
GAZİ ÜNİVERSİTESİ
Gazi
1926

KOLEKTİF ÖĞRENMEDE ALT KÜME BÖLÜMLENDİRMESİNE YENİLİKÇİ BİR YAKLAŞIM

Meerim KUBATOVA

YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANA BİLİM DALI

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

MART 2025

ETİK BEYAN

Gazi Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmasında;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmasında yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Meerim KUBATOVA
20/03/2025

KOLEKTİF ÖĞRENMEDE ALT KÜME BÖLÜMLENDİRMESİNE YENİLİKÇİ BİR YAKLAŞIM

(Yüksek Lisans Tezi)

Meerim KUBATOVA

GAZİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

Mart 2025

ÖZET

Makine öğrenmesi, son yıllarda artan veri hacmiyle birlikte sınıflandırma, regresyon ve öngörülemeyen desenleri keşfetme gibi görevlerde yaygın olarak kullanılmaktadır. Ancak tekil öğrenme algoritmaları, genellikle sınırlı genelleme yetenekleri ve belirli veri yapılarındaki performans düşüklüğü nedeniyle yetersiz kalabilmektedir. Bu kapsamda, bireysel modellerin zayıflıklarını dengeleyerek daha sağlam kararlar üretmeyi amaçlayan kolektif öğrenme yaklaşımları öne çıkmaktadır. Bu tezde, kolektif öğrenmenin etkinliğini artırmaya yönelik olarak literatürde yaygın biçimde kullanılan örnek tabanlı alt kümeleme yöntemlerine alternatif bir çözüm olarak, öznitelik tabanlı alt kümeleme temelli yeni bir yöntem önerilmiştir. Önerilen yaklaşımda, veri kümesi özniteliklerine göre parçalara ayrılarak, her bir alt küme üzerinde farklı öğrenme algoritmaları eğitilmiş ve sonrasında bu modeller kolektif bir yapı içinde birleştirilmiştir. Böylece her model farklı bilgi alanlarına odaklanmış, sistemin çeşitliliği artırılmış ve genel performansı iyileştirilmiştir. Yöntem, iki aşamalı bir yapıda kurgulanmış; ilk aşamada dikey alt kümeleme yoluyla ön işleme yapılmış, ikinci aşamada ise heterojen sınıflayıcılardan oluşan kolektif model eğitilmiştir. Deneysel çalışmalar sonucunda, önerilen yöntemin özellikle F1-skoru ve zaman karmaşıklığı açısından klasik kolektif yöntemlere göre üstün performans gösterdiği gözlemlenmiştir. Çalışma, kolektif öğrenme sistemlerine yeni bir bakış açısı kazandırmakta ve disiplinler arası uygulamalarda değerlendirilebilecek güçlü bir yöntem ortaya koymaktadır.

Bilim Kodu	:	92432
Anahtar Kelimeler	:	Akıllı sistemler, kolektif öğrenimi, dikey/yatay alt kümeleme, heterojen/homojen öğrenme algoritmaları
Sayfa Adedi	:	47
Danışman	:	Prof. Dr. Aydın ÇETİN

AN INNOVATIVE APPROACH TO SUBSET PARTITIONING IN ENSEMBLE LEARNING

(M. Sc. Thesis)

Meerim KUBATOVA

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

March 2025

ABSTRACT

Machine learning has been widely adopted in recent years for tasks such as classification, regression, and the discovery of complex, unforeseen patterns, especially due to the rapid increase in data volume. However, individual learning algorithms often fall short in terms of generalization capability and performance across diverse data structures. To address these limitations, ensemble learning approaches have gained prominence by combining multiple models to produce more robust and accurate predictions. This thesis proposes a novel ensemble learning method aimed at improving performance by introducing an alternative to the commonly used sample-based sub-setting techniques in the literature. Instead, it utilizes a feature-based subsetting approach, in which the dataset is partitioned according to feature groups. Separate learning algorithms are then trained on each subset and later integrated into a collective model. This strategy allows each base learner to specialize in different feature spaces, thereby increasing model diversity and overall predictive performance. The proposed method is structured in two stages: a preprocessing phase involving vertical subsetting, followed by a training phase utilizing heterogeneous classifiers within an ensemble framework. Experimental results demonstrate that the proposed approach outperforms traditional ensemble techniques, particularly in terms of F1-score and time complexity. The findings suggest that the method provides a new perspective to ensemble learning and introduces a powerful and adaptable approach that can be evaluated across a wide range of interdisciplinary smart system applications.

Science Code	:	92432
Key Words	:	Intelligent systems, ensemble learning, vertical/horizontal subsetting, heterogeneous/homogeneous learning algorithms
Page Number	:	47
Supervisor	:	Prof. Dr. Aydın ÇETİN

TEŞEKKÜR

Bu yüksek lisans tezi, yoğun ve titiz bir araştırma sürecinin sonucu olarak ortaya çıkmış olup, akademik gelişimime ve araştırma yetkinliklerime önemli katkılar sağlamıştır. Bu süreçte bana rehberlik eden ve değerli bilgileriyle yol gösteren danışman hocam Prof. Dr. Aydın ÇETİN'e en içten teşekkürlerimi sunarım. Kendisinin destekleyici yaklaşımı, teşvik edici geri bildirimleri ve bilimsel perspektifi, çalışmamın her aşamasında bana yön vermiş ve süreci daha anlamlı kılmıştır.

Ayrıca, çalışmamın geliştirilmesine katkıda bulunan, önerileriyle araştırmamın kapsamını genişletmeme yardımcı olan ve akademik bakış açımı zenginleştiren değerli jüri üyeleri, Prof. Dr. Baha ŞEN'e ve Prof. Dr. Yusuf SÖNMEZ'e teşekkür ederim. Bilimsel tartışmalar ve eleştiriler, çalışmamın daha sağlam bir temele oturmasına büyük katkı sağlamıştır.

Süreç boyunca birlikte çalıştığım, bilgi ve deneyimlerini paylaşarak akademik gelişimime katkı sağlayan kıymetli arkadaşım Bora IŞIKOĞLU'na da teşekkür ederim.

Son olarak, en büyük teşekkürü aileme borçluyum. Bu süreçte bana verdikleri maddi ve manevi destek, sabırları ve teşvikleri sayesinde karşılaştığım zorlukları aşabildim. Onların varlığı, akademik hedeflerime ulaşmamda en büyük güç kaynağım olmuştur.

Bu çalışmanın, akademik yolculuğumda önemli bir adım olduğuna inanıyor ve bu süreçte katkıda bulunan herkese bir kez daha içten teşekkürlerimi sunuyorum.

İÇİNDEKİLER

Sayfa

ÖZET iv
ABSTRACT v
TEŞEKKÜR vi
İÇİNDEKİLER vii
ÇİZELGELERİN LİSTESİ viii
ŞEKİLLERİN LİSTESİ ix
SİMGELER VE KISALTMALAR x
1. GİRİŞ 1
2. LİTERATÜR TARAMASI 3
2.1. Mevcut Yaklaşımlar 3
2.2. Kolektif Öğrenmede Alt Kümelendirme 7
2.2.1. Literatürde kullanılan standart veri alt kümeleme teknikleri 7
3. METODOLOJİ 13
3.1. Kolektif Öğrenme Fayda Analizi 14
3.2. Donanım, Geliştirme Ortamı ve Araçların Belirlenmesi 15
3.3. Veri Setinin Belirlenmesi ve Hazırlanması 19
3.4. Önerilen Öznitelik Tabanlı Veri Alt Kümeleme Tekniği 22
3.4.1. Öznitelik mühendisliği adımlarının uygulanması 26
3.5. Meta Modelin Eğitimi ve Mevcut Yaklaşımlarla Kıyaslama Senaryoları 27
4. BULGULAR VE DEĞERLENDİRME 31
5. SONUÇ VE ÖNERİLER 41
KAYNAKLAR 43
ÖZGEÇMİŞ 47

ÇİZELGELERİN LİSTESİ

Çizelge **Sayfa**

Çizelge 3.1. Kullanılan donanım, araç, kütüphane ve algoritmalar..... 16

Çizelge 3.2. Wine Quality-Red veri kümesinin hedef değişkeninin sınıf dağılımı 19

Çizelge 3.3. Şarap kalitesi-kırmızı veri setinin içeriği..... 20

Çizelge 3.4. Heart Disease veri kümesinin hedef değişkeninin sınıfı dağılımı 21

Çizelge 3.5. Kalp Hastalığı Veri Setinin İçeriği 21

Çizelge 3.7. Bootstrap yöntemi ve dikey bölümleme yöntemi veri bölümleme karşılaştırması 25

Çizelge 3.8. Veri senaryoları 27

Çizelge 3.9. Uygulanan Test Senaryoları 29

Çizelge 4.1. Sert oylama ve XGBoost ile önerilen dikey oylama yönteminin karşılaştırılmasının kesinlik sonuçları 31

Çizelge 4.2. Sert oylama ve XGBoost ile önerilen dikey oylama yönteminin karşılaştırılmasının duyarlılık sonuçları 32

Çizelge 4.3. Sert oylama ve XGBoost ile önerilen dikey oylama yönteminin karşılaştırılmasının f1-skor sonuçları 33

Çizelge 4.4. Bootstrap Aggregation yönteminin önerilen dikey oylama yöntemiyle karşılaştırılmasının deneysel sonuçları 38

ŞEKİLLERİN LİSTESİ

Şekil **Sayfa**

Şekil 2.1. Kolektif öğrenmenin literatürdeki en yaygın yaklaşımları 6

Şekil 2.2. Örnek tabanlı veri bölme (yatay alt kümeleme) 11

Şekil 3.1. Geliştirme sürecinin akış şeması 14

Şekil 3.2. Kolektif öğrenmenin fayda analizi 15

Şekil 3.3. Çalışmada kullanılan araçlar..... 17

Şekil 3.4. Öznitelik tabanlı veri bölme (dikey alt kümeleme) 23

Şekil 4.1. Şarap Kalitesi-Kırmızı veri seti kullanılarak sert oylama ve XGBoost'un önerilen dikey oylama yöntemiyle karşılaştırılması sonuçlarını açıklayan grafik 34

Şekil 4.2. SVC tekil modelinin farklı veri senaryolardaki karmaşıklık matrisi..... 35

Şekil 4.3. Heterojen (SVC+LR+RF) kolektif modelinin farklı veri senaryolardaki karmaşıklık matrisi 36

Şekil 4.4. Önerilen dikey alt kümeleme yönteminin heterojen (SVC+LR+RF) kolektif modelinin birleştirilmiş veri senaryosundaki karmaşıklık matrisi ... 37

Şekil 4.5. Bootstrap Aggregation yönteminin Şarap Kalitesi-Kırmızı kullanılarak önerilen dikey oylama yöntemiyle karşılaştırılması sonuçlarını açıklayan grafik 39

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler	**Açıklamalar**
$\mathbf{F_1 - Skoru}$	Duyarlılık ve keskinliğin harmonik ortalaması
Precision	Kesinlik
Recall	Duyarlılık

Kısaltmalar	**Açıklamalar**
CPU	Merkezi İşlem Birimi
FN	Yanlış Negatif
FP	Yanlış Pozitif
GBDT	Gradyan Artırmalı Karar Ağaçları
LR	Lojistik Regresyon Sınıflandırıcısı
MCS	Çoklu Sınıflandırma Sistemleri
ML	Makine Öğrenmesi
PCA	Temel Bileşenler Analizi
RAM	Rastgele Erişimli Bellek
RF	Rastgele Orman Sınıflandırıcısısı
SVC	Destek Vektör Sınıflandırıcısı
TN	Doğru Negatif
TP	Doğru Pozitif
TruncatedSVD	Truncated Singular Value Decomposition
UCI	University of California Irvine ML Repository
XGBoost	Aşırı Gradyan Artırma
YZ	Yapay Zekâ

1. GİRİŞ

Makine öğrenimi, sürekli olarak gelişen bir alan olarak, tahmine dayalı model performansını artırma çabasıyla pek çok yenilikçi metodolojiyi gündeme getirmiştir. Bu çabaların başında, bireysel öğrenme algoritmalarının güçlü yönlerinden yararlanarak daha sağlam ve doğru tahminler yapmayı amaçlayan kolektif öğrenme (ensemble learning) yer almaktadır. Kolektif öğrenme, temel modellerin bir araya getirilerek daha güçlü bir model oluşturulması fikrini benimser. Bireysel modellerin sınırlamaları aşılmakta ve bunun sonucunda gelişmiş genelleme yeteneği ve tahmin doğruluğu sağlanmaktadır. Dietterich (2000) tarafından vurgulandığı gibi, kolektif yöntemleri, tek bir modelin yetersiz kalabileceği karmaşık problemler için çözümler sunarak geniş bir başarı alanı yaratmıştır [1]. Ancak, kolektif öğrenmenin temel zorlukları arasında, modellerin doğru bir şekilde birleştirilmesi ve verilerin etkili şekilde bölümlendirilmesi bulunmaktadır. Bu tezde, kolektif öğrenme içinde veri bölümlendirme sürecine yönelik yeni bir yaklaşım önerilmektedir.

Kolektif öğrenme ve alt küme bölümlendirmesi, makine öğrenmesinde büyük bir öneme sahiptir. Özellikle, kolektif öğrenmenin güçlü yönlerinden faydalanarak, tek bir modelin hatalarını minimize etmek ve model çeşitliliğini artırmak amaçlanmaktadır. Bununla birlikte, kolektif öğrenme yöntemlerinde veri bölümlendirme ve alt kümeleme süreci, modellerin başarısını doğrudan etkileyen kritik bir aşamadır. Mevcut yaklaşımlar, örneğin Bagging ve Boosting gibi tekniklerle bu süreci optimize etmeye çalışmaktadır. Breiman'ın rastgele orman algoritması (Random Forest) [2], Boosting ve AdaBoost gibi yöntemler [3, 4], kolektif öğrenmenin temellerini atmış ve pek çok uygulama alanında başarılı sonuçlar elde edilmiştir. Bununla birlikte, mevcut tekniklerin veri alt kümelerinin yapılandırılması konusunda daha fazla iyileştirmeye ihtiyaç duyduğu, deneysel çalışmalarla gösterilmiştir.

Bu tezde, kolektif öğrenmede veri bölümlendirme sürecini iyileştirmek amacıyla "dikey alt kümeleme" adı verilen yeni bir yaklaşım sunulmaktadır. Bu yenilikçi yöntem, geleneksel kolektif öğrenimi yöntemlerinden daha iyi sonuçlar elde edilmesini sağlamaktadır. Yürütülen deneysel testler, önerilen yöntemin geleneksel yaklaşımlardan daha yüksek doğruluklar sunduğunu göstermektedir ve bu bulgular Bölüm 4'te detaylı olarak incelenmiştir.

Bu tezin sonraki bölümleri şu şekilde yapılandırılmıştır: Bölüm 2'de kolektif öğrenme yöntemlerinden Bagging, Boosting, Stacking ve Blending gibi öne çıkan teknikler ele alınmıştır ve her birinin benzersiz özellikleri ile uygulamaları tartışılmıştır. Bölüm 2.2'de, veri bölümlendirme süreçleri ve mevcut kolektif öğreniminde kullanılan yaklaşımlar üzerinde durarak, bu alandaki temel sorunlara çözüm önerileri sunmak üzere "dikey alt kümeleme" yönteminin teorik temelleri ve uygulaması açıklanmıştır. Yöntemin geliştirme süreci, kullanılan araçları ve veri kümelerini Bölüm 3'te detaylandırmıştır. Bölüm 4'te, yapılan deneylerin sonuçları sunulmuş, önerilen yöntem ile geleneksel yöntemlerin karşılaştırmalı analizini gerçekleştirmiştir. Son olarak, Bölüm 5'te, çalışmanın sonucu ve öneriler hakkında bilgiler verilmiştir.

2. LİTERATÜR TARAMASI

Kolektif öğrenme yöntemleri, makine öğrenmedeki birçok zorluğun çözümünde öncü yaklaşımları temsil etmektedir. Bu yaklaşımlar, birden fazla modeli kullanarak ve onların tahminlerini birleştirerek, bireysel modellerin tahmin doğruluğunu artırır. Bu bölümde, kolektif öğrenimin genel bir bakışı sunulmuştur, geleneksel ve yenilikçi kolektif yöntemleri incelenmiştir ve bu alandaki güncel zorluklar ve gelişen yönelim keşfedilmiştir.

