. Bu işlem, istenilen miktarda birim elde edilinceye kadar r kez tekrar edilir. Bu işlemler sonucunda, $n = mr$ boyutunda sıralı küme örneklemi elde edilir ve kitle parametreleri tahmin edilir.

Tek tekrarlı SKÖ prosedürü aşağıdaki gibi çalışır:

$$\begin{bmatrix} \boldsymbol{X_{1(1:m)}} \leq X_{1(2:m)} \leq \cdots \leq X_{1(m:m)} \\ X_{2(1:m)} \leq \boldsymbol{X_{2(2:m)}} \leq \cdots \leq X_{2(m:m)} \\ \vdots \\ X_{m(1:m)} \leq X_{m(2:m)} \leq \cdots \leq \boldsymbol{X_{m(m:m)}} \end{bmatrix}$$

$X_{i(j:m)}$; $i = 1,2,\dots,m$ ve $j = 1,2,\dots,m$ olduğunda, i. kümede j. sıralı birimi temsil eder. İlk kümeden en küçük sıralı birim olan $X_{1(1:m)}$, ikinci kümeden en küçük sıralı ikinci birim olan $X_{2(2:m)}$, m. kümeden en küçük sıralı m. birim olan $X_{m(m:m)}$ seçilir.

r tekrarlı SKÖ prosedürü aşağıdaki gibi tanımlanır:

1. Tekrar

$$\begin{bmatrix} \boldsymbol{X_{1(1:m)1}} \le X_{1(2:m)1} \le \cdots \le X_{1(m:m)1} \\ X_{2(1:m)1} \le \boldsymbol{X_{2(2:m)1}} \le \cdots \le X_{2(m:m)1} \\ \vdots \\ X_{m(1:m)1} \le X_{m(2:m)1} \le \cdots \le \boldsymbol{X_{m(m:m)1}} \end{bmatrix}$$

2. Tekrar

$$\begin{bmatrix} \boldsymbol{X_{1(1:m)2}} \le X_{1(2:m)2} \le \cdots \le X_{1(m:m)2} \\ X_{2(1:m)2} \le \boldsymbol{X_{2(2:m)2}} \le \cdots \le X_{2(m:m)2} \\ \vdots \\ X_{m(1:m)2} \le X_{m(2:m)2} \le \cdots \le \boldsymbol{X_{m(m:m)2}} \end{bmatrix}$$

n. Tekrar

$$\begin{bmatrix} \boldsymbol{X_{1(1:m)n}} \le X_{1(2:m)n} \le \cdots \le X_{1(m:m)n} \\ X_{2(1:m)n} \le \boldsymbol{X_{2(2:m)n}} \le \cdots \le X_{2(m:m)n} \\ \vdots \\ X_{m(1:m)n} \le X_{m(2:m)n} \le \cdots \le \boldsymbol{X_{m(m:m)n}} \end{bmatrix}$$

$X_{i(j:m)n}$; $i = 1,2,\dots,m$, $j = 1,2,\dots,m$ ve $n = 1,2,\dots,r$ olduğunda, n. tekrarda i. kümede j. sıralı birimi temsil eder.

Küme sayısı m ve tekrar sayısı r için oluşturulan sıralı küme örneği aşağıdaki gibi gösterilir:

$$\begin{bmatrix} X_{[1]1} & X_{[2]1} & \dots & X_{[m]1} \\ X_{[1]2} & X_{[2]2} & \dots & X_{[m]2} \\ & \vdots & & \\ X_{[1]r} & X_{[2]r} & \dots & X_{[m]r} \end{bmatrix}$$

$X_{[i]n};\ i = 1,2,\dots,m,\ n = 1,2,\dots,r$ olduğunda, n. tekrardaki i. sıralı gözlemi gösterir.

McIntyre (1952), Monte Carlo simülasyonunu kullanarak SKÖ'den elde edilen kitle ortalaması tahmininin, BRÖ'den elde edilen kitle ortalaması tahminine göre daha etkili olduğunu göstermiştir (Zamanzade ve Al-Omari, 2016). SKÖ yönteminden elde edilen kitle ortalaması ve varyans tahmini Eşitlik (2.1) ve Eşitlik (2.2)'deki gibidir.

$$\bar{X}_{SK\ddot{O}} = \frac{1}{rm}\sum_{n=1}^{r}\sum_{i=1}^{m} X_{[i]n} \tag{2.1}$$

$$V(\bar{X}_{SK\ddot{O}}) = \frac{1}{rm^2}\sum_{n=1}^{r}\sum_{i=1}^{m} V\left(X_{[i]n}\right) \tag{2.2}$$

SKÖ prosedüründen yola çıkılarak geliştirilen farklı yöntemler vardır. Bu yöntemler, kitle parametreleri için daha güvenilir tahminler yapabilmek amacıyla geliştirilmiştir. Klasik SKÖ prosedürü ile bu yöntemler arasındaki en temel fark, seçilen birimlerin sıralamalarıdır. Bu çalışmada uç değer, medyan ve yüzdelik sıralı küme örneklemesi yöntemleri üzerinde durulacaktır.

2.2 Uç Değer Sıralı Küme Örneklemesi

USKÖ, her kümede sıralanmış birimlerin minimum ve maksimum değerleri seçilerek, kitle parametrelerinin daha etkili tahminlerini elde etmek amacıyla Samawi vd. (1996) tarafından geliştirilmiştir. SKÖ'nün geliştirilen ilk modifikasyonudur.

USKÖ prosedürü aşağıdaki gibidir:

Adım 1: Kitleden her biri m boyutunda m tane rastgele küme seçilir.

Adım 2: Her küme içindeki birimler, görsel bir incelemeyle ya da ilgilenilen değişkenle ilişkili olan bir değişken yardımıyla sıralanır.

Adım 3: Küme boyutu çift sayı ise, en küçük sıralı birimler $m/2$ setinden, en büyük sıralı birimler diğer $m/2$ setinden seçilir. Küme boyutu tek sayı olduğunda, en küçük sıralı birimler $(m-1)/2$ setlerinden, en büyük sıralı birimler diğer $(m-1)/2$ setlerinden ve ortanca birim kalan son kümenin $m.$ birimi olarak seçilir.

Adım 4: İşlemler mr birim elde edilene kadar r kez tekrarlanır.

$m = 5$ olarak alındığında tek tekrarlı USKÖ aşağıdaki gibi gösterilir:

$$\begin{bmatrix} \boldsymbol{X_{1[1:5]}} \le X_{1[2:5]} \le X_{1[3:5]} \le X_{1[4:5]} \le X_{1[5:5]} \\ \boldsymbol{X_{2[1:5]}} \le X_{2[2:5]} \le X_{2[3:5]} \le X_{2[4:5]} \le X_{2[5:5]} \\ X_{3[1:5]} \le X_{3[2:5]} \le X_{3[3:5]} \le X_{3[4:5]} \le \boldsymbol{X_{3[5:5]}} \\ X_{4[1:5]} \le X_{4[2:5]} \le X_{4[3:5]} \le X_{4[4:5]} \le \boldsymbol{X_{4[5:5]}} \\ X_{5[1:5]} \le X_{5[2:5]} \le \boldsymbol{X_{5[3:5]}} \le X_{5[4:5]} \le X_{5[5:5]} \end{bmatrix}$$

Küme boyutu m tek sayı olduğu için, ilk iki kümeden en küçük sıralı birimler ($X_{1[1:5]}$, $X_{2[1:5]}$); diğer iki kümeden en büyük sıralı birimler ($X_{3[5:5]}$, $X_{4[5:5]}$) ve kalan son kümeden ortanca sıradaki birim ($X_{5[3:5]}$) seçilir.

$m = 6$ olarak alındığında tek tekrarlı USKÖ aşağıdaki gibi çalışır:

$$\begin{bmatrix} \boldsymbol{X_{1[1:6]}} \le X_{1[2:6]} \le X_{1[3:6]} \le X_{1[4:6]} \le X_{1[5:6]} \le X_{1[6:6]} \\ \boldsymbol{X_{2[1:6]}} \le X_{2[2:6]} \le X_{2[3:6]} \le X_{2[4:6]} \le X_{2[5:6]} \le X_{2[6:6]} \\ \boldsymbol{X_{3[1:6]}} \le X_{3[2:6]} \le X_{3[3:6]} \le X_{3[4:6]} \le X_{3[5:6]} \le X_{3[6:6]} \\ X_{4[1:6]} \le X_{4[2:6]} \le X_{4[3:6]} \le X_{4[4:6]} \le X_{4[5:6]} \le \boldsymbol{X_{4[6:6]}} \\ X_{5[1:6]} \le X_{5[2:6]} \le X_{5[3:6]} \le X_{5[4:6]} \le X_{5[5:6]} \le \boldsymbol{X_{5[6:6]}} \\ X_{6[1:6]} \le X_{6[2:6]} \le X_{6[3:6]} \le X_{6[4:6]} \le X_{6[5:6]} \le \boldsymbol{X_{6[6:6]}} \end{bmatrix}$$

Küme boyutu m çift sayı olduğu için, ilk üç kümeden en küçük sıralı birimler ($X_{1[1:6]}$, $X_{2[1:6]}$, $X_{3[1:6]}$); son üç kümeden en büyük sıralı birimler ($X_{4[6:6]}$, $X_{5[6:6]}$, $X_{6[6:6]}$) seçilir.

Küme boyutu tek sayı olduğunda, USKÖ'nün kitle ortalaması ve varyans tahmini aşağıdaki gibi tanımlanır:

$$\bar{X}_{USK\ddot{O}} = \frac{1}{m}\left(\sum_{i=1}^{m-1/2} X_{2i-1[1:m]} + \sum_{i=1}^{m-1/2} X_{2i[m:m]} + X_{m[(m-1/2):m]} \right) \tag{2.3}$$

$$V(\bar{X}_{USK\ddot{O}}) = \frac{1}{m^2}\left(\sum_{i=1}^{m-1/2} V\left(X_{2i-1[1:m]}\right) + \sum_{i=1}^{m-1/2} V\left(X_{2i[m:m]}\right) + V\left(X_{m[(m+1)/2:m]}\right) \right) \tag{2.4}$$

Küme boyutu çift sayı olduğunda, USKÖ'nün kitle ortalaması ve varyans tahmini aşağıdaki gibidir:

$$\bar{X}_{USK\ddot{O}} = \frac{1}{m}\left(\sum_{i=1}^{m/2} X_{2i-1[1:m]} + \sum_{i=1}^{m/2} X_{2i[m:m]} \right) \tag{2.5}$$

$$V(\bar{X}_{USK\ddot{O}}) = \frac{1}{m^2}\left(\sum_{i=1}^{m/2} V\left(X_{2i-1[1:m]}\right) + \sum_{i=1}^{m/2} V\left(X_{2i[m:m]}\right) \right) \tag{2.6}$$

2.3 Medyan Sıralı Küme Örneklemesi

MSKÖ, sıralamada oluşabilecek hataları azaltarak kitle parametrelerini tahmin etmek için Muttlak (1997) tarafından önerilmiştir. MSKÖ prosedürü aşağıdaki gibi tanımlanır:

Adım 1: Kitleden her biri m boyutta m adet küme rastgele olarak seçilir.

Adım 2: Her bir kümedeki birimler, yardımcı bir değişken yardımıyla görsel olarak sıralanır.

Adım 3: Küme boyutu m tek sayı olduğunda, her kümeden $\big((m+1)/2\big)$. sıralı birim seçilir. m çift sayı ise, ilk $m/2$ setlerinden $(m/2)$. sıralı birimler, diğer $m/2$ setlerinden $\big((m+2)/2\big)$. sıralı birimler seçilir.

Adım 4: Bu işlemler, mr birim elde edilene kadar r kere tekrar edilir.

$m = 3$ olarak seçildiğinde tek tekrarlı MSKÖ aşağıdaki gibi gösterilir:

$$\begin{bmatrix} X_{1[1:3]} \leq \boldsymbol{X_{1[2:3]}} \leq X_{1[3:3]} \\ X_{2[1:3]} \leq \boldsymbol{X_{2[2:3]}} \leq X_{2[3:3]} \\ X_{3[1:3]} \leq \boldsymbol{X_{3[2:3]}} \leq X_{3[3:3]} \end{bmatrix}$$

Küme boyutu m tek sayı olduğundan, bütün kümelerin en küçük sıralı ikinci birimleri ($X_{1[2:3]}$, $X_{2[2:3]}$, $X_{3[2:3]}$) seçilir.

$m = 4$ olduğu durumda tek tekrarlı MSKÖ aşağıdaki gibi tanımlanır:

$$\begin{bmatrix} X_{1[1:4]} \leq \boldsymbol{X_{1[2:4]}} \leq X_{1[3:4]} \leq X_{1[4:4]} \\ X_{2[1:4]} \leq \boldsymbol{X_{2[2:4]}} \leq X_{2[3:4]} \leq X_{2[4:4]} \\ X_{3[1:4]} \leq X_{3[2:4]} \leq \boldsymbol{X_{3[3:4]}} \leq X_{3[4:4]} \\ X_{4[1:4]} \leq X_{4[2:4]} \leq \boldsymbol{X_{4[3:4]}} \leq X_{4[4:4]} \end{bmatrix}$$

Küme boyutu m çift sayı olduğu için, ilk iki kümeden en küçük sıralı ikinci birimler ($X_{1[2:4]}$, $X_{2[2:4]}$); diğer iki kümeden en küçük sıralı üçüncü birimler ($X_{3[3:4]}$, $X_{4[3:4]}$) seçilir.

Küme boyutu tek sayı ise, MSKÖ'nün kitle ortalaması ve varyans tahmini aşağıdaki gibidir:

$$\bar{X}_{MSK\ddot{O}} = \frac{1}{m}\sum_{i=1}^{m} X_{i((m+1)/2)} \tag{2.7}$$

$$V(\bar{X}_{MSK\ddot{O}}) = \frac{1}{m^2}\sum_{i=1}^{m} V\left(X_{i((m+1)/2)}\right) \tag{2.8}$$

Küme boyutu çift sayı olduğunda, MSKÖ'nün kitle ortalaması ve varyans tahmini aşağıdaki gibi tanımlanır:

$$\bar{X}_{MSK\ddot{O}} = \frac{1}{m}\left(\sum_{1}^{m/2} X_{i(m/2)} + \sum_{i=(m/2)+1}^{m} X_{i((m+2)/2)}\right) \tag{2.9}$$

$$V(\bar{X}_{MSK\ddot{O}}) = \frac{1}{m^2}\left(\sum_{i=1}^{m/2} V\left(X_{i(m/2)}\right) + \sum_{i=(m/2)+1}^{m} V\left(X_{i((m+2)/2)}\right)\right) \tag{2.10}$$

2.4 Yüzdelik Sıralı Küme Örneklemesi

YSKÖ, kitle ortalamasını tahmin etmek için Muttlak (2003) tarafından önerilmiştir. YSKÖ'nün çalışma prensibi aşağıdaki gibi tanımlanır:

Adım 1: Kitleden rastgele olarak m boyutunda m tane küme seçilir.

Adım 2: Her kümedeki birimler, maliyeti düşük olan bir yöntemle ilgilenilen değişkene göre sıralanır.

Adım 3: Küme boyutu m tek sayı ise; ilk $(m-1)/2$ setlerinden $\left(p(m+1)\right)$. en küçük sıralı birimler, diğer $(m-1)/2$ setlerinden $\left((1-p)(m+1)\right)$. en küçük sıralı birimler seçilir. m çift sayı ise; ilk $m/2$ setlerinden $\left(p(m+1)\right)$. en küçük sıralı birimler, diğer $m/2$ setlerinden $\left((1-p)(m+1)\right)$. en küçük sıralı birimler seçilir. Ortanca birim, kalan son kümenin m. birimi olarak seçilir. p, yüzdelik değer olarak tanımlanır ve 0 ile 1 arasında değer alır. $\left(p(m+1)\right)$ ve $\left((1-p)(m+1)\right)$ ifadeleri en yakın tamsayıya yuvarlanır.

Adım 4: İşlemler mr birim seçilene kadar r kez tekrarlanır.

$m = 5$ ve $p = 0{,}4$ iken tek tekrarlı YSKÖ aşağıdaki gibi gösterilir:

$$\begin{bmatrix} X_{1[1:5]} \le \boldsymbol{X_{1[2:5]}} \le X_{1[3:5]} \le X_{1[4:5]} \le X_{1[5:5]} \\ X_{2[1:5]} \le \boldsymbol{X_{2[2:5]}} \le X_{2[3:5]} \le X_{2[4:5]} \le X_{2[5:5]} \\ X_{3[1:5]} \le X_{3[2:5]} \le X_{3[3:5]} \le \boldsymbol{X_{3[4:5]}} \le X_{3[5:5]} \\ X_{4[1:5]} \le X_{4[2:5]} \le X_{4[3:5]} \le \boldsymbol{X_{4[4:5]}} \le X_{4[5:5]} \\ X_{5[1:5]} \le X_{5[2:5]} \le \boldsymbol{X_{5[3:5]}} \le X_{5[4:5]} \le X_{5[5:5]} \end{bmatrix}$$

İlk iki kümeden en küçük sıralı ikinci birimler ($X_{1[2:5]}$, $X_{1[2:5]}$), üçüncü ve dördüncü kümelerden en küçük sıralı dördüncü birimler ($X_{3[4:5]}$, $X_{4[4:5]}$) seçilir. Son kümeden ise, ortanca birim ($X_{5[3:5]}$) seçilir.

$m = 4$ ve $p = 0{,}2$ iken tek tekrarlı YSKÖ aşağıdaki gibi gösterilir:

$$\begin{bmatrix} \boldsymbol{X_{1[1:4]}} \le X_{1[2:4]} \le X_{1[3:4]} \le X_{1[4:4]} \\ \boldsymbol{X_{2[1:4]}} \le X_{2[2:4]} \le X_{2[3:4]} \le X_{2[4:4]} \\ X_{3[1:4]} \le X_{3[2:4]} \le X_{3[3:4]} \le \boldsymbol{X_{3[4:4]}} \\ X_{4[1:4]} \le X_{4[2:4]} \le X_{4[3:4]} \le \boldsymbol{X_{4[4:4]}} \end{bmatrix}$$

İlk iki kümeden en küçük sıralı birinci birimler ($X_{1[1:4]}$, $X_{2[1:4]}$), son iki kümeden en büyük sıralı birimler ($X_{3[4:4]}$, $X_{4[4:4]}$) seçilir.