Kolektif yöntemler, artık makine öğrenimi görevlerinin geniş bir yelpazesinde önde gelen bir yaklaşım olarak kabul edilmektedir. Bu kabul, UCI'den alınan 121 veri seti üzerinde gerçekleştirilen ve 17 aileden 179 sınıflayıcıyı içeren kapsamlı bir değerlendirmeden kaynaklanmaktadır [5].

Son yıllarda, kolektif öğrenme üzerine yapılan çok sayıda kapsamlı inceleme akademik literatürde yer almıştır. Veri akışı sınıflandırması için kolektif temelli yöntemler, üstün performansları, kolay uygulama süreçleri ve dinamik güncellemelere uyum sağlama yetenekleri ile tanınmaktadır. Bu alandaki en kapsamlı çalışmalar arasında yer alan Gomes ve arkadaşlarının (2017) çalışması, 60'tan fazla kolektif öğrenme algoritmasını analiz ederek detaylı bir taksonomi sunmaktadır. Bu taksonomi; algoritmaların kullandığı kombinasyon stratejileri (örneğin oylama, ağırlıklı ortalama), çeşitlilik ölçütleri (örneğin hata korelasyonu, doğruluk çeşitliliği), yaygın kullanılan açık kaynak yazılımlar (özellikle MOA, scikit-multiflow gibi) ve halen çözülmeyi bekleyen araştırma zorlukları (örneğin sınıf dengesizliği, gerçek zamanlı uyarlama) gibi temel başlıkları kapsamlı bir şekilde ele almaktadır [6].

2.1. Mevcut Yaklaşımlar

Random Forest, veri madenciliği uygulamalarında güçlü bir kolektif denetimli makine öğrenimi tekniği olarak, Bagging ve Boosting yöntemleriyle karşılaştırılabilir performansı ile gelecekteki bir sınıflayıcı olarak büyük bir potansiyele sahiptir. Bu çalışma, Random Forest üzerine mevcut araştırmaları sistematik bir şekilde gözden geçirmekte. Random Forest sınıflayıcısının taksonomisini, mevcut sınıflayıcıların karşılaştırma çizelgesini sunmakta ve doğruluk ile performansı artırmaya yönelik fırsatları belirlemektedir. Ayrıca akış verisi, dengesiz veri sınıflandırması ve yarı denetimli öğrenme için yenilikçi fikirleri

keşfetmektedir, böylece makine öğrenimi alanındaki gelecekteki araştırmalar için değerli içgörüler sunmaktadır [7].

Karar ağaçları, sınıflandırma, regresyon ve hayatta kalma analizi görevlerindeki tahmin yetenekleri ile tanınır ve veri biliminde önemli bir yer edinmiştir; bu çalışma, tekil karar ağaçlarının bir geliştirmesi olarak karar ormanlarını tanıtmakta, bunların oluşturulmasını, değerini ve üretim, çıktı birleştirme ve orman inceltme yöntemlerini açıklamaktadır [8].

Desen sınıflandırma araştırmalarında, geleneksel tekil sınıflayıcı yaklaşımlarının sınırlamalarını aşmak amacıyla farklı modeller ve veri setleri kullanılarak oluşturulan çoklu sınıflayıcı sistemlerine olan ilgi artmaktadır. Bu çalışma, Hibrid Akıllı Sistemler çerçevesinde çoklu sınıflayıcı sistemleri (MCS) üzerine güncel bir inceleme sunmakta, çeşitlilik ve karar birleştirme yöntemleri gibi temel konuları ele almakta ve gelişen bir dizi uygulama hakkında içgörüler sağlamaktadır [9].

Ancak, yukarıda bahsedilen her bir inceleme [6-9], kolektif yöntemlerin belirli alt alanlarına odaklanmışken, hiçbiri tüm kolektif tekniklerin kapsamlı bir genel görünümünü sunmamaktadır. Kolektif modellerin geniş bir bakış açısı sağlayan diğer önemli çalışmalara aşağıda yer verilmiştir.

Kolektif temelli sistemlerin, yani çoklu sınıflayıcı sistemleri veya sınıflayıcılar komitelerinin keşfi, bireysel görüşlerin birleştirilmesiyle otomatik karar verme süreçlerini geliştirme yetenekleri nedeniyle dikkat çekmiştir. Polikar'ın çalışmasında [10], kolektif sistemlerin tekil sınıflayıcılara göre tercih edilmesini sağlayan koşullar, kolektif bileşenler üretmek için kullanılan algoritmalar, farklı birleştirme prosedürleri, popüler kolektif algoritmalar (örneğin Bagging, Boosting, ve AdaBoost) ve yaygın birleştirme kuralları incelenmekte ve artan öğrenme, veri füzyonu, öznitelik seçimi, güven tahmini ve hata düzeltme çıktı kodları gibi uygulamalar için mevcut ve gelecekteki araştırma yönleri tartışılmaktadır.

Birden fazla modeli kullanarak performansı artıran kolektif yöntemler, hesaplamalı zekâ, istatistik ve makine öğrenmesi gibi çeşitli araştırma alanlarında uygulanmaktadır. Ren ve arkadaşlarının çalışması [11], hem geleneksel kolektif teknikleri (ör. Bagging, Boosting) hem de derin öğrenmeye dayalı modern yaklaşımları inceleyen kapsamlı bir derleme

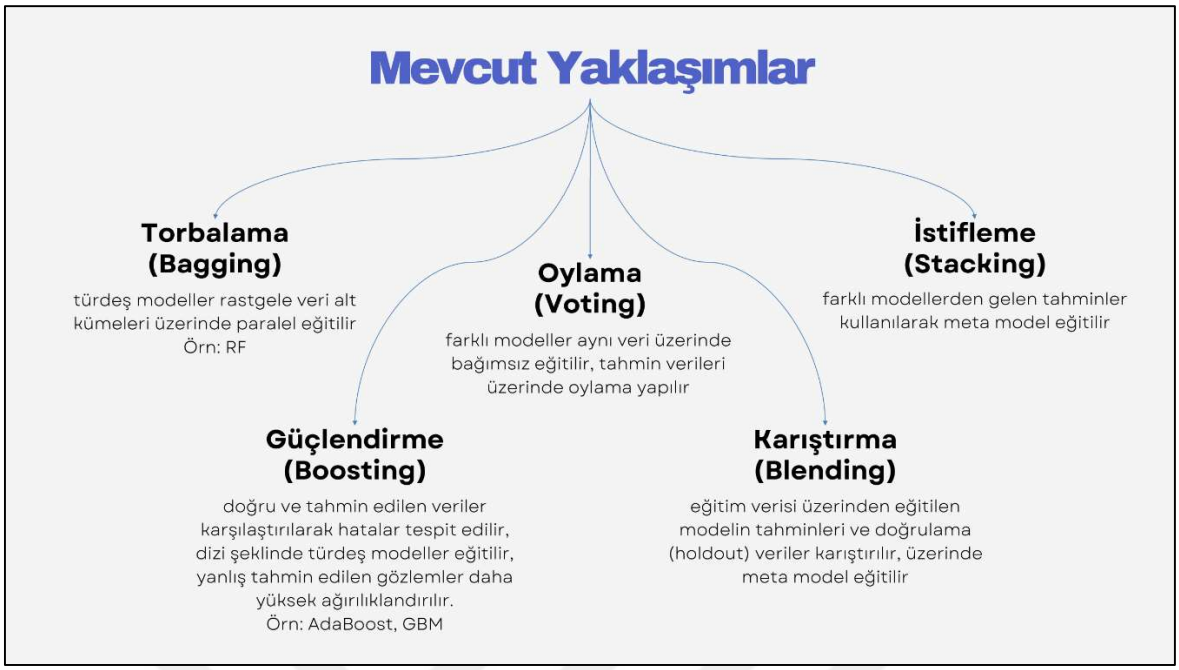
sunmaktadır. Çalışmada; bu yöntemlerin varyasyonları, performans iyileştirmeleri, tipik uygulama alanları ve gelecek araştırmalar için öneriler detaylı biçimde ele alınmaktadır.

Literatürde, kolektif metodolojinin etkinliği geniş bir şekilde kabul edilmektedir. Bu metodoloji, birden fazla modeli birleştirerek tahmin performansını artırır ve istatistik ve yapay zekâ gibi farklı disiplinlerden ilgi görmektedir. Rokach'ın çalışması [12], mevcut kolektif teknikleri gözden geçirmekte ve kolektif temelli sistemler geliştirmeyi amaçlayanlar için bir kılavuz sunmaktadır.

Ancak, bu çalışmalar [10-12], yakın zamanda sunulan ve dikkat çeken bazı önemli gelişmeleri ve bulguları içermemektedir.

Son yıllarda yayımlanan çalışmalar arasında, kolektif öğrenme alanına yön veren kapsamlı bir derleme Sagi ve Rokach tarafından sunulmuştur [13]. Bu çalışma, geleneksel, yenilikçi ve son teknoloji kolektif teknikleri tanıtmakta ve bunları sistematik şekilde incelemektedir. Ayrıca, makine öğrenimi, sınıflandırma algoritmaları, algoritma geliştirme, güncel araştırma eğilimleri ve karşılaşılan zorluklar gibi başlıklar altında alandaki gelişmeleri kapsamlı biçimde analiz ederek araştırmacılara güncel bilgiler sağlamaktadır.

Ayrıca, yukarıda belirtilen yeni yöntemlerin yanı sıra, bu tez kapsamında gerçekleştirilen çalışmada yaygın olarak kullanılan bagging, stacking, blending ve voting gibi temel kolektif tekniklere de kısaca değinilmektedir. Bu yöntemlerin özet bir şekilde karşılaştırıldığı görsel, Şekil 2.1'de Kolektif Öğrenmenin Literatürdeki En Yaygın Yaklaşımları başlığı altında sunulmaktadır.



Şekil 2.1. Kolektif öğrenmenin literatürdeki en yaygın yaklaşımları

Kolektif öğrenme yöntemleri arasında yer alan Bagging (Bootstrap Aggregation) [14], aynı öğrenme algoritmasının farklı alt küme ve setlerde eğitilmiş birden fazla örneğini oluşturur ve bu modellerin tahminlerini birleştirerek genel doğruluğu artırmayı hedefler. Bu yöntem, özellikle varyansı azaltmak ve aşırı öğrenmeyi önlemek için etkilidir. Boosting [15] ise, temel modelleri art arda eğiterek her yeni modelin, önceki modellerin hatalarını düzeltmeye odaklandığı ardışık bir öğrenme yaklaşımıdır; bu sayede modelin genel hatası azaltılır ve doğruluk artırılır. Stacking [16], birden fazla modelin tahminlerini birleştirmek amacıyla bu tahminleri girdi olarak kullanan bir meta-model eğitir. İki katmandan oluşan stacking yapısında, birinci seviye temel modellerden, ikinci seviye ise bu modellerin çıktıları üzerinden öğrenen bir meta-algoritmadan oluşur. Blending [17] yöntemi de stacking ile benzerlik gösterir; ancak genellikle farklı model türleri yerine aynı algoritmanın farklı hiperparametre kombinasyonlarıyla eğitilmiş versiyonlarının çıktıları birleştirilir. Blending, yapısal olarak daha basit bir doğrulama stratejisi kullanır ve özellikle küçük veri setlerinde daha pratik bir çözüm sunabilir.

Voting yöntemi [18], birden fazla modelin tahminlerini birleştirir ve nihai tahmin genellikle çoğunluk oyu ile belirlenir. Farklı model türleri aynı veri üzerinde bağımsız olarak eğitilir ve ardından tahminler birleştirilir, çoğunlukla en sık tahmin edilen sınıf, keskin sınıflandırma yerine bir olasılık dağılımı yaratır ve en yüksek değer seçilir, ağırlıklı oylama

tekniği kullanılır. Eğer bir model, diğer modellere göre daha iyi sonuç veriyorsa, bu modelin daha önemli olmasını sağlamak için bir ağırlık değeri belirlenir ve nihai karar, ağırlıklı ortalama ile yapılır.

Bu keşif yoluyla, okuyuculara kolektif öğrenimi daha derinlemesine bir şekilde anlamalarına yardımcı olmak, bunun gerçek dünyadaki makine öğrenimi zorluklarını çözme konusundaki çok yönlülüğünü ve faydasını takdir etmelerini sağlamak amaçlanmıştır.

2.2. Kolektif Öğrenmede Alt Kümelendirme

Günümüz kolektif öğrenme algoritmalarında bazı yöntemler, veriyi değiştirmeden tüm temel modelleri aynı veri kümesi ile eğitirken, diğer yöntemler veri kümesini rastgele alt kümelere ayırarak her alt kümeyi farklı temel modellerin eğitimi için kullanmaktadır. İkinci yöntemde, tüm modellerde aynı öznitelik kümesi korunurken, kullanılan örnek sayısı değişmektedir. Her iki yaklaşımda da verinin tamamı kullanılabilir. Sonraki bölümlerde, deneysel çalışma sırasında keşfedilen ve temel modelin/modellerin en iyi şekilde bir meta-modele entegre edilmesini amaçlayan yenilikçi bir fikir paylaşılmaktadır.

2.2.1. Literatürde kullanılan standart veri alt kümeleme teknikleri

Bu çalışma, veri manipülasyonu sürecinde kullanılan yeni bir veri bölme tekniği sunmaktadır. Ayrıca, temel modellerin oluşturulması sürecinde farklı öğrenme algoritmalarının kullanımını da içermektedir. Bu yaklaşımın, literatürde mevcut kullanılan senaryolarının Sert Oylama (Hard Voting) ve Bagging (Bootstrap Aggregation) yöntemleriyle ilişkili olduğu gözlemlenebilir. Aynı zamanda, kolektif öğrenme yöntemlerinden güçlendirme (boosting) yöntemine ait XGBoost metodu [19] ile alt kümeleme dışında kalan faktörler incelenebilir.

<u>Sert oyama yöntemi</u>

Sert oylama (hard voting) yöntemi, kolektif (ensemble) öğrenme kapsamında farklı sınıflandırıcıların tahminlerine dayalı olarak karar vermeyi amaçlar. Bu yöntemde birden fazla model, genellikle farklı algoritmalara sahip olacak şekilde eğitilir ve her model test örneği için bir sınıf tahmininde bulunur. Nihai çıktı ise bu tahminlerin çoğunluk oyu

(majority voting) ile belirlenir. Bu teknik, farklı öğrenme algoritmalarının güçlü ve zayıf yönlerini birleştirerek genelleme başarısını artırmayı hedefler.

Bu yöntemin hesaplama maliyeti, her modelin farklı bir öğrenme algoritması kullanmasından ötürü homojen kolektif modellere kıyasla daha değişkendir. Her modelin eğitim süresi birbirinden farklı olabileceği için, toplam eğitim süresi şu şekilde ifade edilebilir:

$$T_{hardvoting} = \sum_{i=1}^{m} t_i \tag{2.1}$$

Burada:

- m: toplam model sayısı (farklı algoritma sayısı),
- t_i: *i.* modelin eğitim süresi (algoritma ve veri yapısına bağlı olarak değişken).

Eğer her model aynı öznitelik seti ile eğitiliyorsa ve model sayısı sabitse, toplam eğitim maliyeti lineer olarak artar. Her model için farklı eğitim algoritmaları kullanılsa da her biri aynı sayıda *n* özelliği kullanıyorsa, ortalama eğitim süresi *a* ile ifade edilirse, toplam eğitim süresi,

$$T_{hardvoting} \approx m \cdot a \tag{2.2}$$

şeklinde ifade edilir.

Sert oylama yönteminde her model farklı algoritmalara sahip olduğundan, hesaplama karmaşıklığı en yavaş modele bağlı olarak değişir. Ancak ortalama bir analizde her modelin eğitim maliyeti *O(f(n))* şeklinde kabul edilirse, toplam karmaşıklık:

$$O_{hardvoting} = \sum_{i=1}^{m} O(f_i(n)) \approx O\big(m \cdot f(n)\big) \tag{2.3}$$

Bu analizde:

- $f_i(n)$: *i.* algoritmanın öznitelik sayısına bağlı asimptotik karmaşıklığıdır.
- Eğer modellerin karmaşıklıkları benzerse, bu ifade $O(m \cdot f(n))$ şeklinde sadeleşir.

Örneğin:

- SVC: $O(n^2)$,
- Lojistik Regresyon: $O(n)$,
- Rastgele Orman: $O(n \cdot \log n)$.

Bu durumda, en yavaş algoritma belirleyici olur.

XGBoost yöntemi

XGBoost (Extreme Gradient Boosting), gradyan artırmalı karar ağaçlarına (GBDT) dayalı, optimize edilmiş ve yüksek performanslı bir kolektif öğrenme yöntemidir. XGBoost, her ağaç yapısını önceki ağaçların hatalarını minimize edecek şekilde art arda inşa eder ve bu süreci ikinci dereceden Taylor açılımı ve düzenlileştirme terimleriyle destekler. Bu yapı hem hata azaltımı hem de model karmaşıklığı kontrolünü bir arada sağlamaktadır.

XGBoost algoritmasında her bir ağaç oluşturulurken veri seti çok sayıda bölme noktası (split point) üzerinde taranır. Ağaç derinliği *d*, öznitelik sayısı *n*, örnek sayısı *m* ve toplam ağaç sayısı *T* olmak üzere, her bir ağacın oluşturma maliyeti şu şekilde yaklaşık ifade edilir:

$$T_{XGBoost} = T \cdot O(n \cdot m \cdot \log m) \tag{2.4}$$

Bu analizde,

- $O(n \cdot m)$: Her split için tüm örnekler ve tüm öznitelikler üzerinden tarama,
- $\log m$: Histogram temelli bölme optimizasyonlarında örneklerin sıralaması ve bölme noktası seçimi için gereken zaman.

Eğer histogram tabanlı XGBoost kullanılıyorsa, bu karmaşıklık sıralama işlemlerinin azaltılması ile $O(n \cdot m)$ seviyelerine düşürülebilir. Bu tür optimizasyonlarla paralelleştirme ve bellek kullanımı azaltılarak daha hızlı model eğitimi elde edilir. XGBoost'un toplam hesaplama karmaşıklığı:

$$O(T \cdot n \cdot m \cdot \log m) \tag{2.5}$$

şeklindedir. Bu, ağaç sayısı *T*, öznitelik sayısı *n* ve veri büyüklüğüne *m* bağlı olarak logaritmik olarak artar.

Bootstrap örnekleme yöntemi

Bootstrap yöntemi, istatistik ve makine öğrenmesinde, sınırlı bir örneklem üzerinden daha fazla bilgi elde etmek amacıyla geliştirilen bir yeniden örnekleme (resampling) tekniğidir. Bu yöntem, orijinal veri kümesinden geri koymalı (with replacement) şekilde çok sayıda alt örneklem (bootstrap örneklemi) üretir. Bu örneklemler aynı boyutta olup, modelin varyansını azaltmak ve genelleme kabiliyetini artırmak amacıyla her biri üzerinde bir model eğitilmesini mümkün kılar. Özellikle Bagging (Bootstrap Aggregating) gibi kolektif öğrenme yöntemlerinin temelini oluşturur.

Her bir model tüm özellik setini kullanarak eğitildiği için, toplam eğitim maliyeti tüm modeller için özellik sayısının toplamına bağlıdır. Eğer *n* toplam özellik sayısını, *m* ise eğitim yapılan model sayısını (ensemble boyutu) temsil ederse, her bir model yaklaşık olarak *n* adet özelliği işler.

Dolayısıyla tek bir modelin eğitim süresi yaklaşık $n \cdot T$ kadardır (burada *T* bir modelin tek bir özelliği işlemesi için gereken sabit zamanı ifade eder). Tüm *m* modelin eğitimi için zaman maliyeti bu değerin *m* ile çarpımıdır. Bootstrap yöntemi için toplam eğitim süresi:

$$T_{Bootstrap} = T \cdot n \cdot m \tag{2.6}$$

Denklem 2.6, her modelin tüm *n* özelliği işlediğini ve bu işlemin *m* kez tekrarlandığını gösterir. Bagging yönteminde her model tam özellik kümesiyle eğitildiğinden, özellik işleme sayısı *n*'nin *m* ile çarpımına eşittir. Bu, bagging yönteminin hesaplama yükünün model

sayısıyla doğru orantılı olarak arttığını gösterir (yani modeller paralel de eğitilse toplam iş yükü m katına çıkar). Nitekim genel olarak kolektif modelin eğitimi, tek bir modelle karşılaştırıldığında daha fazla hesaplama gerektirir; bu, daha yüksek doğruluk elde etmek için ekstra işlem yükü üstlenmek anlamına gelir.

Dolayısıyla asimptotik hesaplama karmaşıklığı:

$$O(T \cdot m \cdot n) \tag{2.7}$$

şeklindedir. Denklem 2.7'de, her bir modelin eğitim süresi (T), veri büyüklüğü (n) ve model sayısı (m) arttıkça doğrudan doğruya işlem sayısının artması anlamına gelir. Ancak bu süreç paralelleştirilebildiğinden (örneğin her alt küme aynı anda üretilebilir), pratikte oldukça verimli bir yöntemdir.

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6
	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
5	7.4	0.66	0.00	1.8	0.075	13.0	40.0	0.9978	3.51	0.56	9.4	5
6	7.9	0.60	0.06	1.6	0.069	15.0	59.0	0.9964	3.30	0.46	9.4	5
	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
6	7.9	0.60	0.06	1.6	0.069	15.0	59.0	0.9964	3.30	0.46	9.4	5
7	7.3	0.65	0.00	1.2	0.065	15.0	21.0	0.9946	3.39	0.47	10.0	7
8	7.8	0.58	0.02	2.0	0.073	9.0	18.0	0.9968	3.36	0.57	9.5	7
9	7.5	0.50	0.36	6.1	0.071	17.0	102.0	0.9978	3.35	0.80	10.5	5

Şekil 2.2. Örnek tabanlı veri bölme (yatay alt kümeleme)

Bootstrap yöntemi, mevcut kolektif öğrenme süreçlerinde kullanılan yatay (örnek tabanlı) alt kümelemeyi temsil etmektedir [14]. Kavramı açıklamak için Şekil 2.2'de gösterilen örnek veri kümesini ele alalım. Bu veri kümesi, 11 öznitelik, 12. "quality" hedef değişken ve 10 örnek içermektedir. Veri, yatay olarak 3 alt kümeye bölünerek her birinde 11 öznitelik ve 4

örnek bulunan 3 homojen modelin eğitimi sağlanmaktadır. Ancak, rastgele örneklemeye dayalı alt kümeleme yöntemi, bazı temel problemlere neden olabilir.

Bagging yönteminin temeli ve problemleri

Bagging yönteminde, orijinal veri kümesinden geri koymalı (with replacement) örnekleme ile çok sayıda alt küme (bootstrap örneklemesi) oluşturulur. Her bir alt küme, orijinal veri kümesi ile aynı boyutta olur ve her alt küme üzerinde bir model eğitilir.

Ancak, rastgele bölme yöntemi kullanıldığında her alt kümenin aynı boyutta tutulması bazı problemlere yol açabilir:

1. Anomali içeren örnekler birden fazla kez birden fazla kümede bulunarak modellerin eğitimine zarar verebilir.
2. Önemli olan bazı örnekler bazı kümede bulunmayabilir.
3. Ayrıca, rastlantısal veya belirlenen hiperparametre ayarlarına bağlı olarak alt kümeler arasında sistematik örtüşmeler meydana gelebilir. Bu durum, modeller arası varyansı azaltarak kolektif öğrenme avantajlarından yararlanılmasını engeller.
4. Aynı zamanda, alt kümelerin eşit büyüklükte olması her zaman işlem süresi açısından bir avantaj sağlamayabilir.

Bu nedenle, sıralanan problemlerin üstesinden gelen alternatif veri bölme yöntemi geliştirilerek daha etkin bir kolektif öğrenme yapısı oluşturulması hedeflenmiştir. Önerilen çözüm Bölüm 3.4'te detaylandırılmıştır.

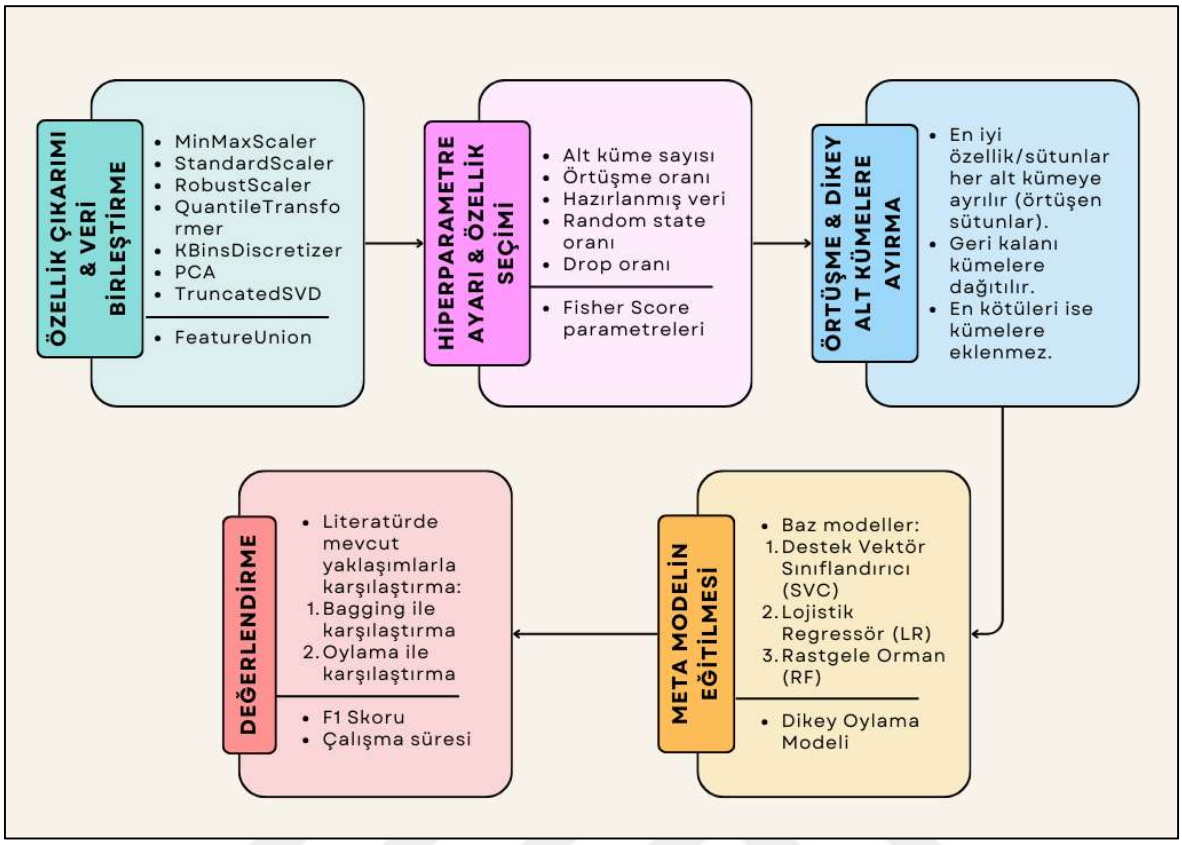
3. METODOLOJİ

Bu çalışmada önerilen metodoloji, kolektif öğrenme sürecinde alt kümelere ayırmanın gücünden yararlanarak meta-model performansını artırmayı hedefleyen dikey bir iş akışına dayanır. Dikey alt kümelendirme yaklaşımının adaptif yapısı, farklı öğrenme algoritmaların farklı öznitelik kümelerinde dinamik bir şekilde modeller geliştirmek için kullanılmış ve hiperparametre optimizasyonuyla desteklenmiştir. Genel süreç, ham verilerin dikey (öznitelik tabanlı) ayırma, model eğitimi ve performans değerlendirmesi gibi bir dizi aşamayı içerir. Bu bölümde, metodolojinin tüm aşamaları ayrıntılı olarak açıklanmış, şekillerle desteklenmiş ve önerilen yöntemin faydaları kapsamlı bir şekilde ele alınmıştır.

Kolektif öğrenme, veri kümesinde yapılan değişiklikler veya çeşitli öğrenme algoritmalarının kullanılmasıyla uygulanabilir. Bölüm 2.2'de açıklandığı üzere, mevcut yaklaşımlar yalnızca yatay alt kümeleme yöntemini kullanmakta ve verideki değişiklikler satır bazında yapılmaktadır. Farklı kolektif öğrenme yöntemlerinde gerçekleştirilen yatay alt kümeleme (örnekler) nedeniyle, alt kümeler arasındaki varyans yeni farklılıklar yaratmaktadır. Bu deneysel çalışma, alt kümeler arasındaki farkın dikey alt kümeleme (öznitelikler) yöntemi uygulanarak oluşturulması durumunda ne kadar fayda sağlayabileceğini test etmeyi amaçlamıştır.

Tez çalışması kapsamında, önerilen yöntem, "Dikey Alt Kümeleme" ile "Sert Oylama (Hard Voting)" yöntemlerinin bir kombinasyonu kullanılarak uygulanmıştır. Önerilen metodolojiyi gerçekleştirmek için, öznitelik mühendisliği, veri hazırlama, model eğitimi ve yeni dikey oylama modelinin uygulanmasını içeren bir geliştirme prosedürü uygulanmıştır. Bu prosedür, genel olarak tüm testleri ve optimizasyonları temsil etmekte olup, farklı veri hazırlama türlerini ve karşılaştırma amacıyla farklı model eğitim türlerini içermektedir.

Metodolojinin genel akış şeması, Şekil 3.1'de gösterilmektedir. Süreç, öznitelik mühendisliği ile başlayıp, dikey alt kümelendirme, kolektif modelinin eğitimi ve performans değerlendirmesi ile devam eder. Bu sistematik akış diyagramı, önerilen yöntemin tüm adımlarını özetlemektedir.



Şekil 3.1. Geliştirme sürecinin akış şeması

3.1. Kolektif Öğrenme Fayda Analizi

Kolektif öğrenme yöntemleri, çeşitli nedenlerden dolayı tahmin performansını sıklıkla artırır [10,20]. Performans artırma nedenleri aşağıda özetlenmiştir.

Aşırı öğrenmenin önlenmesi

Verinin sınırlı olduğu durumlarda, öğrenme algoritmaları eğitim verilerini mükemmel bir şekilde tahmin eden ancak görülmemiş örneklerde düşük performans gösteren çok sayıda hipotez üretebilir. Birden fazla hipotezin ortalamasını almak, yanlış bir hipotez seçme riskini azaltır ve böylece genel tahmin doğruluğunu artırır.

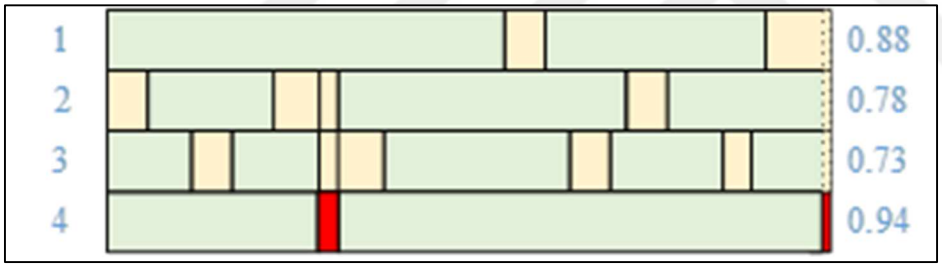
Hesaplama avantajı

Tekil modeller, yerel arama gerçekleştirirken yerel optimumlara takılıp kalabilir. Kolektif öğrenme yöntemleri, birden fazla modeli birleştirerek bu riski azaltır ve yerel minimuma

ulaşma olasılığını düşürür.