Küme sayısı m tek sayı ve $a = [p(m+1)]$, $b = [(1-p)(m+1)]$ olmak üzere, YSKÖ'nün kitle ortalaması ve varyans tahmini aşağıdaki gibi tanımlanır:

$$\bar{X}_{YSK\ddot{O}} = \frac{1}{m}\left(\sum_{i=1}^{(m-1)/2} X_{i[a:m]} + \sum_{i=(m-1/2)+1}^{m-1} X_{i[b:m]} + X_{i[(m+1/2):m]} \right) \tag{2.11}$$

$$V(\bar{X}_{YSK\ddot{O}}) = \frac{1}{m^2}\left(\sum_{i=1}^{(m-1)/2} V\left(X_{i[a:m]}\right) + \sum_{i=(m-1/2)+1}^{m-1} V\left(X_{i[b:m]}\right) + V\left(X_{i[(m+1/2):m]}\right) \right) \tag{2.12}$$

Küme sayı m çift sayı ve $a = [p(m+1)]$, $b = [(1-p)(m+1)]$ olmak üzere, YSKÖ'nün kitle ortalaması ve varyans tahmini aşağıdaki gibidir:

$$\bar{X}_{YSK\ddot{O}} = \frac{1}{m}\left(\sum_{i=1}^{m/2} X_{i[a:m]} + \sum_{i=(m/2)+1}^{m} X_{i[b:m]}\right) \tag{2.13}$$

$$V(\bar{X}_{YSK\ddot{O}}) = \frac{1}{m^2}\left(\sum_{i=1}^{m/2} V\left(X_{i[a:m]}\right) + \sum_{i=(m/2)+1}^{m} V\left(X_{i[b:m]}\right)\right) \tag{2.14}$$

BÖLÜM 3
MAKİNE ÖĞRENMESİ

Makine öğrenmesi, bilgisayara belli özellik ve davranışların öğretilerek verilere dayalı tahminler yapmak için çeşitli algoritma ve teknikler geliştiren yöntemler topluluğudur (Molnar, 2020). Bilgisayarların insan beyni gibi çalışması mantığına dayalı olarak geliştirilmiştir. İnsan beyni, yaşanan olaylardan çıkardığı sonuçlara göre, gelecekte karşılaşacağı olaylar karşısındaki davranışlarına ve kararlarına yön verir. Makine öğrenmesi algoritmaları da kendilerine gösterilen olayları inceleyerek, bu olayların nasıl meydana geldiklerini öğrenirler. Daha sonra, öğrendikleri bilgiler üzerinden genelleme yapma yeteneği kazanırlar (Erdem, 2014). Bu algoritmaların temel amacı, geçmiş verileri kullanarak yeni bilgiler elde etmek, kendilerini geliştirmek ve gelecekte karşılaşılacak problemlere çözüm sunan modeller oluşturmaktır (Çelik ve Altunaydın, 2018).

Makine öğrenmesinde, çözülecek problemin türüne, değişken sayısına, veri setinin yapısına bağlı olarak farklı öğrenme türleri kullanılır (Mahesh, 2018). Dört farklı öğrenme türü vardır. Bu çalışma kapsamında denetimli öğrenme yöntemleri incelenmiştir.

3.1 Öğrenme Türleri

Denetimli, denetimsiz, yarı denetimli ve pekiştirmeli olmak üzere dört farklı öğrenme türü vardır.

3.1.1 Denetimli Öğrenme

Denetimli öğrenme, mevcut girdi verilerini kullanarak çıktı verilerini tahmin etmeyi amaçlayan yöntemler topluluğudur. İki tür denetimli öğrenme vardır.

- Regresyon: Bağımsız değişkenlerin her bir gözlem değerine karşılık, bağımlı değişkende değerleri vardır. Bağımsız değişkenler ile bağımlı değişken

arasındaki ilişki incelenerek bir model oluşturulur ve tahmin yapılır. Bağımlı değişken, sürekli yapıdadır.

- Sınıflandırma: Gözlemlerin, bağımsız değişkenlerin özelliklerine göre, bağımlı değişkende belirlenen kategorilere ayrılmasıdır. Bağımlı değişken kategorik yapıdadır.

3.1.2 Denetimsiz Öğrenme

Denetimsiz öğrenmede, denetimli öğrenmeden farklı olarak, girdi değerleri için bir gözlem vektörü vardır ancak ilişkili bir yanıt değişkeni yoktur. Öğrenme süreci, değişkenler arası ilişkiler ya da gözlemler arasındaki ilişkiler incelenerek gerçekleşir.

3.1.3 Yarı Denetimli Öğrenme

Yarı denetimli öğrenme, veri setinde etiketli ve etiketsiz veriler olduğu durumlarda kullanılır. Etiketlenmemiş veriler hakkında bilgi sahibi olmak amaçlanır (Çelik ve Altunaydın, 2018). Bu durumlarda, makinenin nasıl öğreneceğini belirlemek için geliştirilen algoritmalardır.

3.1.4 Pekiştirmeli Öğrenme

Pekiştirmeli öğrenme, makinenin bir ödül sistemi kullanılarak eğitilmesi yöntemidir. Başlangıç noktası ve hedef noktası vardır. Makine bu iki nokta arasında belli tercihler yaparak ilerler. Eğer doğru yoldan ilerlerse pozitif ödül, yanlış yoldan ilerlerse negatif ödül alır. Öğrenme süreci iki nokta arasındaki yolda ilerlerken gerçekleşir (Bonaccorso, 2017).

3.2 Makine Öğrenmesi Süreci

Makine öğrenmesinde, denetimli öğrenme algoritmaları ilk önce eğitilir, daha sonra ne kadar iyi öğrendikleri test edilir. Mevcut verilerden alınan bilgiler ile model öğrenir

ve yeni verilerle karşılaşıldığında öğrendiklerini kullanarak sonuçlar üretir. Algoritmaların öğrenme süreci ve öğrenme konusunda ne kadar başarılı olduklarının test edilmesi için; veri seti, eğitim ve test seti olarak ayrılır. Eğitim seti, makine öğrenmesi algoritmalarının verileri tanıyarak eğitilmesi ve öğrendiklerine göre tahmin yapması için oluşturulan settir. Test seti ise, eğitilen algoritmanın ne kadar doğru çalıştığını test etmek için kullanılan settir (Çiftler, 2019). Veri setini bölme işlemi istenilen oranda yapılabilir. Örneğin; bir veri seti, %80 eğitim seti, %20 test seti ya da %75 eğitim seti, %25 test seti olarak bölünebilir.

Makine öğrenmesi modellerinde parametre ve hiperparametre kavramlarıyla karşılaşılır. Parametre, modelin veri setini inceleyerek gözlemler içinden çıkardığı katsayılardır. Hiperparametre ise, kullanıcı tarafından belirlenen ve farklı değerler alabilecek dış parametrelerdir ve modeli optimize etmek için kullanılırlar. Kısacası, parametre değerleri model tarafından belirlenirken, hiperparametre değerleri kullanıcı tarafından modelin optimize edilmesi için belirlenir.

Makine öğrenmesinde önemli noktalardan biri, algoritmanın öğrendiği bilgileri genelleme yeteneğine sahip olmasıdır. Genelleme yeteneği, algoritmanın sadece eğitim setinden öğrendiği veriler üzerinde başarılı çalışması değil, daha önce görmediği verilerle de uyum içerisinde çalışmasıdır (Taş, 2020). Bazı durumlarda makine, eğitim verisini çok iyi öğrendiği için test verisi üzerinde başarılı bir genelleme yapamayabilir. Test setinden elde edilen hata değeri, eğitim seti kullanılarak oluşturulan modelin hata değerinden büyük olur (Yeom, Giacomelli, Fredrikson ve Jha, 2018). Bu durum, aşırı öğrenme olarak tanımlanır. Aşırı öğrenmenin önüne geçebilmek için k-katlı çapraz doğrulama yöntemi kullanılır.

3.3 K-katlı Çapraz Doğrulama

K-katlı çapraz doğrulama yöntemi, oluşturulan modellerde karşılaşılabilecek aşırı öğrenme sorununa çözüm sunmakla birlikte, en iyi modelin seçilmesinde de kullanılır. İlk olarak, veri seti eğitim ve test seti olarak ayrılır. Daha sonra, eğitim seti kendi içerisinde k tane eşit parçaya bölünür (Anguita, Ghio, Ridella ve Sterpi, 2009).

Yapılan çalışmalar sonucunda optimum k değerinin 10 olduğu gösterilmiştir. Ancak, az sayıda gözlem değerine sahip veri setlerinde bu değer 2 ya da 5 olarak alınabilir (Erpolat ve Öz, 2010). k parçaya ayrılan eğitim setinde, parçalardan bir tanesi dışarıda bırakılarak, kalan $(k-1)$ tane parçayla model oluşturulur. Oluşturulan model, dışarıda bırakılan parça ile test edilir. Böylece, eğitim setinin içinde ayrı bir doğrulama işlemi gerçekleşir. Bu işlem bütün parçalara uygulandıktan sonra, eğitim seti ile oluşturulmuş bir model elde edilir (Filiz, 2019). Elde edilen modelin test seti ile sınanması sonucunda bir hata değeri elde edilir. Bu hata değeri üzerinden, modelin başarısı yorumlanabilir. Model performansını yorumlamak için, bazı başarı değerlendirme ölçütleri kullanılır.

3.4 Model Başarı Değerlendirme Ölçütleri

Eğitim seti ile eğitilen modelin başarısı, test seti kullanılarak sınanır. Bu işlem sonucunda modelin tahmin başarısını değerlendirmek için bazı başarı değerlendirme ölçütleri kullanılır. Regresyon ve sınıflandırma yöntemleri için farklı değerlendirme ölçütleri kullanılmaktadır.

3.4.1 Regresyon Modellerinin Başarı Değerlendirme Ölçütleri

- Hata Kareler Ortalaması (HKO): Gözlem değerleri ile tahmin değerleri arasındaki farkın karelerinin ortalamasını ölçer. Modelin HKO değerinin sıfıra yakın olması modelin başarılı olduğunu gösterir.

$$HKO = \frac{1}{n}\sum_{i=1}^{n} e_i^2 \tag{3.1}$$

- Hata Kareler Ortalamasının Karekökü (HKOK): Gözlem değerleri ile tahmin değerleri arasındaki hata oranının belirlenmesi için kullanılır. Tahmin hatalarının standart sapmasıdır. Modelin HKOK değerinin sıfıra yakın olması model başarısının yüksek olduğunu gösterir.

$$HKOK = \sqrt{\frac{\sum_{i=1}^{n} e_i^2}{n}} \tag{3.2}$$

- Ortalama Mutlak Hata (OMH): Mutlak hataların aritmetik ortalamasıdır. Modelin OMH değerinin sıfıra yaklaşması modelin başarılı olduğunu gösterir.

$$OMH = \frac{1}{n}\sum_{i=1}^{n} |e_i| \tag{3.3}$$

Bu tez çalışmasında, regresyon modellerinde başarı değerlendirme ölçütü olarak HKOK değeri kullanılmıştır.

3.4.2 Sınıflandırma Modellerinin Başarı Değerlendirme Ölçütleri

- Karmaşıklık Matrisi: Gözlem değerleri ile tahmin değerlerinin karşılaştırılarak, sınıflandırma modelinin performansının değerlendirildiği matristir. Bu matris kullanılarak doğruluk, hata oranı hesaplanabilir.

Tablo 3.1 Karmaşıklık matrisi

	Tahmin Değeri		
Gerçek Değer		Pozitif	Negatif
	Pozitif	Doğru Pozitif (DP)	Yanlış Negatif (YN)
	Negatif	Yanlış Pozitif (YP)	Doğru Negatif (DN)

Doğruluk; modelin doğru sınıflandırma yapma sıklığını gösterir.

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

Hata oranı; modelin yanlış sınıflandırma yapma sıklığını ifade eder.

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

- Roc Eğrisi: Karmaşıklık matrisindeki, doğru pozitif ve yanlış pozitif oranları kullanılarak oluşturulur. X ekseninde yanlış pozitif oranı, Y ekseninde doğru pozitif oranları vardır. Bu oranlar göz önüne alınarak grafik üzerinde bir eğri oluşturulur. Bu eğrinin altında kalan alan ne kadar büyük olursa, model o kadar başarılıdır (Çolak, 2021).

Bu tez çalışmasında; sınıflandırma modellerinin başarı değerlendirme ölçütü olarak doğru sınıflandırma oranı (doğruluk) kullanılmıştır.

BÖLÜM 4
MAKİNE ÖĞRENMESİ YÖNTEMLERİ

4.1 Çoklu Doğrusal Regresyon

Regresyon analizi, değişkenler arasındaki ilişkiyi anlamak ve modellemek için kullanılan istatistiksel bir yöntemdir (Montgomery, Peck ve Vining, 2013). Elde edilen modeller ile geçmişte gerçekleşen olaylara dayanarak geleceğe yönelik tahminler yapılabilir (Demirhan ve Hamurkaroğlu, 2015). Mühendislik, ekonomi, fizik, kimya, biyoloji bilimleri ve sosyal bilimler dahil olmak üzere birçok alanda kullanılır.

Regresyon ifadesi ilk defa, İngiliz antropolog Galton (1886) tarafından kullanmıştır. Galton (1886) yaptığı çalışmada, çocukların boylarının ortalamasının, ebeveynlerin boylarıyla ilişkisini araştırmıştır. İlerleyen yıllarda, regresyon analizi farklı bilimsel alanlara uygulanarak geliştirilmiştir.

Regresyon modellerinde, bağımlı ve bağımsız değişken olarak adlandırılan iki temel kavram vardır. Bağımlı değişken, aldığı değerlerdeki değişimin, bir ya da daha çok sayıda bağımsız değişken tarafından açıklanabildiği yanıt değişkenidir. Bağımsız değişken, yanıt değişkenin değerlerindeki değişim üzerinde etkili olan değişkendir (Demirhan ve Hamurkaroğlu, 2015).

Çoklu doğrusal regresyon (ÇDR), bağımlı değişkenin birden fazla bağımsız değişken ile açıklanmaya çalışıldığı regresyon analizidir. Bağımsız değişken sayısının fazla olması, bağımlı değişken hakkında daha çok bilgi sahibi olmayı sağlar.

ÇDR, matris işlemleri ile çözümlenmektedir. Y bağımlı değişkeni ve X_1, …, X_k bağımsız değişkenlerinden oluşan ÇDR denklemi aşağıdaki gibi tanımlanır:

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_k X_k + \varepsilon \tag{4.1}$$

Eşitlik (4.1)'de; $\beta_j,\ j = 1, \ldots, k$ regresyon katsayıları, ε ise hata terimidir. Bu eşitliğin matris gösterimi aşağıdaki gibidir:

$$\boldsymbol{Y} = \boldsymbol{X\beta} + \ \boldsymbol{\varepsilon} \tag{4.2}$$

Eşitlik (4.2)'de Y; $(n \times 1)$ boyutlu yanıt değişkeni vektörü, X; $n \times (k+1)$ boyutlu bağımsız değişken matrisi, β; $(k+1) \times 1$ boyutlu regresyon katsayıları vektörü ve ε; $(n \times 1)$ boyutlu hata terimleri vektörüdür.

Regresyon denklemi kullanılarak i. gözleme ait bağımlı değişken değerini tahmin etmek için,

$$\hat{y}_i = \hat{\beta}_0 + \ \hat{\beta}_1 x_{1i} + \cdots + \ \hat{\beta}_k x_{ki} \tag{4.3}$$

eşitliği kullanılır. Bu eşitlik,

$$\hat{y}_i = \ \boldsymbol{x_i}'\widehat{\boldsymbol{\beta}} \tag{4.4}$$

biçiminde yazılabilir. Eşitlik (4.4)'teki matris ve vektörler,

$$\hat{y}_i = [1 \quad x_{1i} \quad x_{2i} \ \cdots \ x_{ki}]_{\ 1\times(k+1)} \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \vdots \\ \hat{\beta}_k \end{bmatrix}_{(k+1)\times 1} \tag{4.5}$$

biçiminde gösterilebilir.

4.1.1 Regresyon katsayılarının tahmin edilmesi

ÇDR katsayıları genellikle en küçük kareler (EKK) yöntemiyle tahmin edilir. EKK yönteminde, hata terimlerinin tahmini olan $e_i = y_i - \hat{y}_i$ artıklarının karelerinin

toplamı minimize edilmek istenir. Artıkların kareler toplamı Eşitlik (4.6)'daki gibi yazılabilir.

$$\sum_{i=i}^{n} e_i^2 = \sum_{i=i}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{k} \beta_j x_{ji} \right)^2 \tag{4.6}$$

β'nın EKK yöntemi ile bulunan tahminleri Eşitlik (4.7)'deki gibidir.

$$\widehat{\boldsymbol{\beta}} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{Y} \tag{4.7}$$

EKK tahmin edicisi olan $\hat{\beta}$, β'nın en iyi, doğrusal ve yansız kestiricisidir. $\hat{\beta}$'nın beklenen değer ve varyansı, Eşitlik (4.8) ve Eşitlik (4.9)'da verilmiştir.

$$E(\widehat{\boldsymbol{\beta}}) = E[(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{Y}] = E[(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'(\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon})] = \boldsymbol{\beta} \tag{4.8}$$

$$V(\widehat{\boldsymbol{\beta}}) = E\left[(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta})(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta})'\right] = \sigma^2(\boldsymbol{X}'\boldsymbol{X})^{-1} \tag{4.9}$$

Bir ÇDR modelinin geçerli olması ve kullanılabilmesi için aşağıdaki varsayımların sağlanması gerekir (Montgomery vd., 2013).

- Seçilen örneklem kitleyi temsil edebilmelidir.
- Hatalar kendi aralarında bağımlı olmamalıdır.
- Hata varyansı sabit ve değişmez olmalıdır.
- Tüm bağımsız değişkenler kendi aralarında ilişkisiz olmalıdır.
- Hatalar sıfır ortalama ve sabit varyans ile normal dağılmalıdır.

Hataların kendi aralarında ilişkili olması "otokorelasyon", hataların varyanslarının eşit olmaması "değişen varyanslılık", bağımsız değişkenlerin ilişkili olması "çoklu bağlantı" sorununa neden olmaktadır (Rawlings, Pantula ve Dickey, 2001).

Regresyon modellerinde karşılaşılan çoklu bağlantı sorununa çözüm olması için alternatif yöntemler geliştirilmiştir. Bu tezde; ilgili yöntemlerden, temel bileşenler regresyonu, kısmi en küçük kareler regresyonu, ridge regresyon, lasso regresyon ve elastik net regresyon ele alınmıştır.

4.2 Temel Bileşenler Regresyonu

ÇDR yönteminde, modelin geçerli olması için gereken kriterlerden birisi, bağımsız değişkenlerde çoklu bağlantı olmamasıdır. Ancak, günlük hayat verilerinin çoğunda çoklu bağlantı sorunu ile karşılaşılır. Regresyon katsayılarının en iyi, doğrusal ve yansız tahmini EKK tahmin edicileri ile elde edilir. Ancak çoklu bağlantı durumunda, EKK tahmin edicileri doğru tahminler vermeyebilir. Bu sebepten, daha düşük tahmin varyansı elde etmek için yanlı tahmin yöntemleri tercih edilebilir (Rawlings vd., 2001).

Temel bileşenler regresyonundan (TBR), ilk olarak Kendall (1957) kitabında bahsetmiştir. Bağımsız değişkenlerin ilişkili olması durumunda temel bileşenlere regresyon uygulanmasını önermiştir. TBR kullanımının artmasıyla, hangi temel bileşenlerin regresyon modelinde kalacağını seçme konusunda farklı yaklaşımlar ortaya çıkmıştır. Yaklaşımlardan biri, temel bileşenlerin bağımlı değişken ile ilişkilerini hesaplayıp modelde kalıp kalmayacağına karar vermektir. Diğer bir yaklaşım ise, büyük varyansa sahip bileşenleri modelde tutup küçük varyansa sahip olanları modelden çıkarmaktır (Jolliffe, 1982). Mansfield, Webster ve Gunst (1977), küçük varyansa sahip bileşenlerin yok edilmesi durumunda regresyon tahminlerindeki bilgi kaybının az olacağını belirtmişlerdir. Gunst ve Mason (1980) kitaplarında, temel bileşen seçiminin sadece bileşenlerin varyanslarına bağlı olduğunu iddia etmişlerdir. Aynı şekilde, Mosteller ve Tukey (1977) de düşük varyanslı bileşenlerin regresyon modelinde önemsiz olacaklarını savunmuşlardır. Jeffers (1967), küçük varyansa sahip bileşenlerin bağımlı değişkenle ilişkili olabileceğini, bu yüzden bağımlı değişken ile bütün bileşenler arasındaki ilişkinin incelenmesi gerektiğini savunmuştur. Hawkins (1973), TBR'yi bütün değişkenlere temel bileşenler analizi uygulayıp, küçük varyanslı temel bileşenlerin seçilmesi olarak tanımlamıştır. Ayrıca, düşük varyanslı bileşenlerin önemsiz olacağının net olarak söylenemeyeceğini savunmuştur. Hill, Fomby ve

Johnson (1977), yaptıkları çalışmada sadece varyanslarına bakılarak bileşenlerin belirlenemeyeceğini göstermişlerdir.

TBR, veride çoklu bağlantı sorunu olduğunda kullanılan yanlı tahmin yöntemlerinden birisidir. TBR'de, orijinal değişkenler yerine bunların dik dönüşümleri alınarak yeni bir değişken kümesi elde edilir. Bu değişken kümesine EKK uygulanıp, regresyon katsayı tahminleri yapılır. TBR, standartlaştırılmış bağımsız değişkenler matrisine temel bileşenler analizi uygulanması olarak tanımlanabilir (Bursa, 2019).

TBR analizine başlamadan önce veri standartlaştırılır. Bu işlem, bağımlı ve bağımsız değişkenlerin ortalamalarından farkları alınarak standart sapmalarına bölünmesiyle gerçekleştirilir (Tunç, 2018).

Bağımsız değişkenlerin temel bileşenlere dönüşümü aşağıdaki gibi yapılmaktadır.

$$\boldsymbol{X'X = PDP' = W'W} \tag{4.10}$$

P; $X'X$ in öz vektör matrisini, D; $X'X$ in öz değerlerinin matrisini, W ise; temel bileşenler matrisini ifade etmektedir.

TBR'nin asıl amacı, çoklu bağlantıya neden olan bileşenleri yok etmektir. Bunu yaparken, açıklayıcı değişkenlerin korelasyon matrisinden elde edilen öz değerler kullanılır. Küçük öz değerlerle bağlantılı olan bileşenler analizden çıkartılır (Bursa, 2019). Böylece çoklu bağlantı sorunu ortadan kalkar. Daha sonra, kalan bileşenlere regresyon analizi uygulanır. Regresyon katsayıları Eşitlik (4.11)'deki gibi tahmin edilir.

$$\boldsymbol{\gamma = (W'W)^{-1}W'Y} \tag{4.11}$$

TBR analizi sonrasında elde edilen temel bileşenlerin regresyon katsayılarından orijinal değişkenlerin katsayılarına geçiş yapabilmek için aşağıdaki eşitlikler kullanılabilir.

$$\boldsymbol{\gamma} = \boldsymbol{P}'\widehat{\boldsymbol{\beta}} \tag{4.12}$$

$$\widehat{\boldsymbol{\beta}} = \boldsymbol{P}\boldsymbol{\gamma} \tag{4.13}$$

4.3 Kısmi En Küçük Kareler Regresyonu

Kısmi en küçük kareler regresyonu (KEKKR); bağımsız değişkenlerin sayıca çok ve ilişkili olması durumunda, ÇDR analizine alternatif bir yöntem olarak Wold (1975) tarafından önerilmiştir. İlerleyen yıllarda, kimyasal uygulamalarda elde edilen verileri analiz etmek amacıyla kullanılmıştır (Geladi ve Kowalski, 1986; Mevik ve Wehrens, 2007). Günümüzde ise, ÇDR uygulamalarında çoklu bağlantı olması durumunda, daha doğru ve daha az bileşenli model tahmini elde etmek için sıklıkla kullanılmaktadır (Polat, 2009).

KEKKR, bir regresyon modelinde çoklu bağlantı sorunu ile karşılaşıldığında kullanılan yanlı tahmin yöntemlerinden biridir. Aralarında ilişki olan bağımsız değişkenler, aralarında ilişki olmayan daha az sayıda bileşenlere indirgenir. TBR analizi ile benzer şekilde çalışmaktadır. TBR'de sadece bağımsız değişkenler üzerinden işlemler yapılırken, KEKKR'de hem bağımlı hem de bağımsız değişkenler kullanılarak analiz yapılır (Bursa, 2019).

KEKKR'de; NIPALS, SIMPLS, UNIPAL, SAMPLS ve KERNEL gibi çeşitli algoritmalar kullanılmaktadır (Lindgren ve Rännar, 1998). Klasik NIPALS algoritması, KEKKR analizinin temelini oluşturur (Bulut ve Alın, 2009). Bu algoritma, bağımlı ve bağımsız değişkenler arasındaki kovaryansı maksimize eden gizli bileşenleri bulmayı amaçlar ve bu işlemi yinelemeli olarak yapar. Algoritmanın her adımında bir bileşen ve bu bileşene ait ağırlık değerleri bulunur. Bağımlı ve bağımsız değişkenlerin oluşturduğu değişken blokları arasındaki ilişki EKK skor

vektörleri yardımıyla modellenir (Bursa, 2019). Ayrıştırılan değişken blokları Eşitlik (4.14) ve Eşitlik (4.15)'teki gibi gösterilir.

$$\boldsymbol{X'} = \boldsymbol{TP'} + \boldsymbol{E} \tag{4.14}$$

$$\boldsymbol{Y'} = \boldsymbol{UQ'} + \boldsymbol{F} \tag{4.15}$$

X; $(p \times n)$ boyutlu bağımsız değişkenler matrisini, Y; $(p \times m)$ boyutlu bağımsız değişkenler matrisini, T ve U; $(n \times p)$ boyutlu skor matrislerini, P; $(n \times p)$, Q; $(m \times p)$ boyutlu yükler matrislerini, E; $(n \times p)$, F; $(m \times p)$ boyutlu artık matrislerini gösterir. Kısmi en küçük kareler yöntemi ile t ve u skor matrislerinin arasındaki kovaryansı maksimum yapan, aşağıdaki eşitlikte görülen w ve c ağırlıkları bulunur (Bulut, 2011).

$$[cov(\boldsymbol{t}, \boldsymbol{u})]^2 = [cov(\boldsymbol{Xw}, \boldsymbol{Yc})]^2 \tag{4.16}$$

Elde edilmek istenen bileşen sayısına ulaşıldığında veya bağımlı değişkendeki değişimin yeterli kısmı açıklandığında algoritma sonlanır (Bulut ve Alın, 2009; Bulut, 2011).

KEKKR'de en uygun bileşen sayısına karar verilirken genellikle en küçük HKOK değerini veren bileşen sayısı ya da çapraz doğrulama yöntemi kullanılır (Bulut, 2011).

4.4 Ridge Regresyon

Ridge regresyon (RR), çoklu bağlantı durumunda kullanılan diğer bir yanlı tahmin yöntemi olarak Hoerl (1962) tarafından önerilmiştir. Daha sonra, Hoerl ve Kennard (1970) yaptıkları çalışmada; çoklu bağlantı durumunda EKK tahminlerinin yetersiz olduğunu göstererek, RR ile daha düşük hata değerlerine sahip yanlı tahminler elde edilebileceğini göstermişlerdir. Yanlı bir yöntem olmasına rağmen, tahmin varyanslarını düşürdüğü için tercih edilen bir yöntemdir (Büyükuysal, 2010).

RR yönteminde, bağımsız değişkenler korelasyon matrisinin köşegen elemanlarına küçük bir pozitif sayı (λ_R) eklenir. λ_R, yanlılık ya da ceza parametresi olarak bilinir. Bu durumda, β'nın yanlı tahmin edicisi olan RR tahmin edicisi Eşitlik (4.17)'deki gibidir.

$$\widehat{\boldsymbol{\beta}}_{\boldsymbol{RR}} = (\boldsymbol{X}'\boldsymbol{X} + \lambda_R \boldsymbol{I})^{-1}\boldsymbol{X}'\boldsymbol{Y} \tag{4.17}$$

RR'nin amacı, hata kareler toplamını minimize eden katsayıları bulmaktır. Buna göre RR'nin amaç fonksiyonu Eşitlik (4.18)'deki gibidir.

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 + \lambda_R \sum_{j=1}^{p} {\beta_j}^2 \tag{4.18}$$

RR, EKK yöntemi ile benzer şekilde çalışır. Eşitlik (4.18)'de, $\lambda_R = 0$ olarak alındığında iki yöntem birbirine eşit duruma gelir. Bu sebeple, RR tahmini EKK tahmininin doğrusal dönüşümü olarak tanımlanabilir (Sakallıoğlu ve Kaçıranlar, 2008). $\lambda_R > 0$ olduğunda, RR tahmininin varyansı EKK tahmininin varyansından küçüktür.

$$V\left(\widehat{\boldsymbol{\beta}}_{\boldsymbol{RR}}\right) = (\boldsymbol{X}'\boldsymbol{X} + \lambda_R \boldsymbol{I})^{-1}\boldsymbol{X}'\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X}+\lambda_R \boldsymbol{I})^{-1}\sigma^2 < V\left(\widehat{\boldsymbol{\beta}}\right) = \sigma^2(\boldsymbol{X}'\boldsymbol{X})^{-1} \tag{4.19}$$

İdeal λ_R değerine karar vermek için, genellikle RR'nin grafiksel gösterimi olan ridge izi kullanılmaktadır. Bu grafikte; regresyon katsayıları ($\hat{\beta}_{RR}$) düşey eksende, λ_R değerleri yatay eksende gösterilir. λ_R'a karşılık gelen her bir katsayı değeri için bir eğri veya iz oluşur. Bağımsız değişkenler arasında güçlü ilişki olması durumunda, λ_R'ın küçük değerleri için katsayılar hızlı bir şekilde değişim gösterecektir. λ_R'nın büyük değerleri için ise, katsayılar yavaş yavaş değişecektir (Büyükuysal, 2010). λ_R'daki artışlara karşılık, regresyon katsayılarındaki değişimin azalarak yatay eksene paralel şekilde konumlandığı noktadaki değer ideal λ_R değeridir (Hoerl ve Kennard, 1970).

RR yönteminde, bütün değişkenler regresyon modeline alınır. Analiz sonucu önemsiz bulunan değişkenlerin katsayıları sıfıra yaklaştırılır.

Makine öğrenmesinde, RR parametresine karar vermek için çapraz doğrulama yöntemi kullanılır. İlk önce, λ_R değerlerini içeren bir küme belirlenir. Daha sonra, her bir λ_R değeri için çapraz doğrulama yöntemi kullanılarak hata değerleri hesaplanır. En küçük hata değerinin elde edildiği λ_R değeri, RR parametresi olarak seçilir.

4.5 Lasso Regresyon

Regresyon analizinde çoklu bağlantı durumuyla karşılaşıldığında, tahmin doğruluğu için daralma yapılabilir ya da önemsiz değişkenler 0'a ayarlanabilir. RR tahmin edicisi katsayıları daraltır, ancak önemsiz değişkenleri 0'a ayarlamadığı için model yorumlama sorunuyla karşılaşılır (Küçük, 2019). Bu soruna çözüm olarak, Tibshirani (1996) tarafından lasso regresyon (LR) geliştirilmiştir. Tibshirani (1996), LR'ı Eşitlik (4.20)'deki gibi tanımlamıştır.

$$\hat{\beta}_{Lasso} = argmin \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 \right\} \tag{4.20}$$

LR yönteminin kullanım avantajları vardır. Bazı değişkenlerin katsayılarını sıfıra yaklaştırarak, bazılarını da sıfıra indirerek model parametrelerini ayarlar. Katsayıların küçültülmesi ya da yok edilmesi tahmin varyansının azalmasını sağlar (Fonti ve Belitser, 2017). Bağımlı değişken ile ilişkisi olmayan değişkenleri ortadan kaldırarak modelin yorumlanmasını kolaylaştırır. Özellikle büyük veri setlerinde ve değişken sayısının gözlem sayısından fazla olduğu durumlarda, tahmin doğruluğu yüksek modeller elde edilmesini sağlar. Ayrıca düzenleme ve değişken seçimini eş zamanlı olarak yaptığı için tercih edilen bir yöntemdir (Yaman ve Cengiz, 2021).

RR'de olduğu gibi, LR yönteminin de temel amacı; hata kareler toplamını minimize etmektir. Bu durumda, LR'nin amaç fonksiyonu Eşitlik (4.21)'deki gibi tanımlanır.

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}x_{ij}\beta_j\right)^2+\lambda_L\sum_{j=1}^{p}|\beta_j| \tag{4.21}$$

λ_L, ayarlama parametresidir. Değişken katsayılarını sıfırlamanın ya da küçültmenin kontrolü λ_L ile sağlanır. λ_L yeterince büyük olduğunda katsayılar sıfıra eşitlenir ya da sıfıra yaklaştırılır. Bu şekilde değişken seçimi yapılmış olur. $\lambda_L = 0$ olarak alındığında LR, EKK yöntemi ile aynı şekilde çalışır (Fonti ve Belitser, 2017).

Diğer regresyon tahmini yöntemleriyle karşılaştırıldığında, parametre tahmini ve değişken seçimini eş zamanlı olarak yapması LR yönteminin en önemli avantajıdır (Küçük, 2019).

Makine öğrenmesinde, LR parametresi olan λ_L'in değerine karar vermek için çapraz doğrulama yöntemi kullanılır. λ_L değerlerini içeren bir küme seçilir. Her bir değer için çapraz doğrulama yöntemi kullanılarak hata değerleri hesaplanır. En düşük hata değerinin elde edildiği λ_L değeri, LR parametresi olarak belirlenir.

4.6 Elastik Net Regresyon

RR ve LR yöntemlerinin bazı dezavantajları vardır. Değişken sayısının gözlem sayısından büyük olduğu durumlarda, LR en fazla n tane değişken seçebilir. Bu durum LR yöntemi için sınırlayıcı bir özelliktir. LR, aralarında yüksek ilişki olan bağımsız değişkenler varlığında değişkenlerden birini seçer, hangisinin seçildiği ile ilgilenmez. Gözlem sayısının değişken sayısından büyük olduğu durumlarda, eğer tahminler arasında yüksek ilişki varsa, LR'ın tahmin performansı RR yönteminden daha güçlüdür (Zou ve Hastie, 2005). Elastik net regresyon (ENR); RR ve LR yöntemlerinde karşılaşılan dezavantajları gidermek için, iki yöntemin güçlü yönlerini birleştirerek Zou ve Hastie (2005) tarafından geliştirilmiştir. ENR yöntemi, değişken seçimini LR gibi yaparken, ilişkili değişkenlerin katsayılarını birbirlerine yaklaştırma işleminde RR gibi çalışır (Hastie, Tibshirani ve Friedman, 2009).

RR ve LR yöntemlerini birleştiren ENR Eşitlik (4.22)'deki gibi tanımlanır.

$$\hat{\beta}_{EN} = \left\{ argmin \left(\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda_R \sum_{j=1}^{p} {\beta_j}^2 + \lambda_L \sum_{i=1}^{p} |\beta_j| \right) \right\} \tag{4.22}$$

ENR'de optimum parametre değerlerini elde etmek önemlidir. Makine öğrenmesinde, λ_L ve λ_R parametrelerinin değerlerini belirlemek için, risk tahminini en aza indiren çapraz doğrulama yöntemi kullanılır (Zou ve Hastie, 2005). λ_L ve λ_R için değerler içeren bir küme belirlenir. Her bir λ_L ve λ_R değeri için çapraz doğrulama yöntemi ile hata değerleri elde edilir. En düşük hata değeri ile çalışan λ_L ve λ_R ENR parametreleri olarak belirlenir.

4.7 K-En Yakın Komşuluk Algoritması

K-en yakın komşuluk (KNN) algoritması, denetimli öğrenme yöntemlerinden biridir. Genellikle sınıflandırma problemlerinde kullanılmakla birlikte regresyon problemlerinde de kullanılır. Cover ve Hart (1967) tarafından, "En Yakın Komşu Örüntü Sınıflandırması" çalışmasıyla önerilmiştir. Bir gözlem değerinin hangi sınıfa ait olduğunu k en yakın komşu sayısına, mesafeye göre belirler (Cover ve Hart, 1967).

KNN algoritması, sınıflandırma ya da tahmin yapmak için en yakın komşu gözlemlerini kullanır. Sınıflandırma yaparken, kaç tane en yakın komşu ile çalışılacağı, k değeri ile belirlenir. $k = 1$ olarak alınırsa, yeni gelen gözlem en yakınındaki komşu ile aynı sınıfa tanımlanır (Köktürk, 2012).

En yakın komşular belirlenirken, yeni gözlem değeri ile eğitim setindeki gözlemler arasındaki uzaklıklar hesaplanır. Bulunan uzaklık değerleri küçükten büyüğe doğru

sıralanır. Eğitim setinden, yeni gözlem değerine en yakın olan k tane gözlem belirlenir. Çoğunlukta olan sınıf etiketine göre yeni gözlemin sınıfı belirlenir.

KNN ile regresyon tahmini yapılırken, sınıflandırma problemindeki işlemler takip edilerek k tane en yakın komşu değeri belirlenir. Yeni gözlem değerine en yakın olan komşu değerlerinin ortalaması alınır. Bu değer, yeni gözlem değerinin tahminidir.

Sınıflandırma yöntemlerinde model kendi içinde bir sınıflayıcı belirler ve yeni gözlemler için bu sınıflayıcıyı kullanır. KNN algoritmasında böyle bir sınıflayıcı yoktur. Yeni gözlem değerinin sınıflandırılması için her seferinde eğitim setindeki gözlemler kullanılır. Her yeni gözlemde bütün eğitim seti tekrar kullanıldığından sınıflama süreci uzun sürmektedir (Dilki ve Deniz Başar, 2020; Khan, Ding ve Perrizo, 2002).

KNN algoritmasında en önemli nokta en yakın komşu sayısına (k) karar vermektir. k değeri algoritmanın başında, veri setinin yapısına ve boyutuna bağlı olarak belirlenir. k değeri olması gerekenden küçük seçilirse, aralarında yüksek benzerlik olan gözlemler aynı sınıfa dahil edilir. Bu durum, oluşması gereken sınıflardan bazılarının yok sayılmasına neden olur. k değerinin büyük seçilmesi, aralarında benzerlik olmayan gözlemlerin aynı sınıfa dahil olmasına sebep olarak, doğru sınıflandırma oranını olumsuz yönde etkiler (Dilki ve Deniz Başar, 2020; Köktürk, 2012).

KNN algoritmasının avantajları arasında;

- Basit ve kolay anlaşılır olması,
- Eğitim setinin gürültülü verilerden az etkilenmesi,
- Büyük eğitim setlerinde etkili olması sayılabilir.

Dezavantajları arasında;

- Hesaplamanın karmaşık olması,
- Yüksek bellek alanına ihtiyaç duyması,
- k parametresine ve kullanılan uzaklık ölçüsüne bağlı olarak performansının etkilenmesi,

- Yavaş çalışan bir algoritma olması sayılabilir.

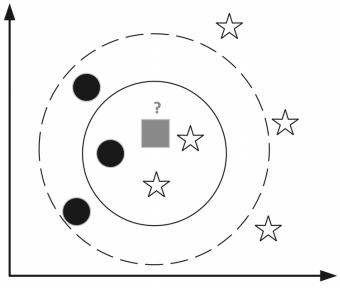
4.7.1 KNN Algoritmasının Çalışma Prensibi

x_q sınıflandırılacak gözlem değerini, $x_1, \dots, x_k$ en yakın komşu değerlerini (k) temsil etsin. x_q gözleminin sınıfını belirlemek için Eşitlik (4.23)'teki eşitlik kullanılır.

$$\hat{f}(x_q) = argmax_{v \in V} \sum_{i=1}^{k} \delta\big(v, f(x_i)\big) \tag{4.23}$$

a ve b eşit ise $\delta(a,b) = 1$ olarak, eşit olmadıkları takdirde $\delta(a,b) = 0$ olarak alınır (Batista ve Silva, 2009).