Temsiliyet

Optimum hipotez, tek bir modelin kapsama alanının ötesinde olabilir. Çeşitli modellerin bir araya getirilmesiyle, kolektif öğrenme yöntemleri arama alanını genişletir ve veri alanı ile daha iyi bir uyum sağlar. Şekil 3.2'de, yeşil renk doğru tahmin edilen veriyi, sarı renk ise yanlış tahmin edilen veriyi göstermektedir. İlk üç modele bakıldığında, yanlış tahmin edilen verilerin bazı yerlerde çakıştığı görülmektedir. Kırmızı renk, çakışan yanlış tahminleri ifade eder. Çakışan bölgeler, iki temel modelin aynı örneklerde veriyi yanlış tahmin etmesi nedeniyle meta-modelin veriyi yanlış tahmin ettiği örnekleri temsil eder. Kolektif öğrenme, bu prensipte üstünlük sağlar. Eğer temel modeller, birbirlerinden farklı bir şekilde eğitilirlerse, meta-model, temel modellerin farklı örneklerde yaptığı hatalardan faydalanabilir.



Şekil 3.2. Kolektif öğrenmenin fayda analizi

Bununla birlikte, Şekil 3.2, tüm modellerin birleştirilmesiyle eğitim örnekleri için mükemmel bir uyum elde edildiğini göstermektedir.

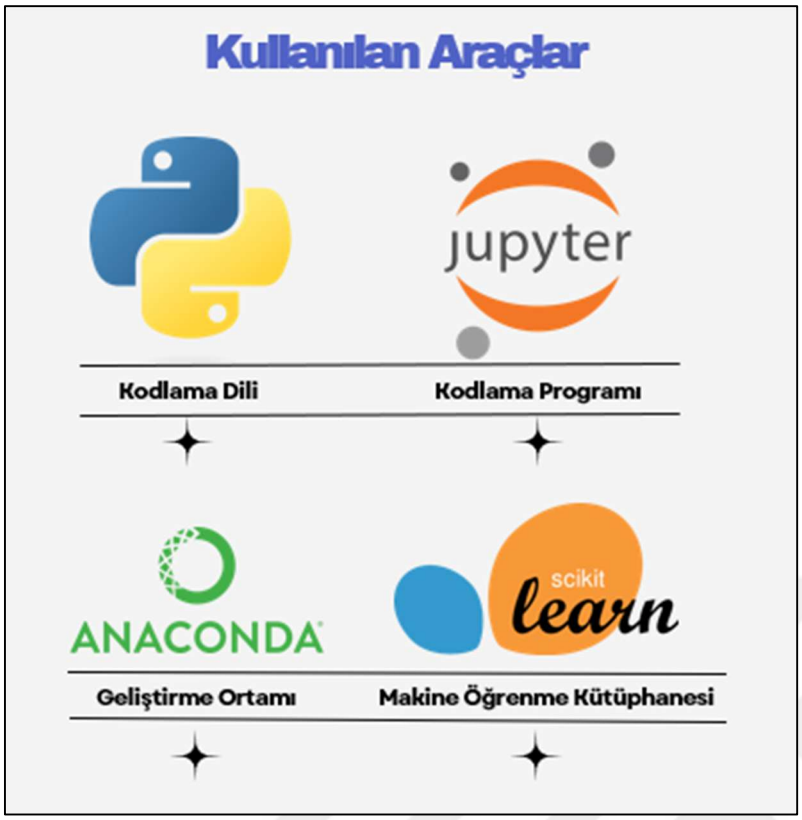
3.2. Donanım, Geliştirme Ortamı ve Araçların Belirlenmesi

Bu tezde, kodlama platformları, veri hazırlama, öznitelik çıkarımı, öznitelik seçimi ve model eğitimi için çeşitli açık kaynaklı araçlar ve kütüphaneler kullanılmıştır. Çizelge 3.1'de, bu çalışmada kullanılan araç ve kütüphaneleri özetlemektedir.

Çizelge 3.1. Kullanılan donanım, araç, kütüphane ve algoritmalar

No	Araç/Kütüphane	Açıklama
1	Python	Programlama dili
2	Jupyter Notebook	Etkileşimli kodlama ortamı
3	Anaconda	Geliştirme ve bağımlılık yönetimi
4	Scikit-learn	Makine öğrenimi algoritmaları ve araçları
5	MinMaxScaler, StandardScaler, RobustScaler, QuantileTransformer, KbinsDiscretizer, PCA, TruncatedSVD	Kullanılan Öznitelik Çıkarma Yöntemleri
6	Pandas, NumPy	Veri işleme ve manipülasyon
7	Feature Union	Öznitelik Birleştirme Algoritması
8	Fisher Score	Öznitelik Seçimi Algoritması
9	Bootstrap Data Splitting	Veri Bölümleme Tekniği
10	Lojistik Regresyon Sınıflandırıcı, Destek Vektör Sınıflandırıcı, Rastgele Orman Sınıflandırıcı	Baz Makine Öğrenimi Algoritmaları
11	Voting CLassifier	Kolektif Öğrenme Algoritması
12	GridSearch	Parametre Optimizasyon Metodu
13	Wine Quality-Red, Heart Desease	Açık Kaynaklı Veri Setleri
14	Intel i7-10875H CPU (2.3 Ghz)	Eğitimin Gerçekleştirildiği İşlemci
15	16x2 GB DDR4 RAM	Eğitimin Gerçekleştirdiği RAM

Python [21] programlama dili, çok yönlülüğü, geniş kütüphane yelpazesi ve güçlü topluluk desteği nedeniyle tercih edilmiştir. Etkileşimli veri analizi ve model prototipleme için Jupyter Notebook [22] kullanılmıştır. Proje bağımlılıklarını yönetmek ve uygun bir geliştirme ortamı oluşturmak için Anaconda [23] tercih edilmiştir. Veri manipülasyonu Pandas [24] ve NumPy [25] kütüphaneleri ile gerçekleştirilmiş, makine öğrenimi görevleri için, sınıflandırma, regresyon ve kümeleme gibi çeşitli algoritmalar ve araçlar sunan, güçlü ve kullanıcı dostu bir kütüphane olan Scikit-learn [26] kütüphanesi tercih edilmiştir. Şekil 3.3, çalışmada kullanılan araçları ifade etmektedir.



Şekil 3.3. Çalışmada kullanılan araçlar

Öznitelik çıkarma sürecinde, MinMaxScaler [27], StandardScaler [28], RobustScaler [29], QuantileTransformer [30], KBinsDiscretizer [31], PCA (Principal Component Analysis) [32] ve TruncatedSVD (Singular Value Decomposition) [33] teknikleri uygulanmıştır. Birden fazla öznitelik çıkarma yöntemini veya ardışık işlem hattını tek bir dönüştürücüde birleştirmek ve farklı öznitelik gruplarını eşzamanlı olarak işlemek amacıyla Feature Union [34] yöntemi kullanılmıştır. Öznitelik seçimi için, farklı sınıfları ayırmada bireysel özniteliklerin ayrım gücünü değerlendiren istatistiksel bir ölçüt olan Fisher Skoru [35], tüm öğrenme algoritmalarında kullanılacak en değerli öznitelikleri belirlemek için kullanılmıştır.

Önerilen yeni dikey veri bölme tekniğinin geçerliliğini karşılaştırmak ve test etmek için mevcut bir yatay veri bölme tekniği olan Bootstrap Aggregation kullanılmıştır [14]. Sonuç olarak, farklı eğitim ve doğrulama veri alt kümeleri oluşturmak amacıyla iki farklı veri alt kümeleme tekniği kullanılmış ve bu alt kümeler temel modellerin girdisi olarak değerlendirilmiştir.

Temel model oluşturma aşamasında, çeşitli denetimli öğrenme algoritmaları üzerinde çalışılmıştır. Bu algoritmalar arasında, ikili sınıflandırma görevleri için uygun bir lineer

model olan Lojistik Regresyon [36], doğrusal olmayan sınıflandırma için Destek Vektör Sınıflandırıcısı (SVC) [37] ve topluluk tabanlı sınıflandırma yetenekleriyle bilinen Random Forest Classifier [38] bulunmaktadır.

Meta-model oluşturma aşamasında, birden fazla sınıflandırıcının toplu bilgeliğinden faydalanmak ve çıktıları birleştirmek için Voting Classifier algoritması temel modeller arasında kullanılmıştır.

Eğitilen modeller ve yapılan tüm alt kümelere ayırma işlemleri için parametrelerin optimizasyonu GridSearch [39] metotu kullanılarak gerçekleştirilmiştir.

Geliştirme süreci boyunca, makine öğrenimi sınıflandırma görevlerinde sıkça kullanılan açık kaynaklı bir veri kümesi olan Wine Quality-Red veri kümesi [40] kullanılmıştır. Bu veri kümesi, kırmızı şarapların çeşitli kimyasal özelliklerini içererek bu özelliklere dayalı olarak şarap kalitesinin tahmin edilmesi için uygun bir ortam sunmaktadır. Ayrıca, önerilen yöntemin genel güvenilirliğini değerlendirmek amacıyla Heart Disease veri kümesine [41] de uygulanmıştır. Bu veri kümesi, her biri hastanın sağlığı hakkında bilgi veren çok sayıda özelliğe sahip hasta kayıtlarından oluşmaktadır.

Modelin eğitimi, 10. nesil bir Intel i7-10875H işlemci [42] üzerinde gerçekleştirilmiştir. Bu işlemci, 8 fiziksel çekirdek ve 16 iş parçacığı (thread) ile yüksek düzeyde çoklu görev işleme kapasitesine sahiptir. Temel saat hızı 2.3 GHz olmakla birlikte, Turbo Boost teknolojisi sayesinde daha yüksek frekanslara çıkabilmektedir. Model eğitimi sırasında herhangi bir derin öğrenme veya yüksek hacimli görüntü işleme yapılmadığı için GPU hızlandırmasına ihtiyaç duyulmamıştır. Bu nedenle CUDA desteği gerektiren NVIDIA grafik kartı kullanılmamıştır. Eğitim süreci, CPU üzerinden verimli bir şekilde tamamlanmıştır. Bu tercih, işlemcinin çok çekirdekli yapısının sağladığı yeterli performans ile birlikte kaynak kullanımının sade tutulmasına olanak sağlamıştır.

Model eğitimi sırasında kullanılan sistemde 16 GB × 2 olmak üzere toplam 32 GB DDR4 SODIMM bellek [43] bulunmaktadır. İkili kanal (dual-channel) yapıdaki bu bellek konfigürasyonu, bellek bant genişliğini artırarak veri aktarım hızını optimize etmektedir. Bu sayede büyük veri kümelerinin belleğe doğrudan yüklenmesi ve eğitim sırasında ortaya çıkan ara verilerin hızlı işlenmesi mümkün olmuştur. Yeterli bellek kapasitesi, disk tabanlı

yavaş erişim ihtiyacını en aza indirerek CPU üzerinde gerçekleştirilen model eğitiminin kesintisiz ve verimli ilerlemesini sağlamıştır.

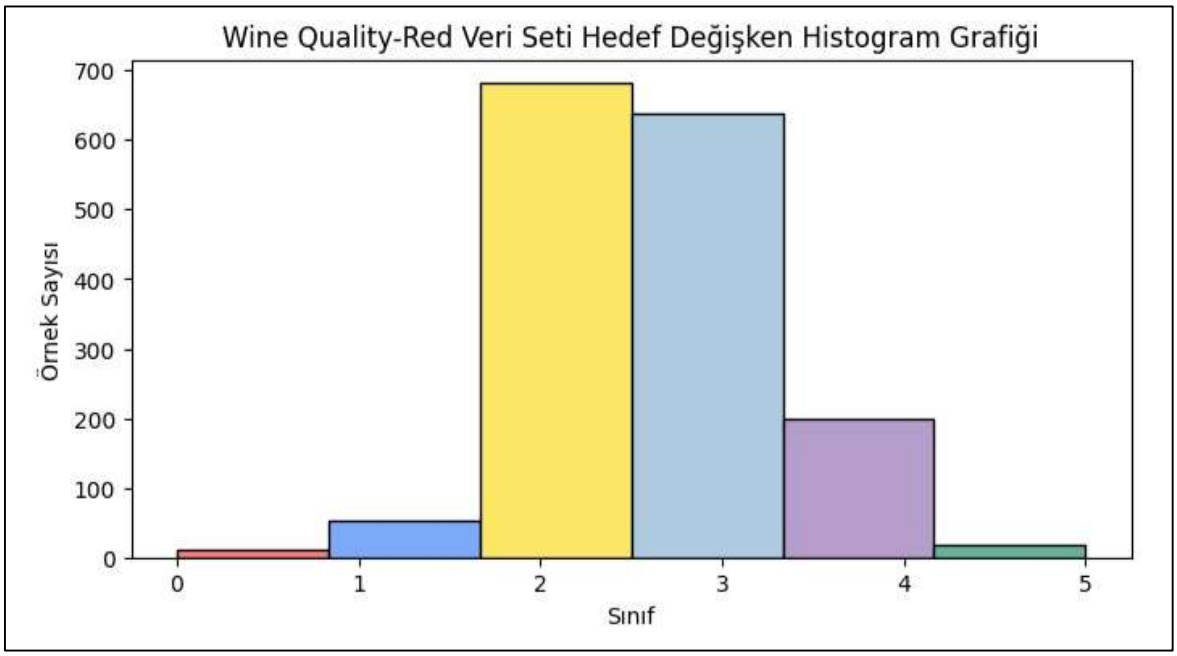
Her bir araç, görevle olan uyumluluğu, makine öğrenimi topluluğundaki popülaritesi ve iş akışına kolay entegrasyonu temel alınarak seçilmiştir. Bu araçlar topluca, problem alanı için makine öğrenimi modellerinin geliştirilmesini, değerlendirilmesini ve yorumlanmasını kolaylaştırmıştır.

3.3. Veri Setinin Belirlenmesi ve Hazırlanması

Bu çalışmada, klasik modellerin bu veri kümelerinde düşük eğitim ve tahmin doğrulukları göstermesi nedeniyle "Wine Quality-Red" [40] ve "Heart Disease" [41] veri kümeleri seçilmiştir.

Wine Quality-Red veri kümesine ait detaylar Çizelge 3.2 ve Çizelge 3.3'te gösterilmektedir. Bu veri kümesi, fizikokimyasal testlere dayalı 11 giriş değişkeni ve 1 hedef değişken ile toplamda 1599 örnek içermektedir.

Çizelge 3.2. Wine Quality-Red veri kümesinin hedef değişkeninin sınıf dağılımı



Çizelge 3.2'de Wine Quality-Red veri kümesinin hedef değişkeninin sınıf dağılımı temsil

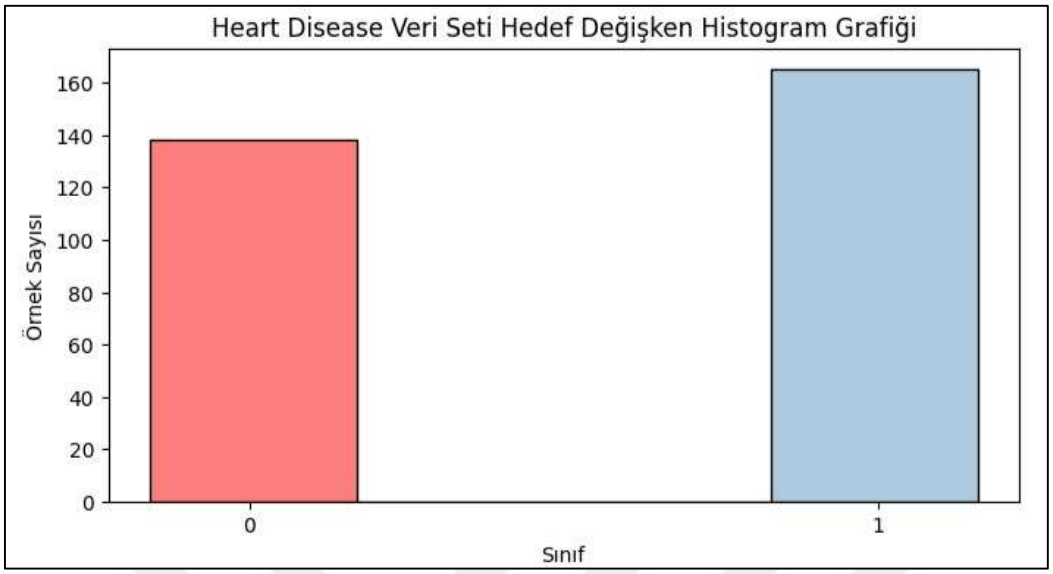
etmektedir. Hedef değişkeni, 0-5 kadar kalite puanlarını belirtmekte olup toplamda 6 sınıf içermektedir. Aynı zamanda, veri kümesinin hedef değişkeni dengesizdir (0: 10, 1: 53, 2: 681, 3: 638, 4: 199, 5: 18).

Çizelge 3.3. Şarap kalitesi-kırmızı veri setinin içeriği

No	Öznitelik Adı	Açıklama
1	Sabit Asitlik	Şarapta baskın olan tartarik, süksinik, sitrik ve malik asitler gibi sabit asitler.
2	Uçucu Asitlik	Şarapta bulunan yüksek asetik asit, hoş olmayan sirke tadına neden olur.
3	Sitrik Asit	Şarapta tazelik ve lezzeti artırmak için kullanılan zayıf bir organik asit.
4	Kalıntı Şeker	Fermentasyon sonrası kalan şeker miktarı.
5	Klorürler	Şaraptaki tuz miktarı. Düşük klorür oranı, daha kaliteli şaraplar üretir.
6	Serbest Kükürtdioksit	Şarabı oksidasyon ve mikrobiyal bozulmadan korumak için kullanılan (SO)2.
7	Toplam Kükürtdioksit	Serbest ve bağlı (SO)2 formlarının toplam miktarı.
8	Yoğunluk	Alkol ve şeker içeriğine bağlıdır. Daha iyi şaraplar genellikle daha düşük yoğunluğa sahiptir.
9	pH	Şarabın asitlik veya alkalilik seviyesini kontrol etmek için kullanılır.
10	Sülfatlar	Şaraba eklenen antibakteriyel ve antioksidan bir ajan.
11	Alkol	Şaraptaki alkol yüzdesi. Daha yüksek alkol konsantrasyonu, genellikle daha iyi kaliteye işaret eder.
12	Kalite (Çıkış Değişkeni)	Şarabın kalite derecesi. (Hedef Değişken)

Heart Disease veri kümesine ait detaylar Çizelge 3.4 ve Çizelge 3.5'te gösterilmektedir. Bu veri kümesi, her biri hastaların sağlık durumlarına ilişkin çeşitli bilgileri içeren toplamda 303 örnek ve 13 giriş değişkeni ile 1 hedef değişken barındırmaktadır. Bu değişkenler, hastalık riskinin belirlenmesinde önemli olan demografik, klinik ve laboratuvar ölçümlerini kapsamaktadır.

Çizelge 3.4. Heart Disease veri kümesinin hedef değişkeninin sınıfı dağılımı



Çizelge 3.4, Heart Disease verisi sınıf dağılımının 0: 138, 1: 165 olduğunu göstermektedir.

Çizelge 3.5. Kalp Hastalığı Veri Setinin İçeriği

No	Öznitelik Adı	Açıklama
1	Yaş	Hastanın yaşı (yıl olarak).
2	Cinsiyet	Hastanın cinsiyeti (1: Erkek, 0: Kadın).
3	Göğüs Ağrısı Türü	Angina tipi (0: Yok, 1-3: Belirli türde göğüs ağrısı).
4	Kan Basıncı	Dinlenme sırasında ölçülen kan basıncı (mmHg).
5	Kolesterol	Serum kolesterol seviyesi (mg/dl).
6	Kan Şekeri	Açlık kan şekeri seviyesi (1: ≥120 mg/dl, 0: <120 mg/dl).
7	Elektrokardiyografi	Dinlenme sırasında EKG sonuçları (0-2: Belirli durumlar).
8	Maksimum Nabız	Egzersiz sırasında elde edilen maksimum kalp atış hızı.
9	Egzersiz Anjinası	Egzersiz sırasında anjina (1: Var, 0: Yok).
10	ST Depresyonu	Egzersiz sonrası dinlenme ile ilgili ST depresyonu değeri.
11	ST Eğimi	Egzersiz sırasında ST segmentinin eğimi (0-2: Farklı durumlar).
12	Majör Damar Sayısı	Floroskopi ile gözlemlenen majör damar sayısı (0-3).
13	Talasemi Türü	Talasemi türü (0-3: Farklı sınıflamalar).
14	Hedef (Çıkış)	Kalp hastalığı durumu (1: Hastalık var, 0: Hastalık yok).

3.4. Önerilen Öznitelik Tabanlı Veri Alt Kümeleme Tekniği

Bu çalışmada, Bölüm 2.2.1'de açıklanan problemleri azaltmak amacıyla temel modellerin eğitimi için verinin gruplara ayrılmasına yönelik yeni bir strateji olan dikey oylama tabanlı bir alt kümeleme yöntemi önerilmektedir. Bu yöntemde, veri yatay olarak örnekler bazında değil, öznitelik bazında dikey şekilde bölünmektedir.

- Öncelikle tüm öznitelikler Fisher Skoru kullanılarak önem sırasına göre sıralanır.
- En önemli öznitelikler, tüm alt küme modellerde ortak olacak şekilde seçilir.
- En önemsiz öznitelikler veri kümesinden çıkartılır.
- Geriye kalan öznitelikler, modeller arasında rastgele dağıtılır.

Bu sayede:

1. Modellerin farklı öznitelik kümesi üzerinde eğitilmesi sağlanarak çeşitlilik artırılır.
2. Aşırı örtüşen örnekleme probleminin önüne geçilir.
3. Her model, veri kümesinin farklı boyutlarına duyarlı hale gelir.

Bu iyileştirmeleri incelemek için Şekil 2.2 ve Şekil 3.4 baz alınırsa, orijinal veri kümesi 11 öznitelik, 1 hedef değişkeni ve 10 örnek içermektedir. Üç modelin eğitimi için, veri 5 öznitelik ve 10 örnek içerecek şekilde 3 gruba bölünmüştür. Bu yöntem, her bir alt küme 5 öznitelik, toplamda da 15 öznitelik içeren bir veri yapısı oluşturmaktadır. Bazı öznitelikler, örneğin "alcohol", önemli bir öznitelik olduğu için her üç alt kümeye dahil edilip örtüşen öznitelik olarak kullanılmıştır. Aynı sebeple, "residual sugar" ve "total sulfur dioxide" öznitelikleri toplamda iki alt kümeye dahil edilerek örtüşen öznitelik olarak kullanılmıştır.

	fixed acidity	volatile acidity	citric acid	residual sugar	alcohol
0	7.4	0.70	0.00	1.9	9.4
1	7.8	0.88	0.00	2.6	9.8
2	7.8	0.76	0.04	2.3	9.8
3	11.2	0.28	0.56	1.9	9.8

	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	alcohol
0	1.9	0.076	11.0	34.0	9.4
1	2.6	0.098	25.0	67.0	9.8
2	2.3	0.092	15.0	54.0	9.8
3	1.9	0.075	17.0	60.0	9.8

	total sulfur dioxide	density	pH	sulphates	alcohol
0	34.0	0.9978	3.51	0.56	9.4
1	67.0	0.9968	3.20	0.68	9.8
2	54.0	0.9970	3.26	0.65	9.8
3	60.0	0.9980	3.16	0.58	9.8

Şekil 3.4. Öznitelik tabanlı veri bölme (dikey alt kümeleme)

Şekil 3.4, dikey alt kümeleme yöntemine bir örnek sunmaktadır. Önerilen veri bölme yöntemi için gerekli olan öznitelik seçme yöntemi olarak, Fisher Skoru kullanılarak özniteliklerin değerlendirilmesiyle gerçekleştirilmiştir. En iyi özniteliklerin modeller arasında örtüşmesini sağlamak için (her alt kümede yer alması için), öznitelik kümesi Fisher Skor tekniği ile önem sıralamasına göre dizilir ve en anlamlıları tüm alt kümelere dahil edilir. En önemsiz öznitelikler veri kümesinden çıkartılır. Kalan öznitelikler, alt kümeler arasında rastgele dağıtılır. Bu yapılandırma, modellerin etiketlerle farklı seviyelerde ilişki kurmasına olanak tanır ve temel modellerde kullanılan verinin aynı anomalilerine maruz kalmak yerine farklı bölümlerinde eğitilmesini sağlar. Ayrıca, veri kümesini daha küçük boyutlu dikey alt kümelere bölerek ve temel modelleri bu daha küçük kümeler üzerinde paralel olarak eğiterek, standart kolektif öğrenme yapılandırmasına kıyasla eğitim iş yükü azaltılır.

Önerilen veri bölme yöntemin matematiksel ifadesi ve hesaplama maliyeti

Bu çalışmada, önerilen öznitelik tabanlı dikey alt kümeleme yöntemi, klasik bootstrap (bagging) yaklaşımı ile karşılaştırmalı olarak değerlendirilmiştir. Kullanılan sistem: Intel i7-10875H CPU (2.3 GHz) ve 32 GB DDR4 RAM (16x2) yapısındadır. Veri setleri olarak UCI Wine Quality [40] ve UCI Heart Disease [41] kullanılmıştır.

Bu yapıda, dikey alt kümeleme yönteminin toplam eğitim maliyeti denklemi:

$$T_{vertical} = \left[c + \frac{(n - c - d)}{m}\right] \cdot T \cdot m \tag{3.1}$$

şeklinde ifade edilir. Burada,

- *n*: toplam öznitelik sayısı (birleştirilmiş ve genişletilmiş veriyle birlikte),
- *c*: her modelde ortak bulunan öznitelik sayısı,
- *d*: çıkarılan, eğitimde kullanılmayan öznitelik sayısı,
- *m*: model sayısı,
- *T*: tek bir özellik için eğitim süreci.

Bu denklem, *c* adet ortak özelliğin her bir modelde tekrarlandığını ve geri kalan *n-c-d* özelliğin tekil olarak paylaştırıldığını yansıtmaktadır. Örneğin, *c = 0* (ortak özellik yok) ve *d = 0* ise $T_{vertical} = n \cdot T$ olur; bu, her özelliğin tam olarak bir modelde kullanıldığı ve toplam işlem yükünün sadece *n* özellik kadar olduğu en uç durumdur. Diğer yandan *c* değeri büyüdükçe formülde $c \cdot m$ terimi öne çıkacak, *c* tüm özelliklere eşit olduğunda *(c = n, d=0)* denklem $T_{vertical} = n \cdot m \cdot T$ ile bagging maliyetine geri döner (yani her özellik her modelde kullanılır hale gelir).

Dikey alt kümeleme tabanlı oylamanın asimptotik karmaşıklık analizleri (Big-O)

Önerilen dikey alt kümeleme yönteminin toplam eğitim karmaşıklığı, her modelin c adet ortak ve $\frac{(n-c-d)}{m}$ adet benzersiz özniteliği işlemesine bağlı olarak yaklaşık şu şekilde ifade edilebilir:

$$O\left(m \cdot \left(c + \frac{(n - c - d)}{m}\right)\right) = O(n + c \cdot m) \tag{3.2}$$

Bu denklem, $(n - d)$ adet düşürülmeyen toplam özniteliği ve ortak özniteliklerin *m* modelde tekrar işlenmesini içeren iki temel bileşeni yansıtır. Tüm eğitimin karmaşıklık analizi Denklem 3.2'nin *T* (tek bir özellik için eğitim süreci) ile çarpılmış halidir. Eğer *c*

(ortak öznitelik sayısı) *n*'ye kıyasla çok küçük bir sabit ise, bu terim ihmal edilebilir ve karmaşıklık yaklaşık olarak $O(n+m)$ şeklinde sadeleşir. Bu durumda, model sayısı ve öznitelik sayısı ayrı ayrı doğrusal etkiler üretir. Öte yandan, en kötü durumda $c = O(n)$ olduğunda, ifade $O(n + n \cdot m) = O(n \cdot m)$ seviyesine çıkarak Bagging yöntemi ile aynı asimptotik karmaşıklığa ulaşır (Denklem 3.4). Bu nedenle, dikey alt kümeleme yönteminin teorik karmaşıklığı $O(n + c \cdot m)$ olup, pratikte ortak öznitelik sayısının küçük tutulması durumunda Bagging'e kıyasla daha iyi ölçeklenme sağlayabilir (Denklem 3.5).

Bu model, özellikle veri yapılarında eğitim maliyetini azaltmak ve model çeşitliliğini artırmak için tasarlanmıştır. Ayrıca bu yöntem, klasik Bagging'de görülen problemlerin önüne geçmektedir (Bkz. Bölüm 2.2.1). Bu sorunlar, öznitelik bazlı ayrım sayesinde azaltılmış ve her modelin veriye farklı bir gözle bakması sağlanmıştır.

Çizelge 3.6. Bootstrap yöntemi ve dikey bölümleme yöntemi veri bölümleme karşılaştırması

Kategori / Yöntem	Bootstrap Yöntemi	Dikey Bölümleme Yöntemi
Alt Kümeleme Tipi	Örnek bazlı, rastgele	Öznitelik bazlı, anlamlı dağıtım
Çeşitlilik	Düşük (sık örtüşme)	Yüksek (öznitelik farklılığı)
Hesaplama Maliyeti	$O(T \cdot m \cdot n)$ (Denklem 2.7)	$O\left(T \cdot m \cdot \left(c + \frac{(n-c-d)}{m}\right)\right)$ (Denklem 3.3)
Paralelleştirme Uygunluğu	Uygun	Çok Uygun
Aşırı Örnekleme Riski	Var	Yok
Performans (F1-Skoru, Kesinlik, Duyarlılık)	Orta	Daha Yüksek (Bölüm 4)

Bölüm 2.2.1'de açıklanan örnek bazlı bölümleme ile önerilen yöntemde kullanılan öznitelik bazlı bölümlemenin hesaplama maliyetleri incelendiğinde $T \cdot m$ ifadelerinin aynı olduğu gözlemlenebilir. Geriye kalan yatay bölümleme için *n* ile dikey bölümleme için $\left(c + \frac{(n-c-d)}{m}\right)$ ifadeleri arasında karşılaştırma yapıldığında,

$$\left(0 \leq c + \frac{(n-c-d)}{m} \leq n\right) \tag{3.3}$$

$$\left(0 \leq n + \frac{(n-n-0)}{m} \leq n\right) \equiv 0 \leq n \leq n \tag{3.4}$$

$$\left(0 \leq 0 + \frac{(n-0-d)}{m} \leq n\right) \equiv 0 \leq \frac{(n-d)}{m} \leq n \tag{3.5}$$

ilişkisi gözlemlenmektedir.

Sonuç olarak, önerilen dikey oylama yöntemi; eğitim verisinin daha dengeli ve bilgi açısından anlamlı şekilde bölünmesini sağlarken, hesaplama yükünü azaltmakta ve model çeşitliliğini artırmak yoluyla genelleme başarısını yükseltmektedir.

3.4.1. Öznitelik mühendisliği adımlarının uygulanması

Öznitelik tabanlı alt kümeler oluşturmak için, her alt kümenin yeterli sayıda anlamlı özelliğe sahip olması gerektiğinden, mevcut verilerden daha fazla öznitelik çıkarmak gerekmektedir. Bu amaçla MinMaxScaler, StandardScaler, RobustScaler, QuantileTransformer, KBinsDiscretizer, PCA (Principal Component Analysis) ve TruncatedSVD (Singular Value Decomposition) yöntemleri uygulanmış ve Şarap Kalitesi-Kırmızı veri kümesinden 69, Kalp Hastalığı veri kümesinden 77 ek öznitelik çıkarılmıştır.

Önerilen yöntemin geçerliliğini karşılaştırmak ve test etmek için beş farklı senaryo oluşturulmuştur ve Çizelge 3.7'de özetlenmiştir.