Yıldızlar ve dairelerin iki ayrı sınıfı temsil ettiği varsayılsın ve kare sınıflandırılacak olan gözlem olsun. $k = 3$ alınarak kareye en yakın 3 komşu incelenirse, 2 adet komşunun yıldız sınıfında, 1 adet komşunun daire sınıfında olduğu görülür. Ağırlıklı olan komşular yıldızlardır. Bu yüzden, kare gözlemi yıldızların sınıfına dahil edilir (Şekil 4.1).



Şekil 4.1 KNN sınıflandırması örneği (Z. Martinasek, Zeman, Malina, J. Martinasek 2015)

4.7.2 Uzaklık Ölçütleri

Farklı veri yapıları için farklı uzaklık ölçütleri önerilmiştir. Sayısal veriler için, Öklid, Manhattan, Minkowski, Chebyschev uzaklık ölçütleri kullanılır (Köktürk,

2012). Minkowski uzaklığı, sınıflandırma ve kümeleme gibi makine öğrenmesi yöntemlerinde sıklıkla kullanılan Öklid ve Manhattan uzaklıklarının genelleştirilmiş halidir. İki nokta arasındaki uzaklık Minkowski ölçütü ile Eşitlik (4.24)'teki gibi hesaplanır.

$$\left(\sum_{i=1}^{n}|x_i - y_i|^p\right)^{1/p} \tag{4.24}$$

Minkowski ölçütü; $p = 2$ olarak alındığında Öklid uzaklığını, $p = 1$ olduğunda Manhattan uzaklığını, n'in sonsuza gittiği durumda ise Chebyschev uzaklığını verir (Kresse ve Danko, 2012).

Öklid uzaklığı, makine öğrenmesi yöntemlerinde en sık kullanılan uzaklık ölçütüdür. İki nokta arasındaki doğrusal uzaklığı ifade eder ve Eşitlik (4.25)'teki gibi tanımlanır.

$$\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{4.25}$$

Manhattan uzaklığı, her iki nokta arasındaki farkların mutlak değerlerinin toplamı olarak tanımlanır. Eşitlik (4.26)'daki gibi hesaplanır.

$$\sum_{i=1}^{n}|x_i - y_i| \tag{4.26}$$

Chebyschev uzaklığı, n sonsuza giderken Minkowski uzaklığının özel bir halidir. Her iki nokta arasındaki farkların mutlak değerlerinin maksimumu olarak tanımlanır. Eşitlik (4.27)'deki gibi hesaplanır.

$$max_{i=1}^{n} = |x_i - y_i| \tag{4.27}$$

4.8 Destek Vektör Makineleri

Destek vektör makineleri (DVM), iki gruplu sınıflandırma problemleri için Cortes ve Vapnik (1995) tarafından geliştirilmiştir. Temeli istatistiksel öğrenme teorisine dayalı olan bir makine öğrenmesi yöntemidir. Diğer makine öğrenmesi yöntemlerine göre, doğrusal olmayan problemlerde daha etkili sonuçlar elde edilmesini sağlar (M. Acı, Avcı ve Ç. Acı, 2016). Genellikle sınıflandırma problemlerinde kullanılmakla birlikte regresyon problemlerinde de kullanılır.

DVM, teorisinin istatistiksel öğrenmeye dayalı olması sebebiyle güçlü bir yöntemdir. Genelleştirme yeteneği güçlüdür ve uygulamalarda yüksek performansla çalışır. Yetenekleri sayesinde; veri madenciliği, yüz tanımlama, finans, ekonomi, biyoloji gibi çeşitli alanlarda kullanılır (Arat, 2014).

4.8.1 Regresyon Problemlerinde Destek Vektör Makineleri

Regresyon problemlerinde kullanılan DVM temeline dayalı yöntem, destek vektör regresyonu (DVR) olarak adlandırılır. DVR, Smola ve Schölkoph (2004) tarafından önerilmiştir. Regresyon problemlerinin doğrusal ve doğrusal olmayan durumları vardır. DVR, esas olarak doğrusal olmayan regresyon problemlerine bir çözüm olarak geliştirilmiştir (Karal, 2018).

DVR yöntemini açıklayabilmek için, $x_i \in R^n$ gelecek vektörünü, $y_i \in R$ hedef çıktısını temsil eden $\{(x_i, y_i), \dots, (x_l, y_l)\}$ eğitim seti kullanılsın. Eğitim setini doğrusal olarak ayrılabilir duruma getirmek için, doğrusal olmayan haritalama fonksiyonu kullanılır. Bu fonksiyon yardımıyla eğitim seti daha yüksek boyutlu bir uzaya taşınarak doğrusal olarak ayrılabilir duruma gelir (M. Acı vd., 2016). Bu şekilde DVR Eşitlik (4.28)'deki gibi tanımlanır.

$$f(x) = \boldsymbol{w}^T \varphi(\boldsymbol{x}) + b \tag{4.28}$$

Bu denklemde, $f(x)$; tahmin değerlerini, w; model parametre vektörünü, φ; doğrusal olmayan haritalama fonksiyonunu, b sapma miktarını gösterir.

DVR'de amaç, tahmin hatasını minimize etmektir. Bu sebeple, eğitim setine doğru yaklaşan bir fonksiyon bulmaya çalışılır. DVR'nin amaç fonksiyonu Eşitlik (4.29)'daki gibidir.

$$min_{w,b,\xi,\xi^*}\left(\frac{1}{2}\right)\boldsymbol{w}^T\boldsymbol{w} + C\sum_{i=1}^{l}(\xi_i + \xi_i^*) \tag{4.29}$$

C; ceza, karmaşıklık parametresi ya da eşik değeri olarak adlandırılır. $\pm\varepsilon$; kabul edilebilir hata miktarını, ξ_i, ξ_i^*; $\pm\varepsilon$ değerlerinin üzerine eklenecek değerleri gösterir.

$$\begin{Bmatrix} \boldsymbol{w}^T\varphi(\boldsymbol{x_i}) + b - y_i \leq \varepsilon + \xi_i \\ y_i - \boldsymbol{w}^T\varphi(\boldsymbol{x_i}) - b \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0, i = 1,2,\dots,l \end{Bmatrix} \tag{4.30}$$

Eşitlik (4.30)'daki kısıtlamalara göre, ikinci dereceden optimizasyon problemi Lagrange çarpanları ile çözülürse model parametre vektörü Eşitlik (4.31)'deki gibi elde edilir.

$$\boldsymbol{w} = \sum_{i=1}^{l}(\lambda_i^* - \lambda_i)\,\varphi(\boldsymbol{x_i}) \tag{4.31}$$

DVR'de eğitim setini daha yüksek boyutlu bir uzaya taşımaya gerek kalmadan çekirdek fonksiyonları kullanılarak işlemler yapılabilir (Karal, 2018). Bu fonksiyonlar literatürde "*kernel trick"* fonksiyonları olarak adlandırılır. Çekirdek fonksiyonları sayesinde çok yüksek boyutlu bir uzayla uğraşmaya gerek kalmaz. Hem işlem kolaylığı sağlar hem de hesaplama süresini azaltır (Karal, 2018; M. Acı vd., 2016). En çok kullanılan çekirdek fonksiyonları Tablo 4.1'de verilmiştir.

Tablo 4.1 En çok kullanılan çekirdek fonksiyonları

Çekirdek Fonksiyonları	Formül
Polinom Kerneli	$K(x,y) = \big((xy)+1\big)^d$
Normalleştirilmiş Polinom Kerneli	$K(x,y) = \dfrac{\big((xy)+1\big)^d}{\big((xy)+1\big)^d\big((yy)+1\big)^d}$
Radyal Temelli Kernel	$K(x,y) = exp(-\gamma\|x-y\|^2)$
Pearson Evrensel Kernel	$K(x,y) = \dfrac{1}{\left[1+\left(\dfrac{2\sqrt{\|x-y\|^2\sqrt{2^{(1/w)}-1}}}{\sigma}\right)^2\right]^w}$

Radyal temelli kernel fonksiyonu kullanılarak DVR fonksiyonu Eşitlik (4.32)'deki gibi elde edilir.

$$f(x) = \sum_{i=1}^{l} (\lambda_i^* - \lambda_i) exp(-\gamma\|x-y\|^2) + b \tag{4.32}$$

4.8.2 Sınıflandırma Problemlerinde Destek Vektör Makineleri

DVM ile sınıflandırma yaparken, iki sınıfı birbirinden en doğru şekilde ayıran hiperdüzlem bulunmak istenir (Cortes ve Vapnik, 1995). Hiperdüzlemin sınırlarına en yakın gözlem noktaları arasındaki uzaklık marjin olarak adlandırılır. En uygun hiperdüzlem, marjini maksimum yapan düzlemdir. Hiperdüzlemin sınırlarına en yakın gözlem noktaları, destek vektörleridir.

DVM'de sınıflandırma yaparken, verilerin doğrusal ayrılabilme ya da ayrılamama durumlarıyla karşılaşılır. Gerçek yaşam verileri genellikle doğrusal olarak ayrılamaz. Bu durumda sınıflandırma yapabilmek için, veriler doğrusal olarak ayrılabilecekleri daha yüksek boyutlu bir uzaya taşınmalıdır (Eray, 2008).

Doğrusal olarak ayrılamayan verilerde, çözüm için pozitif bir değişken (ξ) kullanılır. Marjinin maksimum olması ve sınıflandırma hatasının minimize edilmesi,

ceza parametresi yardımıyla sağlanabilir (Cortes ve Vapnik, 1995). Bu durumda, DVM'nin amaç fonksiyonu Eşitlik (4.33)'teki gibi tanımlanır.

$$f(\boldsymbol{w}, \xi) = \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{l}\xi_i \tag{4.33}$$

Eşitlik (4.33)'teki, w; model parametre vektörünü, C; ceza parametresini gösterir. Bu fonksiyonun kısıtları Eşitlik (4.34)'teki gibidir.

$$\begin{Bmatrix} y_i(\boldsymbol{w}^T\boldsymbol{x_i} + b) \geq 1 - \xi_i, i = 1,2,\dots,l \\ \xi_i \geq 0, i = 1,2,\dots,l \end{Bmatrix} \tag{4.34}$$

Bu kısıtlar ikinci dereceden optimizasyon problemidir. Doğrusal olarak ayrılamayan verilerde bu problemi lagrange fonksiyonu ile çözmek işlem kolaylığı sağlamaktadır (Eray, 2008). Optimizasyon probleminin lagrange fonksiyonu ile çözümünden elde edilen eşitlik aşağıdaki gibi tanımlanır.

$$L_D = \sum_{i=1}^{l}\boldsymbol{\alpha_i} - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\boldsymbol{\alpha_i\alpha_j}\, y_i y_j \boldsymbol{x_i^T x_j} \tag{4.35}$$

En uygun hiperdüzlemin elde edilmesi için L_D'yi maksimum yapan α_i değerleri bulunur. Bu fonksiyona ait kısıtlar Eşitlik (4.36)'da verilmiştir.

$$\begin{gathered} 0 \leq \boldsymbol{\alpha_i} \leq C, i = 1,2,\dots,l \\ \sum_{i=1}^{l}\boldsymbol{x_i}y_i\,, i = 1,2,\dots,l \end{gathered} \tag{4.36}$$

Doğrusal olarak ayrılamayan verileri sınıflandırabilmek için veriler daha yüksek boyutlu bir uzaya taşınır. Bunun için, "*kernel trick"* olarak bilinen çekirdek fonksiyonları kullanılır. En çok kullanılan çekirdek fonksiyonları Tablo 4.1'de verilmiştir. Çekirdek fonksiyonu Eşitlik (4.37)'deki gibi tanımlanır.

$$K(\boldsymbol{x_i}, \boldsymbol{x_j}) = \varphi(\boldsymbol{x_i})\varphi(\boldsymbol{x_j}) \tag{4.37}$$

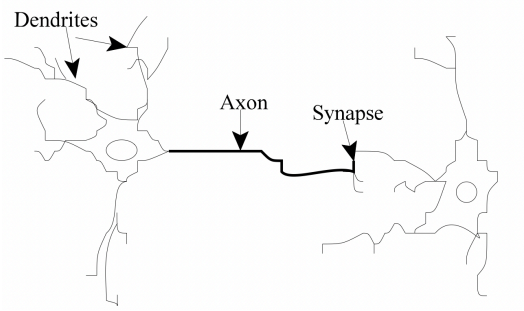
Çekirdek fonksiyonu kullanıldığında DVM fonksiyonu Eşitlik (4.38)'deki gibi gösterilir.

$$f(x) = sign\left(\sum_{i=1}^{l} \boldsymbol{\alpha_i} y_i K(\boldsymbol{x_i}, \boldsymbol{x_j}) + b\right) \tag{4.38}$$

4.9 Yapay Sinir Ağları

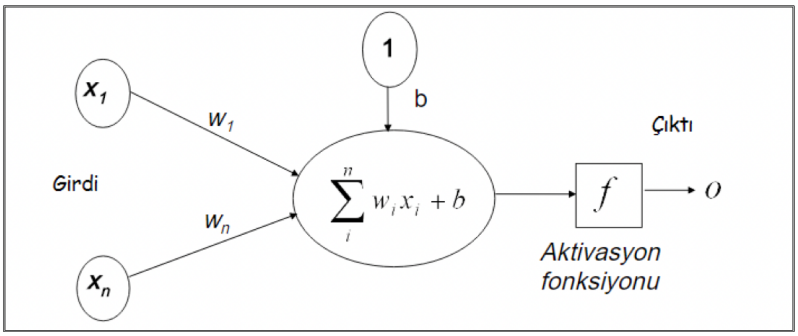
İnsan beyni; yeni şeyler öğrenme, düşünme, hayal etme ve olayları analiz ederek sonuçlar çıkarma konularında oldukça yeteneklidir. Yapay sinir ağları (YSA), insan beynindeki sinir hücrelerinin çalışma prensibini taklit ederek geliştirilen matematiksel modellerdir (Hill, Marquez, O'Connor ve Remus, 1994). YSA, denetimli öğrenmede hem regresyon hem de sınıflandırma problemlerinde kullanılmakla beraber denetimsiz öğrenmede de kullanılır.

Beyin, sinir (nöron) hücrelerinden oluşur. Sinir hücreleri; dendrit, akson ve soma adı verilen yapıları içerir. Dendrit, bir sinir hücresinden başka bir sinir hücresinin gövdesine elektro-kimyasal sinyallerin iletilmesini sağlar. Soma, hücrenin gövdesidir ve hücre çekirdeği ile diğer kimyasal yapıları içerir. Akson, bir sinir hücresinden gelen sinyalleri diğer sinir hücrelerine iletir. İki sinir hücresinin dendritleri arasındaki bağlantı sinapsis olarak adlandırılır (Kukreja, Bharath, Siddesh ve Kuldeep, 2016).



Şekil 4.2 Doğal sinir hücresi örneği (Gershenson, 2003)

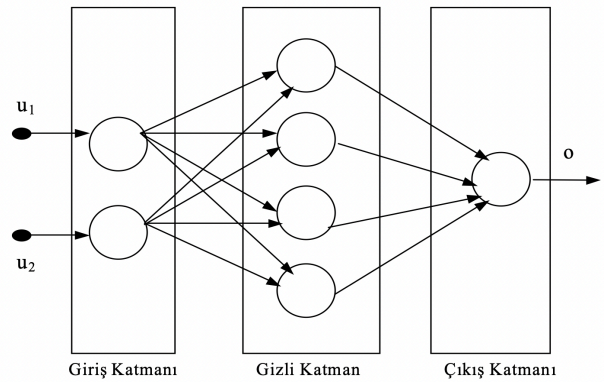
YSA, yapay sinir hücrelerinden oluşur. Bu hücreler, insanlardaki sinir sisteminin biyolojik yapısının matematiksel olarak modellenmesi ile elde edilmeye çalışılır (Asilkan ve Irmak, 2009).



Şekil 4.3 Yapay sinir hücresi örneği (Kurt, Karayılmazlar, İmren ve Çabuk, 2017)

$x_1, \ldots, x_n$; girdileri, $W_1, \ldots, W_n$; YSA arasındaki bağlantıların ağırlık değerlerini temsil eder. Yapay sinir hücrelerine gelen girdiler ile bağlantı ağırlık değerleri çarpılır. Girdi ve ağırlık değerlerinin çarpımları, iletilecek bilginin ne kadar güçlü olduğunu gösterir (Kukreja vd., 2016). Elde edilen çarpım değerleri toplama fonksiyonu yardımıyla toplanır. Toplam değer aktivasyon fonksiyonuna iletilir. Aktivasyon fonksiyonu, çıktı değerinin büyüklüğünü sınırlayarak en doğru sonucun bulunmasını sağlar. Aktivasyon fonksiyonu, kendisine iletilen değeri işleyerek en uygun çıktıyı verir.

YSA, üç katmandan oluşur. Girdilerden oluşan girdi katmanı, çıktılardan oluşan çıktı katmanı ve gizli katmanlar vardır. Girdi değerleri, bağlantıların ağırlık değerleri ile çarpılarak gizli katmana iletilir. Gizli katmana gelen değerler toplanır. Toplam değer, gizli katman ile çıktı katmanı arasında bulunan bağlantıların ağırlık değerleri ile çarpılır ve çıktı katmanına gönderilir. Çıktı katmanı, kendisine gelen değerleri toplayıp çıktı değerini verir (Asilkan ve Irmak, 2009). Girdi ve çıktı katmanlarının sayısına, çalışılan veri setine göre karar verilir. Gizli katman sayısını belirlemek için belirli bir yöntem yoktur. Deneme yanılma yoluyla karar verilir (Öztemel, 2003).



Şekil 4.4 YSA'nın ağ yapısı örneği (Fırat ve Güngör, 2004)

Bağlantıların ağırlık değerlerinin doğru seçilmesi önemlidir. Çünkü, girdi değerlerine göre doğru çıktı değerlerinin elde edilmesi ağırlık değerlerine bağlıdır. Bu değerler algoritmanın başında rastgele olarak belirlenir. Ağların eğitilmesi esnasında, her gözlem değeri ağa gösterilir ve ağın öğrenme kuralına göre ağırlık değerleri değişir. Eğitim setindeki her gözlem için en uygun çıktı değeri elde edilene kadar bu süreç devam eder. Bu süreç sonlandıktan sonra, test setindeki gözlemler ağa gösterilerek çıktı değerleri elde edilir. Doğru çıktı değerlerinin elde edilmesi, ağın eğitilmiş olduğu anlamına gelir (Öztemel, 2003).

YSA, yapay sinir hücreleri arasındaki bağlantıların şekillerine, yönlerine göre ileri ve geri beslemeli ağlar olarak ikiye ayrılır. İleri beslemeli ağlarda, veriler girdi katmanından çıktı katmanına doğru ilerler. Aynı katmandaki gözlemler arasında ya da daha önceki katmanlar arasında bağlantı kurulamaz. Geri beslemeli ağlarda, veriler sadece ileri doğru değil aynı zamanda geriye doğruda ilerleyebilir. YSA'da eğitim hatasını minimize etmek için genellikle geri beslemeli ağlar kullanılır (Asilkan ve Irmak, 2009). Geri beslemeli ağlarda; kabul edilebilir bir hata elde edilene kadar, ağ çıktısı ile gerçek çıktı arasındaki farkları minimize etmek için bağlantı ağırlıkları değiştirilir.

4.10 Sınıflandırma ve Regresyon Ağaçları

Sınıflandırma ve regresyon ağaçları (SRA), doğrusal olmayan makine öğrenmesi yöntemlerinden biridir. Çalışılan veri setine göre farklı adlandırılır. Bağımlı değişken kategorik ise sınıflandırma, sürekli ise regresyon ağacı kullanılır (Chang ve Wang, 2006). SRA'da; karmaşık veri yapılarını daha basit, anlaşılabilir karar yapılarına dönüştürmek amaçlanır (Berry ve Linoff, 2004).