İlk senaryoda, orijinal veri kullanılmıştır. İkinci senaryoda, çıkarılmış veri kullanılmıştır. Üçüncü senaryoda, birleştirilmiş veri kullanılmıştır. Bu senaryoda, orijinal veri ile çıkartılmış veri, veri birleştirme işlemi olan Feature Union yöntemiyle gerçekleştirilmiştir. Sonuç olarak, Şarap Kalitesi-Kırmızı veri kümesi için birinci senaryoda 11, ikinci senaryoda 69, üçüncü senaryoda ise 80 öznitelik kullanılmıştır. Benzer şekilde, Kalp Hastalığı veri kümesi için birinci senaryoda 13, ikinci senaryoda 77 ve üçüncü senaryoda 90 öznitelik kullanılmıştır.

Önerilen modelin test edilmesi için, birleştirilmiş veri üç alt kümeye bölünerek dördüncü bir

veri kümesi oluşturulmuştur. Şarap Kalitesi-Kırmızı veri kümesi için, Fisher Skoru kullanılarak 80 öznitelik içinden en önemli 32 öznitelik seçilmiş ve bu öznitelikler her üç alt kümeye eklenmiştir. Kalan 48 öznitelik, üç alt kümeye rastgele dağıtılmıştır. Kalp Hastalığı veri kümesi için ise Fisher Skoru ile 90 öznitelikten en önemli 69 öznitelik seçilmiş ve her üç alt kümeye eklenmiştir. Kalan 21 öznitelik, üç alt kümeye rastgele dağıtılmıştır. Bu şekilde, her iki veri kümesi için dördüncü senaryo, kolektif öğrenme süreci için üçer alt küme (şarap kalitesi-kırmızı için 48 ve kalp hastalığı için 76 öznitelik içeren alt kümeler) olarak oluşturulmuştur.

Çizelge 3.7. Veri senaryoları

Senaryo Numarası	Adı	Veri (Öznitelik * Örnek)	
		Şarap Kalitesi-Kırmızı	Kalp Hastalığı
1	Orijinal Veri	11*1599	13*303
2	Çıkarılmış Veri	69*1599	77*303
3	Birleştirilmiş Veri	80*1599	90*303
4	Dikey Alt Kümeler (3x)	48*1599	76*303
5	Rastgele Yatay Alt Kümeler (3x)	80*1599	90*303

İlgili alt küme parametreleri, GridSearch kullanılarak optimize edilmiştir. Çalışmada en iyi sonuç alınan küme parametreleri kullanılarak hem ön öznitelik mühendisliği hem de model eğitimi işlemleri gerçekleştirilmiştir.

Önerilen modelin mevcut Bootstrap Aggregation tekniği ile karşılaştırılması için birleştirilmiş veri kümesi rastgele yatay alt kümelere bölünerek test edilmiştir. Oluşturulan veri senaryoları Çizelge 3.7'de 5. sırada sunulmaktadır. Çizelge 4.3'te yapılan tüm testlere kıyasla birleştirilmiş veri daha iyi sonuç vermiştir. Dolayısıyla, kümeleme işlemini uygulamak üzere birleştirilmiş veri seçilmiştir.

3.5. Meta Modelin Eğitimi ve Mevcut Yaklaşımlarla Kıyaslama Senaryoları

Bu aşamada, veriler hazırlanmış olup, birçok farklı kombinasyon ile uygulanabilecek kolektif öğrenme ve oylama mekanizmalarının gerçekleştirilmesi gerekmektedir. Örneğin, üç modelin hazırlanmış verilerle eğitileceği bir sert oylama ortamı seçilmiştir.

Hazırlanan üç model, senaryoları Çizelge 3.7'de gösterilen verilerle ayrı ayrı eğitildikten sonra, değerlendirme sonuçları bir sert oylama mekanizması ile birleştirilmiş ve nihai tahmin elde edilmiştir.

Kolektif Öğrenme için Destek Vektör Sınıflandırıcısı (SVC), Lojistik Regresyon ve Rastgele Orman (Random Forest) olmak üzere üç temel model seçilmiştir. Bu modeller öncelikle, herhangi bir bölme yapılmadan hazırlanan tüm veri setinde ayrı ayrı eğitilmiştir (Bkz. Çizelge 3.8'de orijinal, çıkartılmış ve birleştirilmiş verileri). Daha sonra, meta-modeller, birleştirilmiş veriden hazırlanmış alt kümeler üzerinde yatay oylama ve dikey oylama mekanizmaları ile eğitilmiş ve test verisi kullanılarak değerlendirilmiştir.

Bu modellerin çıktıları, bir sert oylama algoritması kullanılarak birleştirilmiş ve her bir örnek için karşılaştırmaya hazır nihai bir tahmin elde edilmiştir. Ayrıca, boosting yöntemine ait XGBoost algoritması kullanarak da veri seti başına 3 toplam 6 test gerçekleştirilmiştir.

Çizelge 3.8'de ilk üç satırda (SVC, LR, RF) yer alan ve "Orijinal veri", "Çıkartılmış veri" ve "Birleştirilmiş veri" şeklinde sıralanan hücreler, kümeleme yapılmamış tam veriler üzerinden gerçekleştirilen 9 tekil eğitim senaryolarını, dört ve beşinci satırda (Sert Oylama ve XGBoost) 6 kolektif eğitim senaryolarını göstermektedir. SVC, LR ve RF modellerinin "Alt Kümelenmiş Birleştirilmiş Veri" sütununda, dikey ve yatay alt kümeleme yapılmış bölük veriler üzerinden 6 homojen ve 2 heterojen kolektif eğitim senaryolarını göstermektedir. Çizelge 3.8'in en alt satırında yer alan "Sert Oylama (SVC+LR+RF)" modelinin "Dikey Alt Kümeleme" altındaki hücresi ise en iyi sonuçları veren önerilen metodolojiye ait dikey alt kümelenmiş ve kolektif öğretim yapılan eğitim senaryosuna aittir.

Özetlemek gerekirse, seçilen her bir açık kaynaklı veri kümesi ve bunların türevleri (çıkarılmış, birleştirilmiş ve bölünmüş varyasyonları) için toplamda 23 test gerçekleştirilmiştir. Bu testlerin 15 (3 adeti sert oylama öğrenme için, 3 adeti XGBoost algoritması için ve geriye kalan 9 adeti temel modeller için) adeti alt küme kullanmayan eğitimlerin gerçekleştirilmesi için oluşturulmuştur. Geriye kalan 8 (4 tanesi Bootstrap yöntemiyle rastgele alt kümelere bölünmüş, 4 tanesi ise önerilen dikey alt küme yöntemiyle alt kümelere bölünmüştür) adeti ise alt küme kullanan eğitimlerin gerçekleştirilmesi için oluşturulmuştur.

Bölüm 4'te bu testlerin her biri detaylı bir şekilde incelenmiş ve sonuçlar hem tablo hem de grafik şeklinde paylaşılmıştır.

Çizelge 3.8. Uygulanan Test Senaryoları

<table>
<tr><td rowspan="3">Modeller</td><td colspan="3">Veri Senaryolarında Yapılan Testler</td></tr>
<tr><td rowspan="2">Alt Kümelenmemiş Veri</td><td colspan="2">Alt Kümelenmiş Birleştirilmiş Veri</td></tr>
<tr><td>Rastgele Yatay Alt Kümeleme</td><td>Dikey Alt Kümeleme</td></tr>
<tr><td rowspan="3">SVC</td><td>Orijinal veri</td><td rowspan="3">Homojen
3 SVC meta-model
(3 Alt Küme)</td><td rowspan="3">Homojen
3 SVC meta-model
(3 Alt Küme)</td></tr>
<tr><td>Çıkartılmış veri</td></tr>
<tr><td>Birleştirilmiş veri</td></tr>
<tr><td rowspan="3">LR</td><td>Orijinal veri</td><td rowspan="3">Homojen
3 LR meta-model
(3 Alt Küme)</td><td rowspan="3">Homojen
3 LR meta-model
(3 Alt Küme)</td></tr>
<tr><td>Çıkartılmış veri</td></tr>
<tr><td>Birleştirilmiş veri</td></tr>
<tr><td rowspan="3">RF</td><td>Orijinal veri</td><td rowspan="3">Homojen
3 RF meta-model
(3 Alt Küme)</td><td rowspan="3">Homojen
3 RF meta-model
(3 Alt Küme)</td></tr>
<tr><td>Çıkartılmış veri</td></tr>
<tr><td>Birleştirilmiş veri</td></tr>
<tr><td rowspan="3">Sert Oylama (SVC+LR+RF)</td><td>Orijinal veri</td><td rowspan="3">SVC+LR+RF
meta-model
(3 Alt Küme)</td><td rowspan="3">SVC+LR+RF
meta-model
(3 Alt Küme)</td></tr>
<tr><td>Çıkartılmış veri</td></tr>
<tr><td>Birleştirilmiş veri</td></tr>
<tr><td rowspan="3">XGBoost</td><td>Orijinal veri</td><td rowspan="3">-</td><td rowspan="3">-</td></tr>
<tr><td>Çıkartılmış veri</td></tr>
<tr><td>Birleştirilmiş veri</td></tr>
</table>

İlgili test senaryoları oluşturulurken hem homojen hem de heterojen model sistemleri aynı zamanda farklı boyutlarda oluşturulmuş veri setleri çapraz olarak kullanılmıştır. Bu sayede, Bölüm 3.3'te açıklandığı üzere, önerilen yöntemin çok sınıflı veri kümeleri, az örnekli veri kümeleri, hedef değişkeni dengesiz olan veri kümeleri ve az öznitelikli veri kümeleri üzerindeki yetkinliği geniş bir yelpazede test edilebilmiştir.

4. BULGULAR VE DEĞERLENDİRME

Eğitilmek üzere olan tüm modelleri Çizelge 3.8'de listelenen farklı model eğitim senaryoları kullanarak karşılaştırmak amacıyla, her model Kesinlik (Denklem 4.1), Duyarlılık (Denklem 4.2) ve F1-Skoru (Denklem 4.3) ile değerlendirilmiştir.

$$Kesinlik\ (Precission) = \frac{TP}{TP + FP} \tag{4.1}$$

$$Duyarlılık\ (Recall) = \frac{TP}{TP + FN} \tag{4.2}$$

$$\mathrm{F}_1 - Skoru = 2 * \frac{Kesinlik * Duyarlılık}{Kesinlik + Duyarlılık} \tag{4.3}$$

Değerlendirme iki aşamalı olup, ilk test, önerilen modelin Sert Oylama yöntemi ve XGBoost yöntemi ile karşılaştırılmasını amaçlamıştır ve detayları Çizelge 4.1, Çizelge 4.2, Çizelge 4.3'te paylaşılmıştır. İkinci test ise, önerilen modelin Bootstrap Aggregation yöntemi ile karşılaştırılmasını hedeflemiştir, detayları da Çizelge 4.4'te sunulmuştur.

Çizelge 4.1. Sert oylama ve XGBoost ile önerilen dikey oylama yönteminin karşılaştırılmasının kesinlik sonuçları

Model Adı		Farklı Veri Senaryolarından Elde Edilen Kesinlik Sonuçları			
		Orijinal Veri	Çıkartılmış Veri	Birleştirilmiş Veri	3 Dikey Alt Kümelenmiş
Şarap Kalitesi	SVC	0,5308	0,6138	0,6229	-
	LR	0,6008	0,6092	0,6133	-
	RF	0,6283	0,6429	0,6262	-
	XGBoost	0,6374	0,6393	0,6418	-
	Sert Oylama	0,6512	0,6368	0,6533	0,6673
Kalp Hastalığı	SVC	0,9016	0,8857	0,9034	-
	LR	0,9032	0,9185	0,9224	-
	RF	0,9016	0,9032	0,8888	-
	XGBoost	0,8438	0,8484	0,8750	-
	Sert Oylama	0,9032	0,9185	0,9185	0,9344

Çizelge 4.1 ve Çizelge 4.2'de verilen kesinlik ve duyarlılık değerleri Denklem 4.1 ve 4.2'de verilen formüllere göre deneysel çalışmalarda elde edilen sonuçlardan hesaplanmıştır. İlgili çizelgelerde önerilen dikey oylama yöntemi ile mevcut sert oylama (Hard Voting) ve XGboost tekniğini her iki veri kümesi üzerinde karşılaştırmak amacıyla gerçekleştirilen test sonuçları, değerlendirme metriği F1-Skor değerleriyle ölçülmüştür.

Çizelge 4.2. Sert oylama ve XGBoost ile önerilen dikey oylama yönteminin karşılaştırılmasının duyarlılık sonuçları

Model Adı		Farklı Veri Senaryolarından Elde Edilen Duyarlılık Sonuçları			
		Orijinal Veri	Çıkartılmış Veri	Birleştirilmiş Veri	3 Dikey Alt Kümelenmiş
Şarap Kalitesi	SVC	0,6312	0,6343	0,6437	-
	LR	0,6218	0,6281	0,6312	-
	RF	0,6583	0,6587	0,6562	-
	XGBoost	0,6562	0,6562	0,6644	-
	Sert Oylama	0,6281	0,6500	0,6687	0,66739
Kalp Hastalığı	SVC	0,9016	0,8852	0,9016	-
	LR	0,9016	0,9180	0,9180	-
	RF	0,9016	0,9016	0,8852	-
	XGBoost	0,8709	0,9032	0,9333	-
	Sert Oylama	0,9016	0,9180	0,9180	0,9344

Çizelge 4.3'te verilen F1-Skor değerleri Denklem 4.3'te verilen formüle göre hesaplanmıştır. Çalışmanın bütününde yapılan karşılaştırmalar bu F1-Skorları üzerinden yapılmıştır. Çizelge 4.3, önerilen dikey oylama yöntemi ile mevcut sert oylama ve XGBoost tekniğini her iki veri kümesi üzerinde karşılaştırmak amacıyla gerçekleştirilen test sonuçları, değerlendirme metriği F1-Skor değerlerini içermektedir.

Kolektif öğrenmenin performansını göstermek için, meta-modeller (Heterojen Sert Oylama, yani SVC+LR+RF) ile temel modelleri (SVC, LR, RF- ayrı ayrı) karşılaştırılmıştır. Ayrıca önerilen yöntemin bahsedilen standart yöntemler olan Heterojen Sert Oylama, temel modeller ve XGBoost artırma tabanlı kolektif öğrenme yöntemi ile karşılaştırılması için elde edilen sonuçlar paylaşılmıştır. Bu modeller, orijinal, öznitelik çıkarımı uygulanmış ve

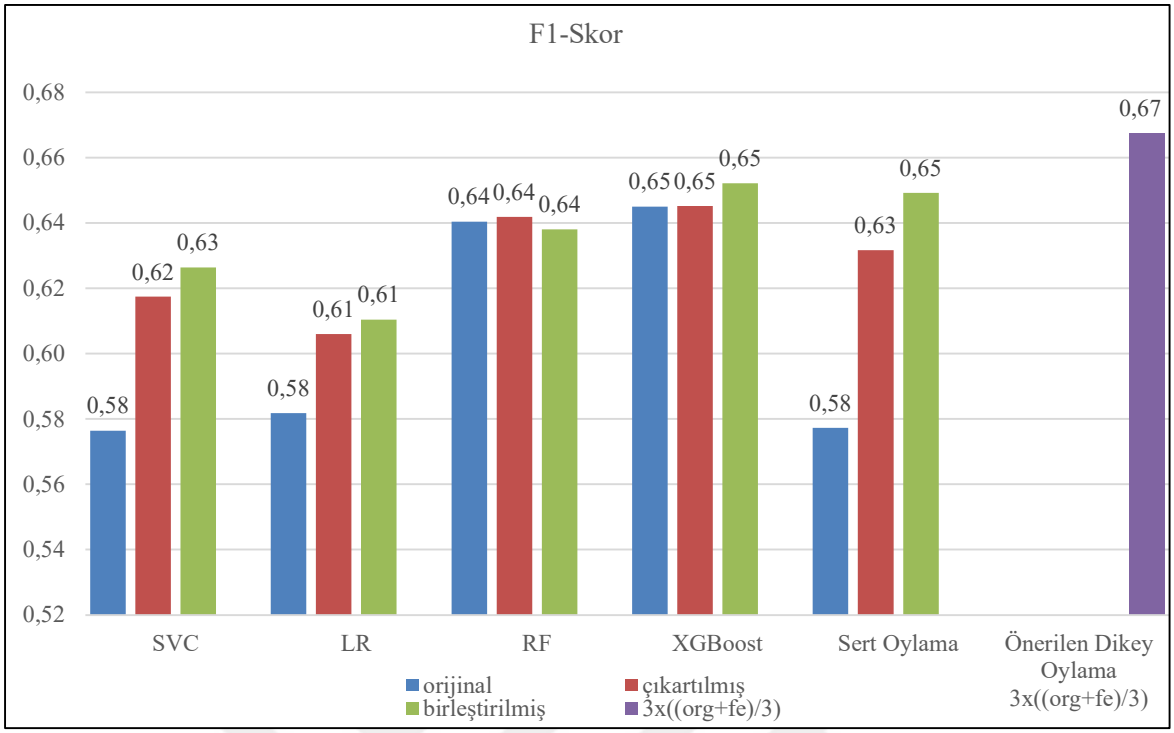
birleştirilmiş veri üzerinde eğitilmiştir.

Çizelge 4.3. Sert oylama ve XGBoost ile önerilen dikey oylama yönteminin karşılaştırılmasının f1-skor sonuçları

Model Adı		Farklı Veri Senaryolarından Elde Edilen F1-Skor Sonuçları			
		Orijinal Veri	Çıkartılmış Veri	Birleştirilmiş Veri	3 Dikey Alt Kümelenmiş
Şarap Kalitesi	SVC	0,5764	0,6174	0,6264	-
	LR	0,5818	0,6060	0,6104	-
	RF	0,6404	0,6418	0,6380	-
	XGBoost	0,6450	0,6452	0,6521	-
	Sert Oylama	0,5773	0,6316	0,6492	0,66739
Kalp Hastalığı	SVC	0,9016	0,8852	0,9015	-
	LR	0,9014	0,9180	0,9180	-
	RF	0,9016	0,9014	0,8848	-
	XGBoost	0,8571	0,8750	0,9032	-
	Sert Oylama	0,9014	0,9180	0,9180	0,9344

Ardından, heterojen sert oylama modeli, birleştirilmiş verinin üç dikey alt kümeye bölünmesiyle oluşturulan yeni veri setleri ile tekrar eğitilmiştir. Wine Quality-Red veri kümesi için, 32 öznitelik alt kümeler arasında ortak kullanılmış, kalan 48 öznitelik ise rastgele dağıtılmıştır. Heart Disease veri kümesi için ise 69 öznitelik alt kümeler arasında ortak kullanılmış, kalan 21 öznitelik ise rastgele dağıtılmıştır (önerilen dikey alt kümeleme tekniği).

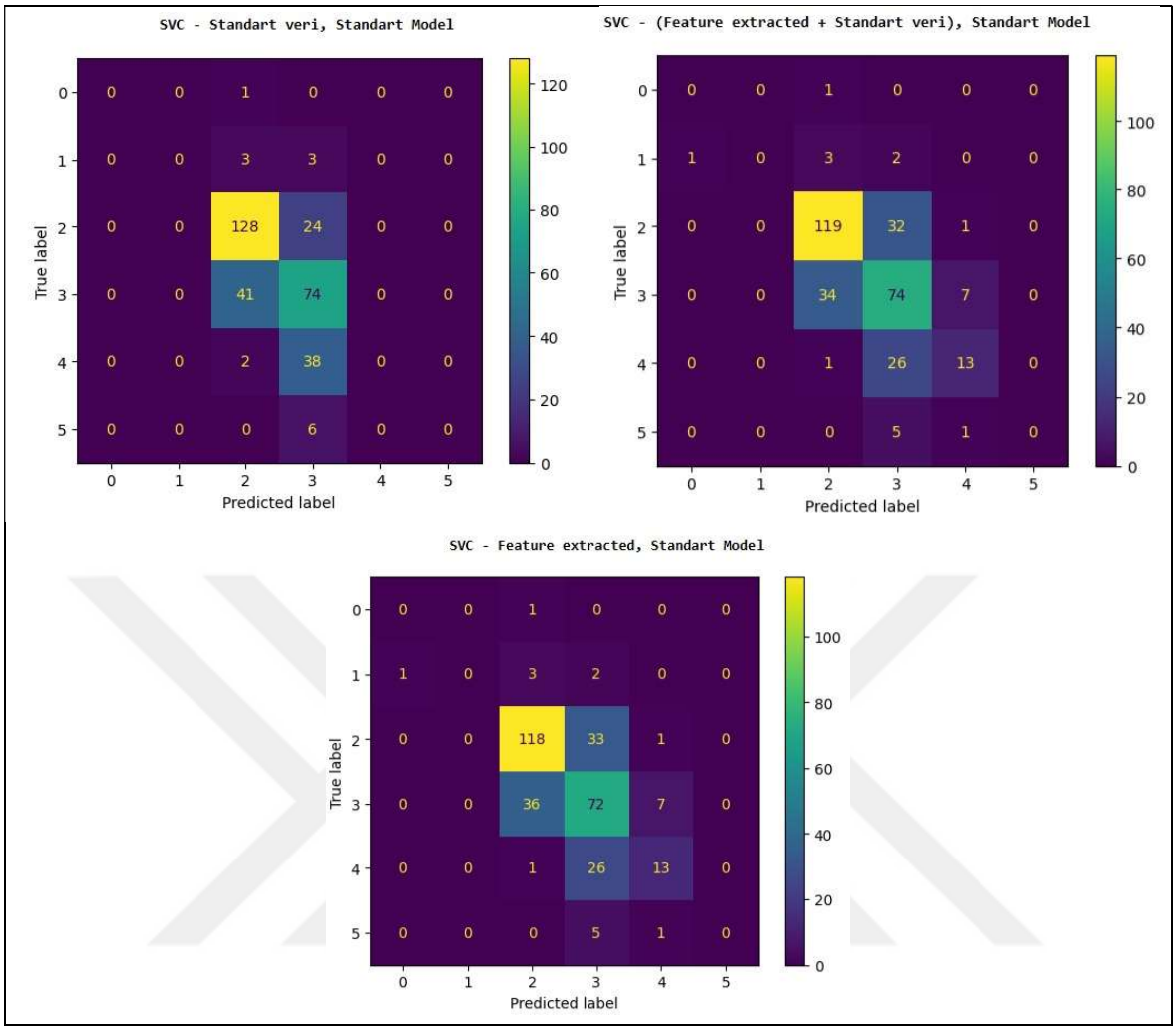
Şekil 4.1 ve Çizelge 4.3'te gösterilen sonuçlar, önerilen dikey alt kümeleme tabanlı kolektif öğrenme modelinin, mevcut sert oylama yöntemi ve XGBoost yöntemine kıyasla F1-Skoru, Kesinlik ve Duyarlılık açısından daha yüksek bir performans sağladığını göstermektedir.



Şekil 4.1. Şarap Kalitesi-Kırmızı veri seti kullanılarak sert oylama ve XGBoost'un önerilen dikey oylama yöntemiyle karşılaştırılması sonuçlarını açıklayan grafik

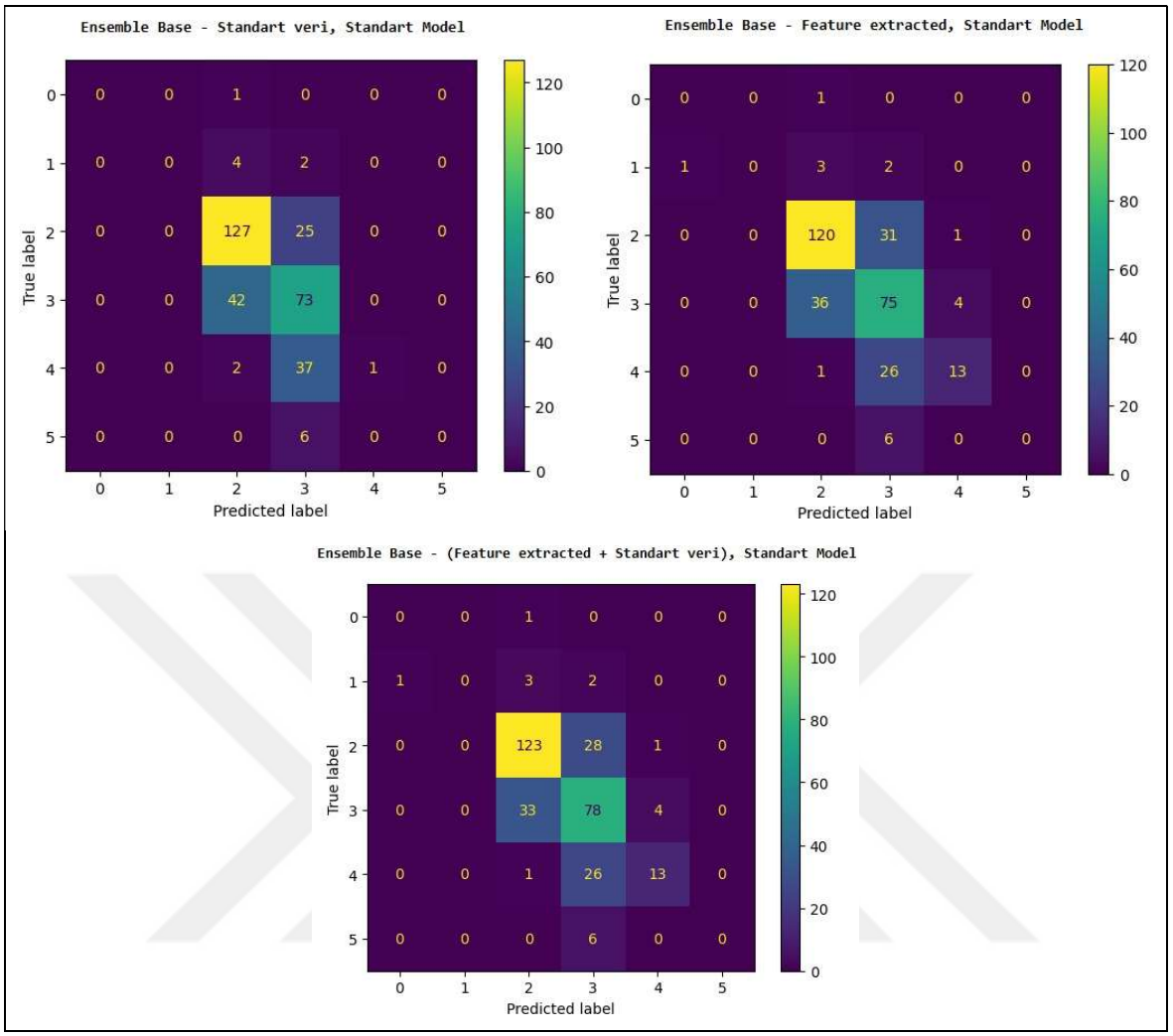
F1-Skoru, kesinlik ve duyarlılık metrikleri dışında alınan sonuçlarının efektifliğini anlayabilmek için karmaşıklık matrislerinin de incelenmesine gerek duyulmuştur.

Şekil 4.2, Şekil 4.3 ve Şekil 4.4 Şarap Kalitesi-Kırmızı veri seti üzerinde farklı veri senaryoları ile yapılan eğitimlerin karmaşıklık matrisleri ve bu matrisler üzerinden yapılan çıkarımlar incelenmiştir.



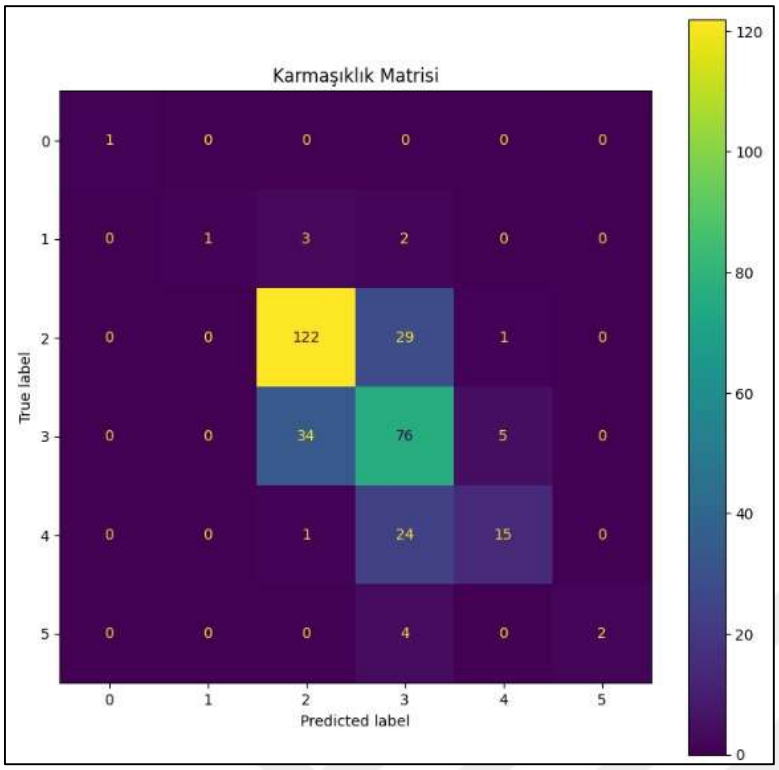
Şekil 4.2. SVC tekil modelinin farklı veri senaryolardaki karmaşıklık matrisi

Şekil 4.2'de, verinin dengesiz dağılımı sebebiyle az sayıda örnek bulunan sınıflarda hatalar gözlemlenmektedir. Bu inceleme sırasında, gösterilen üç karmaşıklık matrisleri arasında birleştirilmiş veri üzerinde yapılan SVC modelinin eğitimi sonucu elde edilen karmaşıklık matrisi, önceki metrik çizelgelerinde alınan sonuçlar gibi, daha iyi sonuç vererek hedef değişkeni sınıflarını daha iyi ayırt edebilmiştir ve detayları Çizelge 4.1, Çizelge 4.2 ve Çizelge 4.3'te sunulmuştur. Aynı durum, diğer LR ve RF tekil modeller için de gözlemlenmiştir.



Şekil 4.3. Heterojen (SVC+LR+RF) kolektif modelinin farklı veri senaryolardaki karmaşıklık matrisi

Şekil 4.2 ile Şekil 4.3 karşılaştırıldığında, öznitelikle birleştirilmiş veri üzerinde kümeleme yönteminin az sayıda örneğe sahip hedef değişken sınıflarında daha yakın sonuçlar elde edilmiştir.



Şekil 4.4. Önerilen dikey alt kümeleme yönteminin heterojen (SVC+LR+RF) kolektif modelinin birleştirilmiş veri senaryosundaki karmaşıklık matrisi

Şekil 4.4'te gösterilen karmaşıklık matrisi incelendiğinde, heterojen kolektif modelinin sonuçlarına göre hem genel olarak hem de az örneğe sahip sınıflar özelinde daha başarılı sonuçlar elde edildiği gözlemlenmiştir.

İkinci değerlendirme aşaması, önerilen modelin Bootstrap Aggregation yöntemi ile karşılaştırılmasını amaçlamaktadır. Çizelge 4.4, her iki veri kümesi üzerinde gerçekleştirilen testlerin sonuçlarını paylaşarak, önerilen dikey oylama yöntemi ile mevcut Bootstrap Aggregation tekniğini karşılaştırmaktadır.

Bootstrap Aggregation tekniğinde, veri bölme işlemi yatay alt kümeler kullanılarak gerçekleştirilmekte ve eğitim süreci homojen öğrenme algoritmaları ile yürütülmektedir. Bölüm 2.2.1'te incelendiği gibi, yatay alt kümeleme rastgele bölmelere ayrılmakta olup, bu yöntemin bazı dezavantajları bulunmaktadır.