SRA, veri setinin yapısına göre sorular sorarak karar kuralları oluşturur. Bu kurallara göre, ağaç yapısına benzeyen sınıflandırma ve regresyon modelleri elde edilir.

SRA'nın yapısı; kök, düğüm, dal ve yapraklardan oluşur. Kök, bütün gözlemleri içerir. Kökten aşağı doğru inildikçe, veri setini homojen alt gruplara ayıran dallar görülür. Düğüm, kökten dallara doğru büyüyen ağaçtaki boğumlar olarak tanımlanır (Pehlivan, 2006). Veri setinde, bağımlı değişkeni en çok etkileyen bağımsız değişken en fazla bölünmeye sebep olan değişkendir. Bu değişken üzerinden bölünme başlar. Düğümler üzerinde en fazla bölünmeye sebep olan değişkenlere test işlemi uygulanır. Bu testin sonucunda, düğümlerden dallar oluşarak ağaç büyür. Dallar, testlerin sonuçlarını gösterir. Elde edilen sonuçlara göre, algoritma sonlanırsa yaprak elde edilir. Eğer algoritma sonlanmazsa, tekrar bir düğüm oluşur. Aynı işlemler optimal sonuca ulaşana kadar devam eder.

SRA'nın avantajlarından biri, aykırı değerlere karşı sağlam olmasıdır. Diğer bir avantajı ise, ağaç yapısının bağımsız değişkenlerin dönüşümlerine karşı duyarsız olmasıdır. Değişkenlerin logaritmik ya da karekök dönüşüm değerleri kullanılsa bile ağacın yapısı değişmez (Timofeev, 2004).

SRA yönteminde, maksimum ağacın oluşturulması, budanması ve en uygun ağaç yapısının seçilmesi şeklinde üç aşama vardır.

Maksimum ağacın oluşturulması: Veri setindeki bütün gözlemleri içeren kökten başlayıp, her düğümde belirlenen değişkene göre dallara ayrılarak ağaç oluşturulur. En küçük alt grup elde edilene kadar bölünme devam eder. Bölünme işlemi bazı saflık ölçütleri (Gini katsayısı, Twoing) kullanılarak yapılır. En yaygın kullanılan saflık ölçütü Gini katsayısıdır. Gini ölçütü, her bölünmede en büyük veri kümesini verir ve bölünmeden sonra ilgilenilmeyen kısmı dışarıda bırakır. Ayrıca, gürültülü verilerin varlığında da iyi sonuçlar verir (Yavuz ve Vupa Çilengiroğlu, 2020; Timofeev, 2004).

Ağaç budama: SRA algoritmasında herhangi bir durdurma parametresi olmadığı için, ağaç sürekli bölünerek büyür. Bu durum, ağacın yanlış sınıflandırma oranı düşük olmasına rağmen, yeni verilere uygulandığında yanlış sonuçlar vermesine sebep olabilir (Sezgin, 2006). Buna engel olmak için, ağacın karmaşıklığı ile tahmin başarısı arasında bir denge kurulmalıdır (Yavuz ve Vupa Çilengiroğlu, 2020). Bu dengeyi kurmak için de, ağaca budama işlemi uygulanır. Bazı dallar ya da alt dallar kaldırılarak bu dallardaki gözlemler, kendilerine benzer özellikler taşıyan başka dallara yerleştirilir. Böylece budama işlemi gerçekleşmiş olur (Kuyucu, 2012).

En uygun ağaç yapısının seçimi: SRA algoritmasında ayarlama parametresi karmaşıklık katsayısıdır. Karmaşıklık katsayısı, ağacın dallanmasını sınırlayarak budama işleminin ne zaman yapılacağını belirler. En uygun ağaç yapısını seçmek için çapraz doğrulama yöntemi kullanılır. Bu yöntem ile, ağacın karmaşıklığı ve tahmin hatası arasındaki optimal orantıyı elde etmek amaçlanır (Timofeev, 2004). Çapraz doğrulama sürecinde, karmaşıklık katsayısı için bir değer kümesi belirlenir. Her bir değer sonucunda elde edilen ağaçlar test seti ile sınanır. Tahmin başarısı en yüksek olan ağaç seçilir.

4.10.1 Sınıflandırma Ağaçları

Sınıflandırma ağaçları; Breiman, Friedman, Olshen ve Stone (1984) tarafından, verileri ağaçlar yardımıyla analiz etmek için geliştirilen bir algoritmadır. Sınıflandırma ağacı oluşturmak için, bir öğrenme örneğine ihtiyaç vardır. Öğrenme örneği, bağımsız değişkenler matrisi ve sınıf değerlerinden oluşur (Sezgin, 2006). Sınıflandırma ağacı,

öğrenme örneğinin homojen daha küçük parçalara bölünmesi için karar kuralları oluşturur. Her bölünmede maksimum homojenliğe sahip düğümler elde edilmeye çalışılır. Maksimum homojenlik için safsızlık fonksiyonu kullanılır.

t_P; üst düğümü, t_l ve t_r; üst düğümün bölünmesiyle elde edilen sol ve sağ alt düğümleri, x_j; j. bağımsız değişkeni, M; değişken sayısını, N; gözlem sayısını K; toplam sınıf sayısını, x_j^R; x_j'nin en iyi bölme değerini ve $i(t)$; safsızlık fonksiyonunu göstersin. Sol ve sağ alt düğümlerinin maksimum homojenliği, safsızlık fonksiyonun değişiminin maksimumuna eşittir. Bu durumda Eşitlik (4.39) yazılabilir.

$$\Delta i(t) = i(t_P) - E[i(t_c)] \tag{4.39}$$

t_c; sol ve sağ alt düğümleri, P_l ve P_r; sağ ve sol alt düğümlerin olasılıklarını temsil etsin. Bu durumda, sınıflandırma ağacının maksimizasyonu Eşitlik (4.40)'daki gibi tanımlanır.

$$argmax_{(x_{j\leq}x_j^R, j=1,\ldots,M)}\left[i(t_p) - P_l i(t_l) - P_r i(t_r)\right] \tag{4.40}$$

Uygulamalarda kullanılan safsızlık fonksiyonları Gini ve Twoing yöntemleridir (Timofeev, 2004). Gini safsızlık fonksiyonu Eşitlik (4.41)'deki gibidir.

$$\begin{aligned} argmax_{x_{j\leq}x_j^R, j=1,\ldots,M}\Bigg[& -K\sum_{k=1}^{K} p^2(k|t_p) \\ & + P_l\sum_{k=1}^{K} p^2(k|t_l) + P_r\sum_{k=1}^{K} p^2(k|t_r)\Bigg] \end{aligned} \tag{4.41}$$

Twoing safsızlık fonksiyonu Eşitlik (4.42)'deki gibi tanımlanır.

$$argmax_{x_{j\leq}x_j^R, j=1,\ldots,M}\left(\frac{P_l P_r}{4}\left[\sum_{k=1}^{K}|p(k|t_l) - p(k|t_r)|\right]^2\right) \tag{4.42}$$

4.10.2 Regresyon Ağaçları

Tarihte bilinen ilk regresyon ağaçları, Morgan ve Messenger (1973) tarafından geliştirilen Otomatik Etkileşim Algılama algoritmasıdır. Regresyon ağaçlarında, bağımlı değişken (Y) sürekli değerler alır. Bağımlı değişkenin tahmin değerlerini elde etmek için ağacın her düğümünde bir regresyon modeli yer alır (Loh, 2011). Regresyon ağaçlarında önceden belirlenen sınıflar olmadığından bölme kurallarını belirlemek için safsızlık fonksiyonlarına ihtiyaç duyulmaz (Timofeev, 2004). Bölme işlemi, iki düğüm için beklenen toplam varyansın minimize edilmesi mantığına dayanarak yapılır. Regresyon ağacının bölünme kuralı Eşitlik (4.43)'teki gibi tanımlanır.

$$argmax_{x_j \leq x_j^R, j=1,\ldots,M} [P_l V(Y_l) + P_r V(Y_r)] \tag{4.43}$$

4.11 Topluluk Öğrenme Yöntemleri

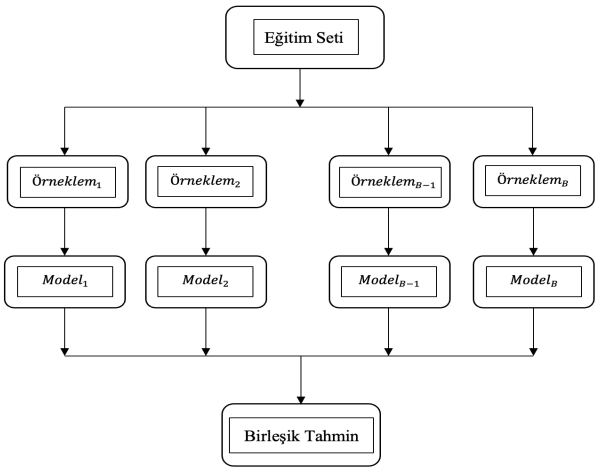
Topluluk öğrenme yöntemleri, aynı problemi çözmek için birden fazla öğrenicinin eğitildiği makine öğrenmesi algoritmalarıdır (Zhou, 2009). Birden fazla sınıflandırıcı ya da tahmin edici ile modeller oluşturulur ve tahmin sonuçları birleştirilir. Bu yöntemler ile sınıflandırma ve tahmin doğruluğunu artırmak amaçlanır. Topluluk öğrenme yöntemleri kendi içlerinde, torbalama ve yükseltme algoritmaları olarak iki gruba ayrılır.

4.11.1 Torbalama Algoritması

Torbalama algoritması (TA); Breiman (1996) tarafından, var olan makine öğrenmesi prosedürlerine (karar ağaçları) göre daha düşük varyanslı tahminler elde etmek amacıyla geliştirilmiştir. Eğitim setinden, yerine konularak ve rastgele bir şekilde (bootstrap) n tane alt örneklem seçilir. Bu işlem genellikle 50 ya da 100 tekrarla yapılır (Işıkhan, 2014). Alt örneklemlerin her biri ile eş zamanlı olarak karar ağacı modelleri oluşturulur. Karar ağaçlarını oluşturmak için SRA yöntemi kullanılır. Eğitim setinden seçilen alt örneklemlerin birbirinden bağımsız olması sebebiyle,

oluşan karar ağaçları da birbirlerinden bağımsızdır. Her alt örneklem seçiminde, eğitim setindeki gözlemler eşit olasılıkla seçilir. Dolayısıyla, gözlemlerin oluşturulan her bir karar ağacında kullanılma olasılıkları eşittir (Breiman, 1996; Özkan, 2019). Eğitim setindeki gözlemlerin yaklaşık olarak %67'si karar ağaçlarının oluşturulması için, kalan %33'ü ise ağaçların performans değerlendirilmesi için kullanılır (B. Singh, Sihag ve K. Singh, 2017).

Alt örneklemler ile oluşturulan karar ağacı modellerinden elde edilen tahminler birleştirilerek değerlendirilir ve nihai sonuca karar verilir. Regresyon problemlerinde sonuç, model tahminlerinin ortalaması alınarak belirlenir. Sınıflandırma problemlerinde ise, çoğunluk oylaması yapılarak en sık gözlenen sınıf final tahmini olarak belirlenir (Ay, 2019).



Şekil 4.5 Torbalama algoritmasının çalışma prensibi

4.11.2 Rastgele Orman Algoritması

Rastgele orman (RO) algoritması, Breiman (2001) tarafından önerilmiştir. SRA ile oluşturulan birden çok karar ağacından elde edilen tahminlerin birleştirilerek değerlendirilmesi mantığı ile çalışır. Karar ağaçlarını oluşturacak gözlemler, TA'da olduğu gibi, eğitim setinden yerine konularak ve rastgele şekilde seçilir. TA'dan farklı olarak, düğümlerin bölünmesine karar verirken kullanılacak değişkenler rastgele olarak seçilir. Her bir düğümü, bütün değişkenler arasından en çok bölünmeye neden

olan değişken ile bölmek yerine, veri setinden rastgele olarak seçilen alt değişken kümesi içinde en çok bölünmeye neden olan değişken ile böler. Karar ağaçları oluşturulurken SRA yöntemi kullanılır. Elde edilen ağaçlar budanmaz (Breiman, 2001).

RO algoritmasını başlatmak için bazı parametrelerin tanımlanması gerekir. Düğümlerde bölünmeye karar verirken kullanılacak değişken sayısı (m) ve oluşturulacak ağaç sayısı (k) kullanıcının tanımlayacağı parametrelerdir (Breiman, 1999). İlk olarak, eğitim setinin 2/3'ünden yerine konarak ve rastgele şekilde alt örneklemler seçilir. Eğitim setinin geri kalan 1/3'ü ile ağaçların tahmin performansları test edilir. Alt örneklemlerin her biri ile budanmamış ağaçlar oluşturulur. Düğümlerde m tane değişken, veri setindeki değişkenlerin alt kümesi olarak seçilir. Seçilen değişkenlerden en çok bölünmeye neden olan değişken belirlenerek, bu değişkene göre bölünmeler yapılır (Akar ve Güngör, 2012; Breiman, 2004). Ağaçlardan elde edilen modeller birleştirilerek değerlendirilir ve sonuca karar verilir. Torbalama yönteminde olduğu gibi; regresyon problemlerinde model tahminlerinin ortalaması, sınıflandırma problemlerinde ise en sık gözlenen sınıf final tahmini olarak belirlenir.

RO algoritmasının birçok avantajı vardır. Büyük hacimli ve fazla değişkenli veri setlerinde, değişken ya da gözlem çıkarmaya gerek olmadan verimli çalışır (Breiman, 2004). Ağaçlarda bölünmeye karar vermek için seçilecek değişkenin, veri setindeki değişkenlerin alt kümesinden seçilmesi, ağaçlar arasındaki korelasyonu azaltır. Bu durum, tahmin doğruluğu yüksek modeller elde edilmesini sağlar (Suchetana, Rajagopalan ve Silverstein, 2017).

4.11.3 Yükseltme Algoritmaları

Yükseltme algoritmaları, sınıflandırma ve regresyon problemlerinde kullanılan diğer öğrenme yöntemlerine göre daha güçlü tahminler elde etmek için Schapire (1990) tarafından önerilmiştir. Temel çalışma prensipleri, veri setindeki gözlemlere ağırlık değerleri vererek ağaçlar topluluğu oluşturmaktır. Amaç; birden fazla model

oluşturarak, hepsinden gelen bilgileri toplayıp güçlü ve tahmin başarısı yüksek bir final modeli elde etmektir.

Bu çalışmada, literatürdeki yükseltme algoritmalarından; gradyan yükseltme ve aşırı gradyan yükseltme makineleri incelenecektir.

4.11.4 Gradyan Yükseltme Makineleri

Gradyan yükseltme makineleri (GYM); zayıf öğrenen birçok modeli tekrarlı olarak birleştirerek, güçlü öğrenen bir final modeli elde etmek amacıyla Friedman (2001) tarafından geliştirilmiştir.

GYM algoritması, kayıp fonksiyonunu minimize eden modeli bulmayı amaçlayan bir optimizasyon problemidir (Touzani, Granderson ve Fernandes, 2018). Kayıp fonksiyonu, model tahminlerinin gerçek değerlerden ne kadar farklı olduğunu hesaplayarak modelin başarısını ölçer.

GYM algoritması aşamalı bir yöntemdir. İlk iterasyonda, tahmin fonksiyonu olarak bir karar ağacı oluşturulur. Bu ağaçtan elde edilen tahminler ile gerçek değerler arasındaki farklar alınarak kayıp fonksiyonunda saklanır. İkinci iterasyonda, tahmin ve kayıp fonksiyonları birleştirilerek tekrar bir ağaç oluşturulur. Her adımda tahmin hatalarının eklenmesi, tahmin fonksiyonunun başarısını artırır (Parr ve Howard, 2018). Kullanıcının belirlediği iterasyon sayısı kadar bu işlemler devam eder.

GYM'de, kullanıcı tarafından ayarlanması gereken parametreler vardır. Ağaç derinliği (d), ağacın ne kadar dallanacağını kontrol eder. İterasyon sayısı (K), oluşturulacak ağaç sayısını ifade eder. Öğrenme oranı (α), algoritmanın öğrenme oranını gösterir ve genellikle 0 ile 1 arasında küçük bir değer alır (Muratlar, 2020). η, her tekrarda ağaç oluştururken kullanılacak minimum gözlem sayısıdır.

Öğrenme oranının (α) değeri sıfıra ne kadar yakınsa, modelin tahmin başarısı o kadar artar. Ancak, bu durum algoritmanın yavaşlamasına neden olabilir. Bu problemi çözmek için, iterasyon sayısı (K) artırılabilir (Touzani vd., 2018).

GYM algoritmasının adımları şu şekilde tanımlanabilir (Lu, Karimireddy, Ponomareva ve Mirrokni, 2020):

1. Başlangıçta; artık değerleri (gözlem ve tahmin değerleri arasındaki farklar) ve tahmin fonksiyonu, boş fonksiyon olarak tanımlanır.
2. Kayıp fonksiyonu tanımlanır.

$$L\big(y, f(x)\big) \tag{4.44}$$

3. Kayıp fonksiyonun birinci dereceden türevi alınarak artık hesaplaması yapılır.

$$r^m = -\left[\frac{\partial L\big(y_i, f^m(x_i)\big)}{\partial f^m(x_i)}\right], i = 1, \dots, n \tag{4.45}$$

4. En iyi zayıf öğrenen modelin parametreleri bulunur.

$$\tau_k = argmin \sum_{i=1}^{n} \left(r_i^k - b_\tau(x_i)\right)^2 \tag{4.46}$$

5. Ağaç oluştururken kullanılacak minimum gözlem sayısına karar verilir.

$$argmin_\eta = \sum_{i=1}^{n} L\left(y_i, f^k(x_i) + \eta b_{\tau_k}(x_i)\right) \tag{4.47}$$

6. Her tekrarda model güncellenir.

$$f^{k+1} = f^k(x) + \eta_k b_{\tau_k}(x) \tag{4.48}$$

4.11.5 Aşırı Gradyan Yükseltme Makineleri

Aşırı gradyan yükseltme makineleri (AGY), diğer makine öğrenmesi yöntemlerinin zorluklarına, eksikliklerine çözüm olarak Chen ve Guestrin (2016) tarafından önerilmiştir. AGY algoritmasının başarılı olmasının temel sebebi, olası bütün senaryolara uyum sağlayabilmesi, ölçeklenebilmesidir. Aynı zamanda, var olan yöntemlere göre on kat daha hızlı çalışan bir algoritmadır (Chen ve Guestrin, 2016). Genel olarak GYM ile benzer şekilde çalışır. GYM'den farklı olarak, amaç fonksiyonuna bir düzenleme terimi ekler (Liang, Luo, Zhao ve Wu, 2020). AGY algoritmasının amaç fonksiyonu Eşitlik (4.49)'daki gibi tanımlanır.

$$\sum_{i=1}^{n} L\big(y_i, f(x_i)\big) + \sum_{i=1}^{t} \Omega(f_i) + C \tag{4.49}$$

$L\big(y_i, f(x_i)\big)$; kayıp fonksiyonunu, $\Omega(f_i)$; düzenleme terimini, C; sabit terimi ifade eder. GYM'deki gibi, kayıp fonksiyonunun birinci dereceden türevini almak yerine ikinci dereceden türevi de alınarak amaç fonksiyonu tanımlanır ve model hedeflenen kayıp fonksiyonuna göre eğitilir (Mo, Sun, Liu ve Wei, 2019).

$$\sum_{i=1}^{t} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) \tag{4.50}$$

g_i ve h_i; kayıp fonksiyonundaki birinci ve ikinci dereceden türevlerini gösterir. Model karmaşıklığını azaltmak ve diğer veri setlerinde kullanılabilirliğini sağlamak için düzenleme terimi Eşitlik (4.51)'deki gibi tanımlanır.

$$\Omega(f_t) = \gamma T + \frac{1}{2} \lambda \|w\|^2 \tag{4.51}$$

T; ağaçlardaki yaprak sayısını, w; yaprakların ağırlık değerlerini, γ ve λ; yaprak sayısı ve ağırlık değerlerini kontrol eden ceza parametrelerini ifade eder. Ceza

parametreleri, ağaçların karmaşıklığını azaltmak ve algoritma tarafından üretilen modelleri basitleştirmek için kullanılır (Pan, 2018).

AGY algoritmasında, en yüksek performansı elde edebilmek için ayarlanması gereken birkaç parametre vardır. İterasyon sayısı, oluşturulacak ağaç sayısıdır. Maksimum ağaç derinliği, ağaçlardaki maksimum bölünme sayısıdır. Maksimum derinliğin artırılması modelin başarısını düşürebilir. Modeli eğitmek için kullanılacak gözlem oranı diğer bir ayarlama parametresidir. Her iterasyonda, yaprakların ağırlık değerlerini azaltmak ve modelin tahmin başarısını yükseltmek için eğitim oranı parametresi kullanılır. Oluşturulacak ağaçlarda değişkenlerden alınacak örneklem oranı, ayarlanması gereken diğer bir parametredir. λ ve α, düzenleme terimleridir ve modelin öğrenme sürecindeki hatalarını kontrol ederler (Parsa, Movahedi, Taghipour, Derrible ve Mohammadian, 2020).

4.12 Lojistik Regresyon Analizi

Lojistik regresyon analizi (LRA), sınıflandırma problemlerinde kullanılan makine öğrenmesi yöntemlerindendir. LRA, regresyon analizinde olduğu gibi, bağımlı değişken ile bağımsız değişkenler arasındaki ilişkiyi tanımlayabilen bir model elde etmeyi amaçlar. LRA'yı regresyon analizinden ayıran temel özellik ise, bağımlı değişkenin kategorik değişken olmasıdır. İkili, çoklu ya da sıralı kategorilere sahip bağımlı değişkenin, bağımsız değişkenlerle ilişkisi modellenmeye çalışılır (Çolak, 2001).

LRA'yı regresyon analizinden ayıran üç önemli özellik vardır (Atakurt, 1999):

1. Bağımlı değişken; regresyon analizinde sürekli yapıdayken, LRA'da kategorik (kesikli) yapıdadır.
2. Regresyon analizinde bağımsız değişkenin değeri tahmin edilirken, LRA'da bağımlı değişkeninin beklenen değerleri olasılık olarak tahmin edilir.
3. Regresyon analizinde bağımsız değişkenlerin normal dağılması koşulu varken, LRA'da bağımsız değişkenlerin dağılımıyla ilgili bir koşul yoktur.

4.12.1 Lojistik Regresyon Modeli

Regresyon analizinde; bağımsız değişken (X) bilinirken, bağımlı değişkeninin (Y) ortalama değeri bulunmak istenir. Aslında, koşullu bir ortalama değeri elde edilmeye çalışılır. Bu değer, $E(Y|X)$ şeklinde gösterilir. x değeri, $+\infty$ ile $-\infty$ arasında değer aldığı için, $E(Y|X)$ de olası bütün değerleri alabilir. LRA'da ise, bağımlı değişkeninin beklenen değerleri olasılık olarak tahmin edildiği için, 0 ile 1 arasında değer almak zorundadır (Çolak, 2001).

$$P_i = E(Y|X_i) = \frac{e^{\beta_0+\beta_1 X_i}}{1 + e^{\beta_0+\beta_1 X_i}} \tag{4.52}$$

$T_i = \beta_0 + \beta_1 X_i$ olarak dönüşüm yapıldığında, lojistik dağılım fonksiyonu Eşitlik (4.53)'teki gibi yazılabilir.

$$P_i = \frac{1}{1 + e^{-T_i}} \tag{4.53}$$

T_i'nin ve dolayısıyla x_i'nin, lojistik dağılım fonksiyonuyla doğrusal bir ilişkiye sahip olmadığı Eşitlik (4.53)'de görülmektedir. LRA'da, modelin yorumlanabilirliği önemli olduğu için doğrusal bir model elde edilmek istenir. Beklenen olasılık değerleri bazı dönüşümler yardımıyla, $+\infty$ ile $-\infty$ arasında tanımlı hale getirilir. Böylece, x_i ve lojistik dağılım fonksiyonu arasında doğrusal bir ilişki elde edilir. Bunun için, lojistik dağılım fonksiyonuna logit dönüşümü uygulanır (Gujarati ve Porter, 2010). P_i, kategorik değişken için istenilen olayın gerçekleşme olasılığıdır. Olayın gerçekleşmeme olasılığı ise, $(1 - P_i)$ olarak ifade edilir. Olayın gerçekleşme olasılığının gerçekleşmeme olasılığına oranı odds oranı olarak adlandırılır ve Eşitlik (4.54)'teki gibi tanımlanır.

$$\frac{P_i}{1 - P_i} = \frac{1 + e^{T_i}}{1 + e^{-T_i}} = e^{T_i} \tag{4.54}$$

Eşitlik (4.54)'ün logaritması alınır.

$$g(x) = ln\left(\frac{P_i}{1-P_i}\right) = lne^{T_i} \tag{4.55}$$

T_i değeri yerine yazılırsa lojistik regresyon modeli elde edilir.

$$g(x) = ln\left(e^{\beta_0+\beta_1 X_i}\right) = \beta_0 + \beta_1 X_i \tag{4.56}$$

$g(x)$, logit dönüşüm fonksiyonudur ve doğrusal hale getirilen regresyon modelinin bütün özelliklerini taşır (Atakurt, 1999). $g(x)$; 0,5'ten büyük veya eşit ise $Y = 1$, 0,5'ten küçük ise $Y = 0$ olacak şekilde sınıflandırma yapılır (Agresti, 2010).

4.12.2 Lojistik Regresyon Katsayılarının Önem Kontrolü

LRA'da, regresyon katsayıları tahmin edildikten sonra modele katkılarını araştırmak için önem kontrolü yapılır. Bu kontrolün temeli, önem kontrolü yapılan değişkenin bulunduğu ve bulunmadığı modellerden elde edilen tahmin değerleri ile gerçek gözlem değerlerinin karşılaştırılmasına dayanır (Atakurt, 1999). Bu karşılaştırma, log-olabilirlik fonksiyonu kullanılarak Eşitlik (4.57)'de ifade edildiği gibi yapılır.

$$D = -2ln\left[\frac{Şu\ andaki\ modelin\ olabilirliği}{Doymuş\ modelin\ olabilirliği}\right] \tag{4.57}$$

Bağımsız bir değişkenin önem kontrolünü yapmak için, bağımsız değişkenin bulunduğu ve bulunmadığı durumlardaki D değerleri, 1 serbestlik derecesi ile ki-kare dağılan G istatistiği yardımıyla karşılaştırılır (Hosmer ve Lemeshow, 2000).

$$G = -2ln\left(\frac{Değişkensiz\ modelin\ olabilirliği}{Değişkenli\ modelin\ olabilirliği}\right) \tag{4.58}$$

LRA'da regresyon katsayılarının yorumlanması, regresyon analizindeki yorumlamadan farklıdır. Bağımsız değişkendeki 1 birimlik artışın bağımlı değişkeni

nasıl etkilediği, odds tahmini $\left(\frac{P_i}{1-P_i}\right)$ ile β_i'nin beklenen değerinin çarpımından elde edilen lojistik regresyon fonksiyonundan yararlanılarak bulunur (Aktaş, 2009).

BÖLÜM 5
GERÇEK HAYAT VERİ UYGULAMALARI

Makine öğrenmesi algoritmalarına bilginin öğretilmesi ve öğrenme konusunda ne kadar başarılı olduklarının test edilmesi için veri seti bölme stratejisi kullanılır. Literatürde bu işlem, kullanıcının belirlediği bir oranda rastgele olarak yapılır. Bu bölümde, veri seti bölme işlemi literatürde kullanılan yönteme ek olarak; SKÖ, USKÖ, MSKÖ ve YSKÖ yöntemleri ile yapılmaktadır. Bu yöntemler kullanılarak; farklı çalışma alanlarından seçilen gerçek hayat verileri, eğitim ve test setlerine bölünüp denetimli öğrenme algoritmaları ile R yazılımı kullanılarak analiz edilmiştir. Algoritmalarda model seçimi, 10-katlı çapraz doğrulama yöntemi kullanılarak yapılmıştır. Değerlendirme ölçütü olarak; regresyon problemlerinde HKOK değeri, sınıflandırma problemlerinde ise doğru sınıflandırma oranı esas alınmıştır. Genellenebilir sonuçlar elde etmek için, kullanılan algoritmalar 5000 tekrarla çalıştırılmış, ortalama HKOK ve doğruluk oranı değerleri elde edilmiştir.

5.1 Regresyon Uygulamaları

Regresyon problemlerinde kullanılan denetimli öğrenme algoritmaları 4 gerçek hayat veri setine uygulanmıştır. Veri setleri; rastgele olarak ve SKÖ, USKÖ, MSKÖ ve YSKÖ yöntemleri kullanılarak eğitim ve test setlerine ayrılmıştır. Kullanılan veri setlerinin özellikleri Tablo 5.1'de verilmiştir.

Tablo 5.1 Regresyon uygulamalarında kullanılan gerçek hayat veri setleri

Veri Seti	Değişken Sayısı	Gözlem Sayısı
Boston'daki Konut Fiyatları	14	506
Beton Dayanıklılığı	9	1030
Vücut Yağ Yüzdesi	15	252
İzmir Hava Durumu	10	1461

5.1.1 Gerçek Hayat Veri Uygulaması-1

İlgili veri seti (http://lib.stat.cmu.edu/datasets/boston); ABD Nüfus Servisi tarafından, Boston'ın Mass bölgesindeki konutlarla ilgili toplanan bilgileri içerir. 14 değişken, 506 gözlem değerinden oluşmaktadır.

Veri seti ilk önce; %75 eğitim, %25 test seti olacak şekilde rastgele olarak, daha sonra SKÖ, USKÖ, MSKÖ ve YSKÖ ile bölünmüştür. SKÖ ve modifikasyonları ile veri seti bölünürken; veri setinin %75'ine karşılık gelen 381 gözlem değeri, $m = 3$, $r = 127$ alınarak seçilmiştir. Elde edilen sonuçlar Tablo 5.2'deki gibidir.

Tablo 5.2 Boston konutları verisine ait ortalama HKOK değerleri

Yöntemler	**Rastgele**	**SKÖ**	**USKÖ**	**MSKÖ**	**YSKÖ (p=0,4)**
ÇDR	4,89458	4,865039	**4,839567**	7,240136	7,642516
TBR	5,500465	5,470636	**5,467332**	8,148436	8,609874
KEKKR	**4,884418**	5,008933	5,004721	7,443948	7,831883
RR	4,882042	4,890738	**4,854547**	7,330171	7,714915
LR	4,904839	4,864981	**4,864365**	7,325039	7,732485
ENR	4,892445	**4,870672**	4,872949	7,324655	7,76271
KNN	3,838652	3,789041	**3,785072**	6,968652	7,497769
DVR	**3,419738**	3,465457	3,478317	6,654256	6,842399
YSA	9,267162	**9,187795**	9,195906	11,92301	12,59621
SRA	**4,535681**	4,535936	4,539484	6,463611	6,609511
TA	3,369989	3,360138	**3,359242**	5,503337	5,706337
RO	3,285335	**3,256902**	3.260498	5,84717	6,212126
GYM	**4,172611**	4,209713	4,211961	6,87984	7,114572
AGY	3,314765	3,405998	**3,281136**	5,294572	5,639219

KEKKR, DVR, SRA ve GYM algoritmalarında, rastgele olarak bölünen veri seti ile; ENR, YSA ve RO algoritmalarında, SKÖ kullanılarak bölünen veri seti ile; ÇDR, TBR, RR, LR, KNN, TA ve AGY algoritmalarında ise USKÖ ile bölünen veri seti ile daha düşük hata değerleri elde edilmiştir.

5.1.2 Gerçek Hayat Veri Uygulaması-2

İlgili veri seti (https://archive.ics.uci.edu/ml/datasets/concrete+compressive+strength), Kaliforniya Üniversitesi tarafından oluşturulan makine öğrenmesi veri seti deposundan alınmıştır. İnşaat mühendisliğinde önemli bir malzeme olan betonun, basınç dayanıklılığı ile ilgili bilgiler içerir. 1030 gözlem değeri ve 9 değişkenden oluşmaktadır.

İlk aşamada veri seti; %79 eğitim, %21 test seti olacak şekilde rastgele olarak bölünmüştür. Daha sonra, veri setinin %79'una karşılık gelen 816 gözlem değeri, $m = 2$, $r = 408$ alınarak SKÖ ve modifikasyonları ile seçilmiştir. Sonuçlar Tablo 5.3'teki gibidir.

Tablo 5.3 Beton basınç dayanıklılığı verisine ait ortalama HKOK değerleri

Yöntemler	Rastgele	SKÖ	USKÖ	MSKÖ	YSKÖ (p=0,5)
ÇDR	10,47209	10,445	10,45042	**10,43927**	14,57092
TBR	13,71368	13,68846	13,68125	**13,61521**	20,59968
KEKKR	10,78254	10,9165	**10,69178**	10,74481	15,2116
RR	**10,49052**	10,49394	10,4943	10,49445	14,52185
LR	10,47993	10,44826	10,45063	**10,44097**	14,57322
ENR	10,487	**10,46302**	10,46866	10,47119	14,67996
KNN	**8,891513**	8,91161	8,934214	8,933669	14,12405
DVR	**5,701943**	5,757624	5,723805	5,723639	8,658101
YSA	17,71232	17,74543	**17,63986**	17,73992	23,76996
SRA	7,489904	7,477541	**7,467827**	7,474236	11,21755
TA	5,005375	5,017532	5,009921	**4,997319**	8,944404
RO	5,059752	5,040211	5,049367	**5,038902**	9,711622
GYM	9,109955	9,03683	9,03722	**9,03126**	15,6719
AGY	4,373615	**4,282321**	4,381073	4,378757	7,255895

RR, KNN ve DVR algoritmalarında, rastgele olarak bölünen veri seti ile; ENR ve AGY algoritmalarında, SKÖ kullanılarak bölünen veri seti ile; KEKKR, YSA ve SRA algoritmalarında, USKÖ ile bölünen veri seti ile; ÇDR, TBR, LR, TA, RO ve GYM

algoritmalarında ise MSKÖ kullanılarak bölünen veri seti ile daha düşük hata değerleri elde edilmiştir.

5.1.3 Gerçek Hayat Veri Uygulaması-3

İlgili veri seti (https://www.kaggle.com/fedesoriano/body-fat-prediction-dataset), veri bilimciler için birçok veri setini içeren ve bu veriler üzerinden yarışmaların yapıldığı bir platform olan Kaggle'dan alınmıştır. Veri seti, vücut yağ yüzdesini tahmin etmek için 252 erkekten alınan; yaş, ağırlık, boy ve farklı vücut ölçülerini içerir. 252 gözlem değeri ve 15 değişkenden oluşmaktadır.

Veri seti; %77 eğitim, %23 test seti olacak şekilde rastgele olarak bölünmüştür. Daha sonra, veri setinin %77'sine karşılık gelen 196 gözlem değeri SKÖ ve modifikasyonları kullanılarak seçilmiştir. $m = 4$, $r = 49$ olarak alınmıştır. Sonuçlar Tablo 5.4'teki gibidir.

Tablo 5.4 Vücut yağ yüzdesi verisine ait ortalama HKOK değerleri

Yöntemler	**Rastgele**	**SKÖ**	**USKÖ**	**MSKÖ**	**YSKÖ (p=0,3)**
ÇDR	1,175655	1,426032	1,356743	1,35236	**0,7962619**
TBR	4,445526	3,937632	**3,618792**	4,98011	5,389904
KEKKR	1,881661	2,270852	**1,834858**	2,07387	2,390283
RR	1,235939	1,219043	1,435568	1,41825	**1,182725**
LR	1,135816	1,174807	1,216705	1,326158	**0,7601392**
ENR	**1,121696**	1,163691	1,381953	1,356576	1,16381
KNN	3,479474	3,404113	**3,187743**	5,04436	4,715015
DVR	2,543835	2,52992	**1,687381**	5,144053	4,913521
YSA	6,922871	6,75627	**6,497686**	8,203567	8,226981
SRA	1,64821	1,662771	**1,385427**	3,329818	2.765547
TA	**1,325029**	1,336291	1,364071	2,714089	2,436876
RO	1,924513	1,906167	**1,506004**	3,868859	3,233037
GYM	1,684333	1,641986	**1,246797**	3,90008	3,299145
AGY	1,646011	1,687401	**1,644031**	3,278521	2,611414

ENR ve TA yöntemlerinde, rastgele olarak bölünen veri seti ile; TBR, KEKKR, KNN, DVR, YSA, SRA, RO, GYM ve AGY algoritmalarında, USKÖ kullanılarak bölünen veri seti ile; ÇDR, RR, LR algoritmalarında ise YSKÖ ile bölünen veri seti ile daha düşük hata değerleri elde edilmiştir.

5.1.4 Gerçek Hayat Veri Uygulaması-4

İlgili veri seti (http://funapp.cs.bilkent.edu.tr/DataSets/Data/WI.dat), Bilkent Üniversitesi'nin kaynak araştırması yaparak toparladıkları veri setlerini içeren veri deposundan alınmıştır. Veri seti, 1994 ile 1997 yılları arasında İzmir'deki hava durumu bilgilerini içerir. 1461 gözlem değeri ve 10 değişkenden oluşmaktadır.

Veri seti; %80 eğitim, %20 test seti olacak şekilde rastgele olarak bölünmüştür. Daha sonra, veri setinin %80'ine karşılık gelen 1170 gözlem değeri SKÖ ve modifikasyonları kullanılarak seçilmiştir. $m = 5, r = 234$ olarak alınmıştır. Sonuçlar Tablo 5.5'teki gibidir.

Tablo 5.5 İzmir hava durumu verisine ait ortalama HKOK değerleri

Yöntemler	**Rastgele**	**SKÖ**	**USKÖ**	**MSKÖ**	**YSKÖ (p=0,7)**
ÇDR	**1,259206**	1,259383	1,2601	1,265244	1,267216
TBR	3,830477	3,833836	4,059473	**3,756096**	3,979684
KEKKR	2,00459	1,996648	2,237909	**1,838165**	2,162864
RR	1,259312	1,256209	**1,244379**	1,260241	1,264357
LR	**1,258709**	1,258935	1,260083	1,265209	1,267217
ENR	1,260506	**1,259034**	1,266202	1,260816	1,277498
KNN	**2,626363**	2,645596	2,653257	4,00221	2,864057
DVR	1,967937	1,981489	**1,658036**	3,715895	2,076779
YSA	19,00493	**18,89539**	19,29318	20,59726	19,96765
SRA	1,703819	**1,702664**	1,754533	2,415204	1,749605
TA	1,251318	**1,250481**	1,278896	2,042374	1,358613
RO	1,438293	1,437433	**1,364873**	3,031981	1,818029
GYM	1,95696	1,965608	**1,554424**	4,874149	2,708145
AGY	1,336375	**1,33064**	1,370318	1,933327	1,424818

ÇDR, LR ve KNN algoritmalarında, rastgele olarak bölünen veri seti ile; ENR, YSA, SRA, TA ve AGY algoritmalarında, SKÖ kullanılarak bölünen veri seti ile; RR, DVR, RO ve GYM algoritmalarında, USKÖ ile bölünen veri seti ile; TBR ve KEKKR algoritmalarında ise MSKÖ kullanılarak bölünen veri seti ile daha düşük hata değerleri hesaplanmıştır.