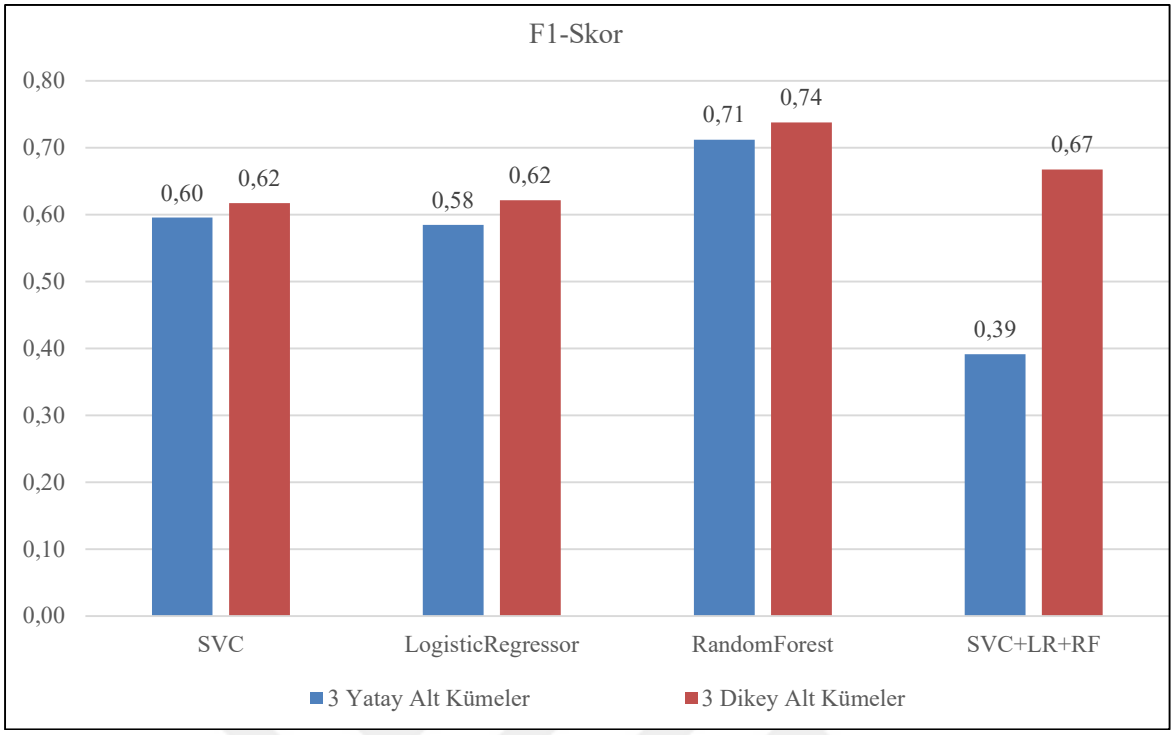
Bootstrap Aggregation yönteminin performansını değerlendirmek için, dikey alt kümeleme yöntemi ile karşılaştırılmıştır. Bu amaçla, homojen meta-modeller (SVC+SVC+SVC, LR+LR+LR, RF+RF+RF) rastgele bölünmüş üç yatay alt küme ve birleştirilmiş veriden

oluşturulan üç dikey alt küme üzerinde eğitilmiştir. Ardından, heterojen meta-modeller (SVC+LR+RF) aynı şekilde üç rastgele bölünmüş yatay alt küme ve üç dikey alt küme ile eğitilerek test edilmiştir.

Çizelge 4.4. Bootstrap Aggregation yönteminin önerilen dikey oylama yöntemiyle karşılaştırılmasının deneysel sonuçları

Model Adı			Birleştirilmiş Veriden Oluşturulmuş Alt Kümeler (F1-Skor)	
			3 Rastgele Yatay Alt Kümeleri	3 Dikey Alt Kümeleri
Şarap Kalitesi	Homojen Modeller	SVC+SVC+SVC	0,5954	0,6172
		LR+LR+LR	0,5845	0,6215
		RF+RF+RF	0,7119	0.7377
	Heterojen Modeller	SVC+LR+RF	0,3914	0,66739
Kalp Hastalığı	Homojen Modeller	SVC+SVC+SVC	0,8195	0,8848
		LR+LR+LR	0,8687	0,9016
		RF+RF+RF	0,8848	0,9014
	Heterojen Modeller	SVC+LR+RF	0,4692	0,9344

Hem Şekil 4.1'de hem de Çizelge 4.4'te gösterildiği gibi, önerilen modelin sonuçları F1-Skor metriği açısından her iki alternatiften daha iyi performans göstermektedir.



Şekil 4.5. Bootstrap Aggregation yönteminin Şarap Kalitesi-Kırmızı kullanılarak önerilen dikey oylama yöntemiyle karşılaştırılması sonuçlarını açıklayan grafik

Şekil 4.5 incelendiğinde hem tekil hem de kolektif öğrenimlerde dikey alt kümeleme yöntemi ile elde edilen sonuçlar daha yüksektir. Bunun dışında özellikle kolektif öğrenme sonuçlarında gözlemlenen fark yatay alt kümeleme yönteminin heterojen modeller ile uyumluluğunun dikey alt kümeleme kadar yüksek olmamasından kaynaklanmakta olduğu gözlemlenmiştir.

Dikey alt kümeleme tekniği, farklı veri türleri ve modellerle uyumluluğu sağlamak için çeşitli hiperparametrelere sahiptir. Bu hiperparametreler aşağıda listelenmiştir:

1. num_subsets: Alt küme sayısını belirler.
2. overlap_ratio: Alt kümeler arasında örtüşen özniteliklerin oranını temsil eder.
3. prepared_data: Alt kümelerin oluşturulacağı hazırlanmış veri kümesini belirtir.
4. random_state: Rastgelelik kontrolü sağlayarak tekrarlanabilir sonuçlar elde edilmesini sağlar.
5. drop_ratio: Veri kümesinden belirli bir oranla özniteliklerin çıkarılmasını (drop edilmesini) kontrol eder.
6. fisher_score: En önemli öznitelikleri seçmek için Fisher Skoru kullanılır.

Bu hiperparametreler sayesinde, istenen alt küme sayısını belirleme, alt kümeler arasındaki ortak özniteliklerin miktarını ayarlama, hangi veri kümesinin kullanılacağını seçme ve örtüşmeyen özniteliklerin alt kümeler arasında nasıl dağıtılacağını yönetme esnekliği sağlanmaktadır.

5. SONUÇ VE ÖNERİLER

Bu çalışmada, makine öğrenimi tekniklerinden biri olan kolektif öğrenme incelenmiş ve mevcut yaklaşımların ötesine geçerek yeni bir yöntem önerilmiştir. Önerilen yöntemde temel modellerin eğitimi için alt kümeler oluşturma sürecinde örnek tabanlı alt kümeleme (sample-based subsetting) yaklaşımı yerine, öznitelik tabanlı alt kümeleme (feature-based subsetting) tekniğinin kullanılması tezin yenilikçi yönü olarak çıkmaktadır.

Özniteliklere dayalı alt kümeleme yaklaşımı, her temel modelin farklı öznitelik gruplarına odaklanarak eğitilmesini sağlamaktadır. Böylece modeller, birbirinden bağımsız desenleri öğrenebilmekte ve kolektif yapının genel değerlendirme gücü artmaktadır. Bu süreçte, öznitelik tabanlı ön işleme adımı ile birlikte birden fazla öğrenme algoritması kullanılarak oluşturulan eğitim adımı bir arada değerlendirilmiştir.

Deneysel çalışmalar sonucunda, önerilen yaklaşımın hem tahmin doğruluğunu artırdığı hem de işlem süresi açısından maliyeti azalttığı gözlemlenmiştir. Bununla birlikte, yöntemin geliştirilmesine yönelik bazı gelecek çalışmalara dair öneriler de sunulmuştur.

Bu önerilerden ilki, özniteliklerin alt kümelere ayrılması sürecinde korelasyon matrisi, kovaryans analizi veya Fisher skorları gibi istatistiksel tekniklerden yararlanılarak veriye katkısı düşük veya gereksiz özniteliklerin elimine edilmesidir.

Ayrıca, büyük ölçekli veri kümeleriyle çalışılırken, öznitelik seçimi sırasında tüm veri yerine küçük bir örneklem alınarak sıralama yapılabilir ve bu sıralama sonuçları tüm veri setine genellenebilir. Önerilen yöntem, bu çalışmada Wine Quality – Red ve Heart Disease veri setleri ile sınanmış olmakla birlikte, ileride farklı alanlardan veri kümeleriyle test edilerek literatürdeki diğer yöntemlerle karşılaştırmalı analizler yapılabilir.

Buna ek olarak, önerilen sistem paralel programlama olanaklarıyla daha da hızlandırılabilir. Örneğin, veri kümesi alt kümelere ayrılarak farklı algoritma tabanlı modellerle multi-threading tabanlı paralel eğitim yapılabilir, zaman karmaşıklığını olumlu yönde etkileyerek sistemin ölçeklenebilirliğini artırabilir.

Son olarak, kolektif yapının sınıflandırma problemlerinin ötesine taşınması da mümkündür. Özellikle, bu tez kapsamında kullanılan veri kümesi için ve gelecekteki çalışmalarda, sınıflandırma yerine regresyon tabanlı modellerin entegrasyonu değerlendirilebilir. Bu iyileştirme önerisi, sunulan yöntemin kullanım yelpazesini genişletmekle birlikte probleme bağlı olarak daha iyi sonuçlar sergileyebilir.

KAYNAKLAR

1. Dietterich, T.G. (2000). An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization. *Machine Learning*, 40(2), 139-158.

2. Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5-32.

3. Freund, Y., Schapire, R.E. (1996). Experiments with a New Boosting Algorithm. *Thirteenth International Conference on Machine Learning (ICML 1996)*, 325-332.

4. Friedman, J.H. (2001). Greedy Function Approximation: A Gradient Boosting Machine. *Annals of Statistics*, 29(5), 1189-1232.

5. Fernández-Delgado, M., Cernadas, E., Barro, S., Amorim, D. (2014). Do We Need Hundreds of Classifiers to Solve Real World Classification Problems? *Journal of Machine Learning Research*, 15, 3133-3181.

6. Gomes, H.M., Barddal, J.P., Enembreck, F., Bifet, A. (2017). A Survey on Ensemble Learning for Data Stream Classification. *ACM Computing Surveys*, 50(2), 1-36.

7. Kulkarni, V.Y., Sinha, P.K. 2013. Random Forest Classifiers: A Survey and Future Research Directions. *International Journal of Advanced Computing*, 36(1), 1144-1153.

8. Rokach, L. (2016). Decision Forest: Twenty Years of Research. *Information Fusion*, 27, 111-125.

9. Woźniak, M., Graña, M., Corchado, E. (2014). A Survey of Multiple Classifier Systems as Hybrid Systems. *Information Fusion*, 16, 3-17.

10. Polikar, R. (2006). Ensemble Based Systems in Decision Making. *IEEE Circuits and Systems Magazine*, 6(3), 21-45.

11. Ren, Y., Zhang, L., Suganthan, P.N. (2016). Ensemble Classification and Regression – Recent Developments, Applications and Future Directions (Review Article). *IEEE Computational Intelligence Magazine*, 11(1), 41-53.

12. Rokach, L. (2010). Ensemble-based Classifiers. *Artificial Intelligence Review*, 33(1), 1-39.

13. Sagi, O., Rokach, L. (2018). Ensemble Learning: A Survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4), e1249.

14. Breiman, L. 1996. Bagging Predictors. *Machine Learning*, 24(2), 123-140.

15. Freund, Y., Schapire, R.E. (1996). Experiments with a New Boosting Algorithm. *Thirteenth International Conference on Machine Learning (ICML 1996)*, 325-332.

16. Wolpert, D.H. (1992). Stacked Generalization. *Neural Networks*, 5(2), 241-259.

17. Seni, G., Elder, J.F. (2010). *Ensemble Methods in Data Mining: Improving Accuracy Through Combining Predictions*. Cham: Springer.

18. Chen, T., Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2016)*, 785-794.

19. Dietterich, T.G. (2000). Ensemble Methods in Machine Learning. İçinde J. Kittler, F. Roli (Ed.), *Multiple Classifier Systems Lecture Notes in Computer Science*, 1857(1), 1-15. Berlin: Springer.

20. Dietterich, T.G. (2000). An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization. *Machine Learning*, 40(2), 139-158.

21. İnternet: Python Programlama Dili. URL: https://www.python.org. Son Erişim Tarihi: 7 Mart 2025.

22. İnternet: Project Jupyter (Jupyter Notebook). URL: https://jupyter.org. Son Erişim Tarihi: 7 Mart 2025.

23. İnternet: Anaconda (Python Dağıtımı). URL: https://www.anaconda.com. Son Erişim Tarihi: 7 Mart 2025.

24. McKinney, W. (2010). Data Structures for Statistical Computing in Python. *9th Python in Science Conference (SciPy 2010)*, 51-56.

25. Harris, C.R., Millman, K.J., van der Walt, S.J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N.J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M.H., Brett, M., Haldane, A., Fernández del Río, J., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., Oliphant, T.E. (2020). Array Programming with NumPy. *Nature*, 585(7825), 357-362.

26. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830.

27. İnternet: scikit-learn MinMaxScaler Sınıfı Dokümanı. URL: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html. Son Erişim Tarihi: 7 Mart 2025.

28. İnternet: scikit-learn StandardScaler Sınıfı Dokümanı. URL: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html. Son Erişim Tarihi: 7 Mart 2025.

29. İnternet: scikit-learn RobustScaler Sınıfı Dokümanı. URL: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.RobustScaler.html. Son Erişim Tarihi: 7 Mart 2025.

30. İnternet: scikit-learn QuantileTransformer Sınıfı Dokümanı. URL: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.QuantileTransformer.html. Son Erişim Tarihi: 7 Mart 2025.

31. İnternet: scikit-learn KBinsDiscretizer Sınıfı Dokümanı. URL: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.KBinsDiscretizer.html. Son Erişim Tarihi: 7 Mart 205.

32. İnternet: scikit-learn PCA Sınıfı Dokümanı. URL: https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html. Son Erişim Tarihi: 7 Mart 2025.

33. İnternet: scikit-learn TruncatedSVD Sınıfı Dokümanı. URL: https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.TruncatedSVD.html. Son Erişim Tarihi: 7 Mart 2025.

34. İnternet: scikit-learn FeatureUnion Sınıfı Dokümanı. URL: https://scikit-learn.org/stable/modules/generated/sklearn.pipeline.FeatureUnion.html. Son Erişim Tarihi: 7 Mart 2025.

35. İnternet: Fisher Skoru ile Özellik Seçimi Yöntemi. URL: https://ranasinghiitkgp.medium.com/implementing-feature-selection-methods-for-machine-learning-bfa2e4b4e02. Son Erişim Tarihi: 7 Mart 2025.

36. İnternet: scikit-learn LogisticRegression Sınıfı Dokümanı. URL: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html. Son Erişim Tarihi: 7 Mart 2025.

37. İnternet: scikit-learn SVC Sınıfı Dokümanı. URL: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html. Son Erişim Tarihi: 7 Mart 2025.

38. İnternet: scikit-learn RandomForestClassifier Sınıfı Dokümanı. URL: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html. Son Erişim Tarihi: 7 Mart 2025.

39. İnternet: scikit-learn GridSearchCV Sınıfı Dokümanı. URL: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html. Son Erişim Tarihi: 7 Mart 2025.

40. İnternet: UCI Makine Öğrenmesi Deposu – Wine Quality Veri Kümesi. URL: https://archive.ics.uci.edu/dataset/186/wine+quality. Son Erişim Tarihi: 18 Mart 2025.

41. İnternet: UCI Makine Öğrenmesi Deposu – Heart Disease Veri Kümesi. URL: https://archive.ics.uci.edu/dataset/45/heart+disease. Son Erişim Tarihi: 18 Mart 2025.

42. İnternet: Intel Core i7-10875H İşlemcisi Teknik Özellikleri. URL: https://www.intel.com/content/www/us/en/products/sku/202329/intel-core-i710875h-processor-16m-cache-up-to-5-10-ghz/specifications.html. Son Erişim Tarihi: 18 Mart 2025.

43. İnternet: Samsung 16GB DDR4-2666 SO-DIMM Bellek Özellikleri. URL: https://semiconductor.samsung.com/dram/module/sodimm/m471a2k43cb1-ctd/. Son Erişim Tarihi: 18 Mart 2025.



Gazili olmak ayrıcalıktır