5.2 Sınıflandırma Uygulamaları

Sınıflandırma problemlerinde kullanılan denetimli öğrenme algoritmaları 3 gerçek hayat veri setine uygulanmıştır. Veri setleri; rastgele olarak ve SKÖ, USKÖ, MSKÖ ve YSKÖ yöntemleri kullanılarak eğitim ve test setlerine ayrılmıştır. Kullanılan veri setlerinin özellikleri Tablo 5.6'da verilmiştir.

Tablo 5.6 Sınıflandırma uygulamalarında kullanılan gerçek hayat veri setleri

Veri Seti	Değişken Sayısı	Gözlem Sayısı
Pima Kızılderelileri Diyabet	9	768
İyonosfer	34	351
Denetim Riski	25	775

5.2.1 Gerçek Hayat Veri Uygulaması-1

İlgili veri seti (https://www.kaggle.com/uciml/pima-indians-diabetes-database), veri bilimciler için birçok veri setini içeren Kaggle platformundan alınmıştır. Pima kızıldereli kadınlarının diyabet hastası olup olmadıklarını tahmin etmek için, bazı tanısal ölçüleri içerir. 9 değişken, 768 gözlem değerinden oluşmaktadır.

Veri seti ilk önce; %80 eğitim, %20 test seti olacak şekilde rastgele olarak, daha sonra SKÖ, USKÖ, MSKÖ ve YSKÖ ile bölünmüştür. SKÖ ve modifikasyonları ile veri seti bölünürken; veri setinin %80'ine karşılık gelen 615 gözlem değeri, $m = 3$, $r = 205$ alınarak seçilmiştir. Elde edilen sonuçlar Tablo 5.7'deki gibidir.

Tablo 5.7 Pima kızılderelileri diyabet verisine ait ortalama doğru sınıflandırma oranları

Yöntemler	Rastgele	SKÖ	USKÖ	MSKÖ	YSKÖ (p=0,4)
LRA	0,7748745	0,7759882	**0,7764431**	0,7406641	0,6837229
KNN	0,7320085	0,7321465	0,7321318	0,7379506	**0,7502615**
DVM	**0,7716784**	0,7715634	0,7715752	0,7431124	0,6933203
YSA	**0,7750562**	0,7736458	0,7734614	0,746681	0,691502
SRA	**0,7377935**	0,7373307	0,7371908	0,7151699	0,6689242
TA	0,7512392	0,7517608	**0,7518118**	0,7283216	0,6782261
RO	**0,7625712**	0,7602405	0,7610667	0,7357647	0,6807582
GYM	**0,7573294**	0,755583	0,7549059	0,7110431	0,6266183
AGY	0,7416706	0,7395686	**0,7475268**	0,6833359	0,5623268

DVM, YSA, SRA, RO ve GYM algoritmalarında, rastgele olarak bölünen veri seti ile; LRA, TA ve AGY algoritmalarında, USKÖ ile bölünen veri seti ile; KNN algoritmasında ise YSKÖ kullanılarak bölünen veri seti ile daha yüksek doğru sınıflandırma oranları hesaplanmıştır.

5.2.2 Gerçek Hayat Veri Uygulaması-2

İlgili veri seti (https://archive.ics.uci.edu/ml/datasets/ionosphere), Kaliforniya Üniversitesi'nin oluşturduğu makine öğrenmesi veri seti deposundan alınmıştır. İyonosferde bir yapının olup olmadığını tahmin etmek için, serbest elektronların bazı özelliklerini içerir. 34 değişken, 351 gözlem değerinden oluşur.

Veri seti; %75 eğitim, %25 test seti olacak şekilde rastgele olarak bölünmüştür. Daha sonra, veri setinin %75'ine karşılık gelen 264 gözlem değeri SKÖ ve modifikasyonları kullanılarak seçilmiştir. $m = 4$, $r = 66$ olarak alınmıştır. Sonuçlar Tablo 5.8'deki gibidir.

Tablo 5.8 İyonosfer verisine ait ortalama doğru sınıflandırma oranları

Yöntemler	Rastgele	SKÖ	USKÖ	MSKÖ	YSKÖ (p=0,6)
LRA	0,8666805	0,8651586	**0,8909862**	0,8529724	0,857846
KNN	0,8298025	0,8315196	0,8232241	**0,8359129**	0,8341927
DVM	0,9290391	0,928469	**0,9344713**	0,9204115	0,9226828
YSA	0,897869	0,8989586	**0,9258276**	0,870931	0,8765931
SRA	0,8752023	0,875331	**0,8873655**	0,8667172	0,8659057
TA	0,9174483	0,9181402	**0,9219425**	0,9124483	0,9149517
RO	0,9304598	0,9298345	**0,9349701**	0,9247264	0,9250069
GYM	0,891531	0,890246	**0,9296782**	0,8671195	0,8674161
AGY	0,9278184	0,9276322	**0,9413977**	0,9172138	0,9163172

LRA, DVM, YSA, SRA, TA, RO, GYM ve AGY algoritmalarında, USKÖ ile bölünen veri seti ile; KNN algoritmasında ise MSKÖ kullanılarak bölünen veri seti ile daha yüksek doğru sınıflandırma oranları hesaplanmıştır.

5.2.3 Gerçek Hayat Veri Uygulaması-3

İlgili veri seti (https://archive.ics.uci.edu/ml/datasets/Audit+Data), Kaliforniya Üniversitesi tarafından oluşturulan makine öğrenmesi veri seti deposundan alınmıştır. Farklı sektörlerden firmaların mevcut ve geçmiş risk faktörü özelliklerine göre, dolandırıcı (sahte) firma olup olmadıklarını tahmin edecek bir sınıflandırma modeli kurmak amacıyla oluşturulmuştur. 775 gözlem değeri, 25 değişken içerir.

Veri seti; %76 eğitim, %24 test seti olacak şekilde rastgele olarak bölünmüştür. Daha sonra, veri setinin %76'sına karşılık gelen 590 gözlem değeri SKÖ ve modifikasyonları kullanılarak seçilmiştir. $m = 5, r = 118$ olarak alınmıştır. Sonuçlar Tablo 5.9'daki gibidir.

Tablo 5.9 Denetim riski verisine ait ortalama doğru sınıflandırma oranları

Yöntemler	Rastgele	SKÖ	USKÖ	MSKÖ	YSKÖ (p=0,5)
LRA	0,9638757	0,9637524	**0,968147**	0,9584195	0,9602357
KNN	0,9674351	0,9671478	**0,9680813**	0,9675237	0,9675297
DVM	0,9795168	0,9795935	**0,9808735**	0,9768811	0,9768411
YSA	0,9921935	0,9921946	**0,9936227**	0,9910519	0,9911416
SRA	**0,9996919**	0,9910173	0,9650454	0,9722032	0,9721935
TA	**0,9996551**	0,9995665	0,999627	0,9995005	0,999507
RO	0,9987276	0,9987514	**0,9990249**	0,998333	0,9984054
GYM	0,9990454	0,9990746	**0,9993935**	0,9986854	0,9986811
AGY	0,9853449	0,9851265	**0,990267**	0,9669135	0,9784314

LRA, KNN, DVM, YSA, RO, GYM ve AGY algoritmalarında, USKÖ ile bölünen veri seti ile; SRA ve TA yöntemlerinde ise rastgele olarak bölünen veri seti ile daha yüksek doğru sınıflandırma oranları elde edilmiştir.

BÖLÜM 6
SONUÇLAR

Makine öğrenmesi algoritmalarında, bilgisayara bazı davranışların öğretilmesi ve öğrenme konusunda ne kadar başarılı olduklarının incelenmesi için, çalışılacak veri kümesi eğitim ve test seti olarak ikiye ayrılır. Bu tez çalışmasında, veri seti bölme işlemi; hem literatürdeki gibi rastgele olarak hem de SKÖ, USKÖ, MSKÖ ve YSKÖ yöntemleri ile yapılarak elde edilen sonuçların karşılaştırılması amaçlanmıştır. Bölüm 5'te; gerçek hayat veri setlerinin belirtilen yöntemlerle bölünerek, denetimli öğrenme algoritmaları ile analiz edilmeleri sonucu elde edilen ortalama HKOK değerleri ve doğru sınıflandırma oranları verilmiştir.

TBR, YSA, RO ve AGY algoritmalarında; regresyon uygulamalarında kullanılan bütün veri setlerinde, bölme işlemi SKÖ ve modifikasyonları kullanılarak yapıldığında daha düşük HKOK değerleri elde edilmiştir. Bu sonuçlara göre; TBR, YSA, RO ve AGY algoritmaları ile çalışırken veri seti bölme işleminin SKÖ ve modifikasyonları ile yapılması önerilebilir.

ÇDR, KEKKR, RR, LR, ENR, SRA, TA ve GYM algoritmalarında; regresyon uygulamalarında kullanılan veri setlerinin sadece bir tanesinde, bölme işlemi rastgele olarak yapıldığında daha düşük HKOK değerleri hesaplanmıştır. Diğer veri setleri ise, SKÖ ve modifikasyonları kullanılarak bölündüklerinde daha düşük HKOK değerleri elde edilmiştir. Bu durumda; ÇDR, KEKKR, RR, LR, ENR, SRA, TA ve GYM algoritmaları ile çalışırken, veri seti bölme işlemi rastgele olarak yapılabileceği gibi SKÖ ve modifikasyonlarının da denenmesi tavsiye edilebilir.

KNN ve DVR algoritmalarında; regresyon uygulamalarında kullanılan veri setlerinden ikisinde, bölme işlemi USKÖ yöntemi ile yapıldığında daha düşük HKOK değerleri elde edilmiştir. Diğer veri setlerinde ise; bölme işlemi rastgele olarak yapıldığında daha düşük HKOK değerleri hesaplanmıştır. Elde edilen sonuçlar incelendiğinde; KNN ve DVR algoritmaları ile çalışırken, veri seti bölme işleminin

rastgele olarak yapılması önerilebileceği gibi, USKÖ yönteminin de denenmesi tavsiye edilebilir.

LRA, KNN ve AGY algoritmalarında; sınıflandırma uygulamalarında kullanılan veri setlerinin hepsinde, veri seti bölme işlemi SKÖ ve modifikasyonları kullanılarak yapıldığında daha yüksek doğru sınıflandırma oranları elde edilmiştir. Bu durumda; LRA, KNN ve AGY algoritmaları ile çalışırken, veri seti bölme işleminin SKÖ ve modifikasyonları kullanılarak yapılması tavsiye edilebilir.

DVM, YSA, RO, TA ve GYM algoritmalarında; sınıflandırma uygulamalarında kullanılan veri setlerinin sadece birinde, veri seti bölme işlemi rastgele olarak yapıldığında daha yüksek doğru sınıflandırma oranları hesaplanmıştır. Diğer veri setlerinde ise; bölme işlemi USKÖ yöntemi ile yapıldığında daha yüksek doğru sınıflandırma oranları elde edilmiştir. Bu sonuçlara göre; DVM, YSA, RO, TA ve GYM algoritmaları ile çalışılırken, veri seti bölme işlemi rastgele olarak yapılabileceği gibi, USKÖ yönteminin de denenmesi önerilebilir.

SRA algoritmasında; sınıflandırma uygulamalarında kullanılan veri setlerinden ikisinde, bölme işlemi rastgele olarak yapıldığında daha yüksek doğru sınıflandırma oranları elde edilmiştir. Diğer veri seti ise; USKÖ yöntemi kullanılarak bölündüğünde daha yüksek doğru sınıflandırma oranı hesaplanmıştır. Elde edilen sonuçlara göre; SRA algoritması ile çalışırken, veri seti bölme işleminin rastgele olarak yapılması önerilebileceği gibi, USKÖ yönteminin de denenmesi tavsiye edilebilir.

Bu tez çalışmasında; veri seti bölme işlemi SKÖ ve modifikasyonlarına dayalı olarak yapıldığında, sonuçların kayda değer bir kısmında, literatürde yer alan rastgele oran yöntemine göre, değerlendirme ölçütlerinde iyileştirme sağladığı gözlemlenmiştir. Dolayısıyla, farklı çalışma alanlarına ait verilerin analiz sonuçlarına dayanarak, bu tezde kullanılan algoritmalar ile çalışırken, veri seti bölme işleminde SKÖ, USKÖ, MSKÖ ve YSKÖ yöntemlerinin de göz önünde bulundurulması tavsiye edilir.

KAYNAKLAR

Acı, M., Avcı, M. ve Acı, Ç. (2017). Destek vektör regresyonu yöntemiyle karbon nanotüp benzetim süresinin kısaltılması. *Journal of the Faculty of Engineering and Architecture of Gazi University*, *32*(3), 901-907. https://doi.org/10.17341/gazimmfd.337642

Agresti, A. (2010). *Analysis of ordinal categorical data* (2. Baskı). John Wiley & Sons. https://doi.org/10.1002/9780470594001

Akar, Ö. ve Güngör, O. (2012). Classification of multispectral images using random forest algorithm. *Journal of Geodesy and Geoinformation*, *1*(2), 105-112. https://doi.org/10.9733/jgg.241212.1

Akman, V. ve Blackburn, P. (2000). Alan Turing and artificial intelligence. *Journal of Logic, Language, and Information*, *9*(4), 391-395. http://repository.bilkent.edu.tr/bitstream/handle/11693/48789/Editorial_alan_turing_and_artificial_intelligence.pdf?sequence=1

Aktaş, C. (2009). Lojistik regresyon analizi: Öğrencilerin sigara içme alışkanlığı üzerine bir uygulama. *Erciyes Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, *1*(26), 107-122. https://dergipark.org.tr/en/download/article-file/219475

Anguita, D., Ghio, A., Ridella, S. ve Sterpi, D. (2009, Temmuz). *K-fold cross validation for error rate estimate in support vector machines*. The 5th International Conference on Data Mining. Nevada, USA. http://www.smartlab.ws/files/pubblications/ICP/AngGhiRidSteDMIN09_final.pdf

Arat, M. M. (2014). *Destek vektör makineleri üzerine bir çalışma* [Yüksek Lisans Tezi]. Hacettepe Üniversitesi.

Asilkan, Ö. ve Irmak, S. (2009). İkinci el otomobillerin gelecekteki fiyatlarının yapay sinir ağları ile tahmin edilmesi. *Süleyman Demirel Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, *14*(2), 375-391. https://dergipark.org.tr/en/download/article-file/194673

Atakurt, Y. (1999). Lojistik regresyon analizi ve tıp alanında kullanımına ilişkin bir uygulama. *Ankara Üniversitesi Tıp Fakültesi Mecmuası, 52*(4), 191-199. https://doi.org/10.1501/Tipfak_0000000406

Ay, Ş. (9 Ağustos 2021). *Ensemble learning – bagging ve boosting.* Medium. https://medium.com/deep-learning-turkiye/ensemble-learning-bagging-ve-boosting-50643428b22b

Bai, Z. ve Chen, Z. (2003). On the theory of ranked-set sampling and its ramifications. *Journal of Statistical Planning and Inference*, *109*(1-2), 81-99. https://doi.org/10.1016/S0378-3758(02)00302-6

Batista, G. E. A. P. A. ve Silva, D. F. (2009, Ağustos). *How k-nearest neighbor parameters affect its performance.* In Argentine symposium on artificial intelligence. Mar Del Plata, Argentina. https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.696.2787&rep=rep1&type=pdf

Berry, M. J. ve Linoff, G. S. (2004). *Data mining techniques: for marketing, sales, and customer relationship management* (2. Baskı). Wiley.

Bohn, L. L. ve Wolfe, D. A. (1992). Nonparametric two-sample procedures for ranked-set samples data. *Journal of the American Statistical Association*, *87*(418), 552-561. https://doi.org/10.2307/2290290

Bonaccorso, G. (2017). *Machine learning algorithms.* Packt Publishing. https://balasahebtarle.files.wordpress.com/2020/01/machine-learning-algorithms_text-book.pdf

Breiman, L. (1996). Bagging predictors. *Machine learning*, *24*(2), 123-140. https://doi.org/10.1007/BF00058655

Breiman, L. (1999). Prediction games and arcing algorithms. *Neural computation*, *11*(7), 1493-1517. https://doi.org/10.1162/089976699300016106

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5-32. https://doi.org/10.1023/A:1010933404324

Breiman, L. (2004). *Consistency for a simple model of random forests.* CiteSeerx. https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.618.90&rep=rep1&type=pdf

Breiman, L., Friedman, J. H., Olshen, R. A. ve Stone, C. J. (1984). *Classification and regression trees.* Wadsworth International Group.

Bulut, E. ve Alın, A. (2009). Kısmi en küçük kareler regresyon yöntemi algoritmalarından NIPALS ve PLS-KERNEL algoritmalarının karşılaştırılması ve bir uygulama. *Dokuz Eylül Üniversitesi İktisadi İdari Bilimler Fakültesi Dergisi*, *24*(2), 127-138. https://dergipark.org.tr/tr/download/article-file/211076

Bulut, Y. M. (2011). *Çoklu iç ilişki durumunda kısmi en küçük kareler regresyonu ve alternatif yöntemlerle karşılaştırılması* [Yüksek Lisans Tezi]. Eskişehir Osmangazi Üniversitesi.

Bursa, N. (2019). *Bağımsız bileşenler analizi ile çoklu bağlantı sorununa bir yaklaşım* [Doktora Tezi]. Hacettepe Üniversitesi.

Büyükuysal, M. Ç. (2010). *Ridge regresyon analizi ve bir uygulama* [Yüksek Lisans Tezi]. Uludağ Üniversitesi.

Chang, L. Y. ve Wang, H. W. (2006). Analysis of traffic injury severity: An application of non-parametric classification tree techniques. *Accident Analysis & Prevention*, *38*(5), 1019-1027. https://doi.org/10.1016/j.aap.2006.04.009

Chen, T. ve Guestrin, C. (2016). Xgboost: A scalable tree boosting system. *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining,* 785-794. https://doi.org/10.1145/2939672.2939785

Chen, Z. (1999). Density estimation using ranked-set sampling data. *Environmental and Ecological Statistics*, *6*(2), 135-146. https://doi.org/10.1023/A:1009661919622

Chen, Z. (2001). Ranked-set sampling with regression-type estimators. *Journal of Statistical Planning and Inference*, *92*(1-2), 181-192. https://doi.org/10.1016/S0378-3758(00)00140-3

Chen, Z. ve Shen, L. (2003). Two-layer ranked set sampling with concomitant variables. *Journal of Statistical planning and Inference*, *115*(1), 45-57. https://doi.org/10.1016/S0378-3758(02)00145-3

Chen, Z., Bai, Z. ve Sinha, B. (2003). *Ranked set sampling: theory and applications*. Springer Science & Business Media.

Cihantimur Özkan, S. (2014). *Sıralı küme örneklemesi yöntemi ve ormancılıkta bir uygulama* [Yüksek Lisans Tezi]. Hacettepe Üniversitesi.

Cortes, C. ve Vapnik, V. (1995). Support-vector networks. *Machine learning*, *20*(3), 273-297. https://doi.org/10.1007/BF00994018

Cover, T. ve Hart, P. (1967). Nearest neighbor pattern classification. *IEEE transactions on information theory*, *13*(1), 21-27. https://doi.org/10.1109/TIT.1967.1053964

Çelik, Ö. ve Altunaydın, S. S. (2018). A research on machine learning methods and its applications. *Journal of Educational Technology and Online Learning*, *1*(3), 25-40. https://doi.org/10.31681/jetol.457046

Çiftler, A. F. (1 Eylül 2021). *Veri bilimi notları 1- makine öğrenmesine giriş.* Linkedin. https://www.linkedin.com/pulse/veri-bilimi-notları-1-makine-öğrenmesine-giriş-çiftler/

Çolak, C. (2001). *Lojistik regresyon analizi ve sağlık bilimlerinde bir uygulama* [Yüksek Lisans Tezi]. İnönü Üniversitesi.

Çolak, U. (6 Eylül 2021). *Makine öğrenmesi – model başarı değerlendirme ölçütleri.* Medium. https://ufukcolak.medium.com/makine-öğrenmesi-model-başarı-değerlendirme-ölçütleri-3-26014630d7ce

David, H. A. ve Levine, D. (1972). Ranked set sampling in the presence of judgment error. *Biometrics*, *28*, 553-555.

Dell, T. R. ve Clutter, J. L. (1972). Ranked set sampling theory with order statistics background. *Biometrics*, *28*(2), 545-555. http://doi.org/10.2307/2556166

Demirhan, H. ve Hamurkaroğlu, C. (2015). *İstatistiksel yöntemlere giriş* (2. Baskı). Hacettepe Üniversitesi Yayınları.

Dilki, G. ve Deniz Başar, Ö. (2020). İşletmelerin iflas tahmininde k-en yakın komşu algoritması üzerinden uzaklık ölçütlerinin karşılaştırılması. *İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi*, *19*(38), 224-233. https://dergipark.org.tr/en/download/article-file/1407214

Eray, O. (2008). *Destek vektör makineleri ile ses tanıma uygulaması* [Yüksek Lisans Tezi]. Pamukkale Üniversitesi.

Erdem, E. S. (2014). *Ses sinyallerinde duygu tanıma ve geri erişimi* [Yüksek Lisans Tezi]. Başkent Üniversitesi.

Erpolat, S. ve Öz, E. (2010). Kanser verilerinin sınıflandırılmasında yapay sinir ağları ve destek vektör makinelerinin karşılaştırılması. *İstanbul Aydın Üniversitesi Dergisi*, *2*(5), 71-83. https://dergipark.org.tr/tr/download/article-file/319296

Fırat, M. ve Güngör, M. (2004). Askı madde konsantrasyonu ve miktarının yapay sinir ağları ile belirlenmesi. *Teknik Dergi*, *15*(73), 3267-3282. https://dergipark.org.tr/tr/download/article-file/136710

Filiz, E. (2019). *Makine öğrenmesi yöntemleri ve eğitim verisi üzerine bir uygulama: Uluslararası matematik ve fen eğilimleri araştırması 2015 Türkiye örneği* [Doktora Tezi]. Yıldız Teknik Üniversitesi.

Fonti, V. ve Belitser, E. (2017). Feature selection using lasso. *VU Amsterdam Research Paper in Business Analytics*, *30*, 1-25.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, *29*(5), 1189-1232. https://www.jstor.org/stable/2699986

Galton, F. (1886). Regression towards mediocrity in hereditary stature. *The Journal of the Anthropological Institute of Great Britain and Ireland*, *15*, 246-263. https://doi.org/10.2307/2841583

Geladi, P. ve Kowalski, B. R. (1986). Partial least-squares regression: A tutorial. *Analytica chimica acta*, *185*, 1-17. https://doi.org/10.1016/0003-2670(86)80028-9

Gershenson, C. (2003). *Artificial neural networks for beginners.* arXiv. https://arxiv.org/abs/cs/0308031

Gujarati, D. N. ve Porter, D. C. (2010). *Essentials of econometrics* (4. Baskı). Irwin/McGraw-Hill.

Gunst, R. F. ve Mason, R. L. (1980). *Regression analysis and its application: a data-oriented approach*. CRC Press. https://doi.org/10.1201/9780203741054

Halls, L. K. ve Dell, T. R. (1966). Trial of ranked-set sampling for forage yields. *Forest Science*, *12*(1), 22-26.

Hastie, T., Tibshirani, R. ve Friedman, J. (2009). *The elements of statistical learning: data mining, inference, prediction* (2. Baskı). Springer Verlag.

Hawkins, D. M. (1973). On the investigation of alternative regressions by principal component analysis. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, *22*(3), 275-286. https://doi.org/10.2307/2346776

Hettmansperger, T. P. (1995). The ranked-set sample sign test. *Journal of Nonparametric Statistics*, *4*(3), 263-270. https:/doi.org/10.1080/10485259508832617

Hill, R. C., Fomby, T. B. ve Johnson, S. R. (1977). Component selection norms for principal components regression. *Communications in Statistics-Theory and Methods*, *6*(4), 309-334. https://doi.org/10.1080/03610927708827494

Hill, T., Marquez, L., O'Connor, M. ve Remus, W. (1994). Artificial neural network models for forecasting and decision making. *International journal of forecasting*, *10*(1), 5-15. https://doi.org/10.1016/0169-2070(94)90045-0

Hoerl, A. E. (1962). Application of ridge analysis to regression problems. *Chemical Engineering Progress*, 58, 54-59.

Hoerl, A. E. ve Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, *12*(1), 55-67. https://doi.org/10.2307/1267351

Hosmer, D. W. ve Lemeshow, S. (2000). *Applied lojistic regression* (2. Baskı). John Wiley & Sons. https://doi.org/10.1002/0471722146

Işıkhan, S. Y. (2014). *Mikrodizilim gen ifade çalışmalarında genelleştirme yöntemlerinin regresyon modelleri üzerine etkisi* [Doktora Tezi]. Hacettepe Üniversitesi.

İskit, T. (2018). *Medyan sıralı küme örneklemesi yönteminde kitle ortalamasının tahmini* [Yüksek Lisans Tezi]. Hacettepe Üniversitesi.

Jeffers, J. N. (1967). Two case studies in the application of principal component analysis. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, *16*(3), 225-236. https://doi.org/10.2307/2985919

Jolliffe, I. T. (1982). A note on the use of principal components in regression. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, *31*(3), 300-303. https://doi.org/10.2307/2348005

Karal, Ö. (2018). Compression of ECG data by support vector regression method. *Journal of the Faculty of Engineering and Architecture of Gazi University*, *33*(2), 743-755. https://doi.org/10.17341/gazimmfd.416527

Kendall, M. G. (1957). *A course in multivariate analysis*. Griffin.

Khan, M., Ding, Q. ve Perrizo, W. (2002). K-nearest neighbor classification on spatial data streams using P-trees. *Advances in Knowledge Discovery and Data Mining* içinde (517-528). Springer. https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.93.5092&rep=rep1&type=pdf

Köktürk, F. (2012). *K-en yakın komşuluk, yapay sinir ağları ve karar ağaçları yöntemlerinin sınıflandırma başarılarının karşılaştırılması.* [Yüksek Lisans Tezi]. Bülent Ecevit Üniversitesi.

Kresse, W. ve Danko, D. M. (Ed.). (2012). *Springer handbook of geographic information*. Springer Science & Business Media.

Kukreja, H., Bharath, N., Siddesh, C. S. ve Kuldeep, S. (2016). An introduction to artificial neural network. *International Journal Advance Research Innovative Ideas Education*, *1*(5), 27-30.

Kurt, R., Karayılmazlar, S., İmren, E. ve Çabuk, Y. (2017). Yapay Sinir Ağları ile öngörü modellemesi: Türkiye kağıt-karton sanayi örneği. *Bartın Orman Fakültesi Dergisi*, *19*(2), 99-106. https://dergipark.org.tr/en/download/article-file/354240

Kuyucu, Y. E. (2012). *Lojistik regresyon analizi (LRA), yapay sinir ağları (YSA) ve sınıflandırma ve regresyon ağaçları (C&RT) yöntemlerinin karşılaştırılması ve tıp alanında bir uygulama* [Yüksek Lisans Tezi]. Gaziosmanpaşa Üniversitesi.

Küçük, A. (2019). *Doğrusal regresyonda Ridge, Liu ve LASSO tahmin edicileri üzerine bir çalışma* [Yüksek Lisans Tezi]. Hacettepe Üniversitesi.

Liang, W., Luo, S., Zhao, G. ve Wu, H. (2020). Predicting hard rock pillar stability using GBDT, XGBoost, and LightGBM algorithms. *Mathematics*, *8*(5), 765. https://doi.org/10.3390/math8050765

Lindgren, F. ve Rännar, S. (1998). Alternative partial least-squares (PLS) algorithms. *3D QSAR in Drug Design*, *12*, 105-113. https://doi.org/10.1007/0-306-46858-1_7

Loh, W. Y. (2011). Classification and regression trees. *Wiley interdisciplinary reviews: Data mining and knowledge discovery*, *1*(1), 14-23. https://doi.org/10.1002/widm.8

Lu, H., Karimireddy, S. P., Ponomareva, N. ve Mirrokni, V. (2020). Accelerating gradient boosting machines. *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics, 108*, 516-526.

Mahesh, B. (2018). Machine learning algorithms - a review. *International Journal of Science and Research, 9*(1), 381-386.

Mansfield, E. R., Webster, J. T. ve Gunst, R. F. (1977). An analytic variable selection technique for principal component regression. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, *26*(1), 34-40. https://doi.org/10.2307/2346865

Martinasek, Z., Zeman, V., Malina, L. ve Martinasek, J. (2016). K-Nearest neighbors algorithm in profiling power analysis attacks. *Radioengineering*, *25*(2), 365-382. https://doi.org/10.13164/re.2016.0365

McIntyre, G. A. (1952). A method for unbiased selective sampling, using ranked sets. *Australian Journal of Agricultural Research*, *3*(4), 385-390. https://dx.doi.org/10.1071/AR9520385

Mevik, B. H. ve Wehrens, R. (2007). The pls package: Principal component and partial least regression in R. *Journal of Statistical Software, 18*(2), 1-23. https://doi.org/10.18637/jss.v018.i02

Mo, H., Sun, H., Liu, J. ve Wei, S. (2019). Developing window behavior models for residential buildings using XGBoost algorithm. *Energy and Buildings*, *205*, 109564. https://doi.org/10.1016/j.enbuild.2019.109564

Molnar, C. (2020). *Interpretable machine learning.* Lean Publishing. https://christophm.github.io/interpretable-ml-book/

Montgomery, D. C., Peck, E. A., ve Vining, G. G. (2013). *Introduction to linear regression analysis* (5. Baskı). John Wiley & Sons.

Morgan, J. N. ve Messenger, R. C. (1973). *THAID, a sequential analysis program for the analysis of nominal scale dependent variables.* Institute for Social Research, University of Michigan.

Mosteller, F. ve Tukey, J. W. (1977). *Data analysis and regression: A second course in statistics*. Reading, Mass: Addison-Wesley Pub. Co.

Muratlar, E. R. (27 Ağustos 2021). *Gradient boosted regresyon ağaçları.* VBO. https://www.veribilimiokulu.com/gradient-boosted-regresyon-agaclari/

Muttlak, H. A. (1997). Median ranked set sampling. *Journal of Applied Statistical Sciences, 6*(4)*,* 245-255.

Muttlak, H. A. (2003). Modified ranked set sampling methods. *Pakistan Journal of Statistics, 19*(3), 315-323.

Özkan, Y. (2019). *Hastalık tanısı verilerinde veri ön işlemenin topluluk öğrenme sınıflandırma algoritmaları üzerindeki etkisinin incelenmesi* [Yüksek Lisans Tezi]. Ege Üniversitesi.

Öztemel, E. (2003). *Yapay sinir ağları* (3. Baskı). Papatya Yayıncılık. http://papatyabilim.com.tr/PDF/yapay_sinir_aglari.pdf

Öztürk, Ö. (2018). Ratio estimators based on a ranked set sample in a finite population setting. *Journal of the Korean Statistical Society*, *47*(2), 226-238. https://doi.org/10.1016/j.jkss.2018.02.001

Pan, B. (2018). Application of XGBoost algorithm in hourly PM2. 5 concentration prediction. *IOP conference series: earth and environmental science*, *113*(1), 012127. https://doi.org/10.1088/1755-1315/113/1/012127

Parr, T. ve Howard, J. (19 Ağustos 2021). *How to explain gradient boosting.* explained.ai. https://explained.ai/gradient-boosting/index.html

Parsa, A. B., Movahedi, A., Taghipour, H., Derrible, S. ve Mohammadian, A. K. (2020). Toward safer highways, application of XGBoost and SHAP for real-time accident detection and feature analysis. *Accident Analysis & Prevention*, *136*, 105405. https://doi.org/10.1016/j.aap.2019.105405

Patil, G. P., Sinha, A. K. ve Taillie, C. (1999). Ranked set sampling: a bibliography. *Environmental and Ecological Statistics*, *6*(1), 91–98.

Patil, G. P., Sinha, A. K. ve Taille, C. (1993). Relative precision of ranked set sampling: A comparison with the regression estimator. *Environmetrics, 4*(4), 399-412. https://doi.org/10.1002/env.3170040404

Pehlivan, G. (2006). *CHAID analizi ve bir uygulama* [Yüksek Lisans Tezi]. Yıldız Teknik Üniversitesi.

Polat, E. (2009). *Kısmi en küçük kareler regresyonu* [Yüksek Lisans Tezi]. Hacettepe Üniversitesi.

Presnell, B. ve Bohn, L. L. (1999). U-statistics and imperfect ranking in ranked set sampling. *Journal of Nonparametric Statistics*, *10*(2), 111-126. https://doi.org/10.1080/10485259908832756

Rawlings, J. O., Pantula, S. G. ve Dickey, D. A. (2001). *Applied regression analysis: A research tool* (2. Baskı). Springer Science & Business Media. http://web.nchu.edu.tw/~numerical/course1012/ra/Applied_Regression_Analysis_A_Research_Tool.pdf

Sakallıoğlu, S. ve Kaçıranlar, S. (2008). A new biased estimator based on ridge estimation. *Statistical Papers*, *49*(4), 669-689. https://doi.org/10.1007/s00362-006-0037-0

Samawi, H. M. ve Muttlak, H. A. (1996). Estimation of ratio using rank set sampling. *Biometrical Journal*, *38*(6), 753-764. https:/doi.org/10.1002/bimj.4710380616

Samawi, H. M., Ahmed, M. S. ve Abu-Dayyeh, W. (1996). Estimating the population mean using extreme ranked set sampling. *Biometrical Journal*, *38*(5), 577-586. https://doi.org/10.1002/bimj.4710380506

Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, *3*(3), 210-229. https://doi.org/10.1147/rd.33.0210

Schapire, R. E. (1990). The strength of weak learnability. *Machine learning*, *5*(2), 197-227. https://doi.org/10.1007/BF00116037

Sezgin, Ö. (2006). *Statistical methods in credit rating* [Yüksek Lisans Tezi]. Orta Doğu Teknik Üniversitesi.

Singh, B., Sihag, P. ve Singh, K. (2017). Modelling of impact of water quality on infiltration rate of soil by random forest regression. *Modeling Earth Systems and Environment*, *3*(3), 999-1004. https://doi.org/10.1007/s40808-017-0347-3

Smola, A. J. ve Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and computing*, *14*(3), 199-222. https://doi.org/10.1023/B:STCO.0000035301.49549.88

Stokes, S. L. (1977). Ranked set sampling with concomitant variables. *Communications in Statistics-Theory and Methods*, *6*(12), 1207–1211. https://doi.org/10.1080/03610927708827563

Stokes, S. L. (1980). Estimation of variance using judgment ordered ranked set samples. *Biometrics*, *36*(1), 35–42. https://doi.org/10.2307/2530493

Stokes, S. L. ve Sager, T. W. (1988). Characterization of a ranked-set sample with application to estimating distribution functions. *Journal of the American Statistical Association*, *83*(402), 374-381. https://doi.org/10.2307/2288852

Suchetana, B., Rajagopalan, B. ve Silverstein, J. (2017). Assessment of wastewater treatment facility compliance with decreasing ammonia discharge limits using a regression tree model. *Science of the total environment*, *598*, 249-257. https://doi.org/10.1016/j.scitotenv.2017.03.236

Takahasi, K. ve Wakimoto, K. (1968). On unbiased estimates of the population mean based on the sample stratified by means of ordering. *Annals of the institute of statistical mathematics*, *20*(1), 1-31. https://www.ism.ac.jp/editsec/aism/pdf/020_1_0001.pdf

Taş, B. (1 Eylül 2021). *Makine öğrenmesi modelleri için veri bölme işlemi.* Medium. https://medium.com/kodluyoruz/makine-öğrenmesi-modelleri-için-veri-bölme-işlemi-3b517ed74e37

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, *58*(1), 267-288. https://doi.org/10.1111/j.2517-6161.1996.tb02080.x

Timofeev, R. (2004). *Classification and regression trees (CART) theory and applications* [Yüksek Lisans Tezi]. Humboldt University of Berlin.

Touzani, S., Granderson, J. ve Fernandes, S. (2018). Gradient boosting machine for modeling the energy consumption of commercial buildings. *Energy and Buildings*, *158*, 1533-1543. https://doi.org/10.1016/j.enbuild.2017.11039

Tunç, Z. (2018). *En küçük kareler ve temel bileşenler regresyon analizlerinin karşılaştırılması* [Yüksek Lisans Tezi]. İnönü Üniversitesi.

Turing, A. M. (1950). Computing machinery and intelligence. *Mind A Quarterly Review of Psychology and Philosophy, 236,* 433-460. https://doi.org/10.1007/978-1-4020-6710-5_3

Wold, H. (1975). Soft modelling by latent variables: The non-linear iterative partial least squares (NIPALS) approach. *Journal of Applied Probability*, *12*(S1), 117-142. https://doi.org/10.1017/S0021900200047604

Wolfe, D. A. (2012). Ranked set sampling: Its relevance and impact on statistical inference. *ISRN Probability and Statistics*, *2012*, 1–32. https://doi.org/10.5402/2012/568385

Yaman, A. ve Cengiz, M. A. (2021). LASSO Estimator in Logistic Regression for Small Data Sets. *Veri Bilimi*, *4*(1), 69-72. https://dergipark.org.tr/en/download/article-file/1357140

Yavuz, A. ve Vupa Çilengiroğlu, Ö. (2020). Lojistik regresyon ve CART yöntemlerinin tahmin edici performanslarının yaşam memnuniyeti verileri için karşılaştırılması. *Avrupa Bilim ve Teknoloji Dergisi*, *18*, 719-727. https://doi.org/10.31590/ejosat.691215

Yeom, S., Giacomelli, I., Fredrikson, M. ve Jha, S. (2018, Temmuz). *Privacy risk in machine learning: Analyzing the connection to overfitting*. IEEE 31st Computer Security Foundations Symposium. Oxford, UK. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8429311

Yu, P. L. H. ve Lam, K. (1997). Regression estimator in ranked set sampling. *Biometrics, 53*(3), 1070-1080. https://doi.org/10.2307/2533564

Zamanzade, E. ve Al-Omari, A. I. (2016). New ranked set sampling for estimating the population mean and variance. *Hacettepe Journal of Mathematics and Statistics, 45*(6), 1891-1905. https://doi.org/10.15672/HJMS.20159213166

Zamanzade, E. ve Vock, M. (2015). Variance estimation in ranked set sampling using a concomitant variable. *Statistics & Probability Letters*, *105*, 1-5. https://doi.org/10.1016/j.spl.2015.04.034

Zhang, Z., Liu, T. ve Zhang, B. (2016). Jackknife empirical likelihood inferences for the population mean with ranked set samples. *Statistics & Probability Letters*, *108*, 16-22. https://doi.org/10.1016/j.spl.2015.09.016

Zhou, Z. H. (2009). Ensemble learning. *Encyclopedia of biometrics*, *1*, 270-273. https://doi.org/10.1007/978-0-387-73003-5_293

Zou, H. ve Hastie, T. (2005). Addendum: Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B*, *67*(5), 301-320. https://doi.org/10.1111/j.1467-9868.2005.00503.x