SINIFLANDIRMA PERFORMANS DEĞERLENDİRME KRİTERLERİNİN MELEZ KÜMELEME YÖNTEMLERİ KULLANILARAK İYİLEŞTİRİLMESİ

Elif GÜLERYÜZ

T.C.
BURSA ULUDAĞ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

SINIFLANDIRMA PERFORMANS DEĞERLENDİRME KRİTERLERİNİN MELEZ KÜMELEME YÖNTEMLERİ KULLANILARAK İYİLEŞTİRİLMESİ

Elif GÜLERYÜZ
0000-0001-9760-7631

Doç. Dr. Duygu YILMAZ EROĞLU
(Danışman)

YÜKSEK LİSANS TEZİ
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI

BURSA – 2022

TEZ ONAYI

Elif GÜLERYÜZ tarafından hazırlanan "SINIFLANDIRMA PERFORMANS DEĞERLENDİRME KRİTERLERİNİN MELEZ KÜMELEME YÖNTEMLERİ KULLANILARAK İYİLEŞTİRİLMESİ" adlı tez çalışması aşağıdaki jüri tarafından oy birliği ile Bursa Uludağ Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Danışman : Doç. Dr. Duygu YILMAZ EROĞLU

Başkan	**:**	Doç. Dr. Duygu YILMAZ EROĞLU 0000-0002-7730-2707 Uludağ Üniversitesi, Mühendislik Fakültesi, Endüstri Mühendisliği Anabilim Dalı	İmza
Üye	**:**	Doç. Dr. Ali Yurdun ORBAK 0000-0002-4921-4275 Uludağ Üniversitesi, Mühendislik Fakültesi, Endüstri Mühendisliği Anabilim Dalı	İmza
Üye	**:**	Doç. Dr. Aytaç YILDIZ 000-0002-0729-633X Bursa Teknik Üniversitesi, Mühendislik Fakültesi, Endüstri Mühendisliği Anabilim Dalı	İmza

Yukarıdaki sonucu onaylarım

Prof. Dr. Hüseyin Aksel EREN
Enstitü Müdürü
../../….

B.U.Ü. Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmasında;

- tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- ve bu tezin herhangi bir bölümünü bu üniversite veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.

28/12/2021

Elif GÜLERYÜZ

TEZ YAYINLANMA
FİKRİ MÜLKİYET HAKLARI BEYANI

Enstitü tarafından onaylanan lisansüstü tezin/raporun tamamını veya herhangi bir kısmını, basılı (kâğıt) ve elektronik formatta arşivleme ve aşağıda verilen koşullarla kullanıma açma izni Bursa Uludağ Üniversitesi'ne aittir. Bu izinle Üniversiteye verilen kullanım hakları dışındaki tüm fikri mülkiyet hakları ile tezin tamamının ya da bir bölümünün gelecekteki çalışmalarda (makale, kitap, lisans ve patent vb.) kullanım hakları tarafımıza ait olacaktır. Tezde yer alan telif hakkı bulunan ve sahiplerinden yazılı izin alınarak kullanılması zorunlu metinlerin yazılı izin alınarak kullandığını ve istenildiğinde suretlerini Üniversiteye teslim etmeyi taahhüt ederiz.

Yükseköğretim Kurulu tarafından yayınlanan "**Lisansüstü Tezlerin Elektronik Ortamda Toplanması, Düzenlenmesi ve Erişime Açılmasına İlişkin Yönerge**" kapsamında, yönerge tarafından belirtilen kısıtlamalar olmadığı takdirde tezin YÖK Ulusal Tez Merkezi / B.U.Ü. Kütüphanesi Açık Erişim Sistemi ve üye olunan diğer veri tabanlarının (Proquest veri tabanı gibi) erişimine açılması uygundur.

Danışman Adı-Soyadı Tarih	Öğrencinin Adı-Soyadı Tarih
İmza Bu bölüme kişinin kendi el yazısı ile okudum anladım yazmalı ve imzalanmalıdır.	İmza Bu bölüme kişinin kendi el yazısı ile okudum anladım yazmalı ve imzalanmalıdır.

ÖZET

Yüksek Lisans Tezi

SINIFLANDIRMA PERFORMANS DEĞERLENDİRME KRİTERLERİNİN MELEZ KÜMELEME YÖNTEMLERİ KULLANILARAK İYİLEŞTİRİLMESİ

Elif GÜLERYÜZ

Bursa Uludağ Üniversitesi
Fen Bilimleri Enstitüsü
Endüstri Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Duygu YILMAZ EROĞLU

Günümüzde teknolojinin gelişmesiyle birlikte, üretilen veri miktarı da büyük bir hızla artmaktadır. Üretilen verilerden anlamlı bilgiler elde edebilmek için verilerin işlenmesi gerekmektedir. Karmaşık veriyi anlamlı hale getirmek, bilgi edinme ve karar verebilme sürecini kısaltmaktadır. Bu sebeple araştırmacılar anlamlı bilgi edinmenin yollarını araştırmaktadır. Sınıflandırma, kümeleme gibi teknikleri içeren veri madenciliği yöntemleri de verilerin işlenmesi ile verilere anlam kazandırmak için gerekli işlemler yapılarak bilgiye dönüştürülmesini kapsamaktadır. Tez kapsamında yapılan çalışmanın amacı literatürde sıklıkla kullanılan veri setlerine doğrudan uygulanan sınıflandırma algoritmalarının performansları ve önerilen yöntemler ile sınıflandırma algoritmalarının performanslarını karşılaştırmak, ayrıca en iyi performansı gösteren yöntemi tespit etmektir. Çalışmada belirlenen orta ve büyük veri setleri ilk olarak ön işlemeye tabi tutulmuş ve sonrasında sadece ön işleme yapılmış veriler ile, K-ortalamalar kümeleme yöntemi ile, önerilen melez kümeleme yöntemi ile sınıflandırma işlemleri olmak üzere üç aşama halinde yöntemler uygulanıp elde edilen performans değerlendirme kriterleri karşılaştırılmıştır. Her aşamanın sonunda parametre optimizasyonu ile iyileştirme oranlarının daha yukarıya çekilmesi sağlanmıştır. Tez kapsamında önerilen melez yöntemin veri setlerinin önemli bir kısmında iyileştirme sağladığı gözlemlenmiştir.

Anahtar Kelimeler: Sınıflandırma, melez kümeleme, sınıflandırma performans değerlendirme kriterleri, parametre optimizasyonu
2022, vii + 74 sayfa.

ABSTRACT

MSc Thesis

IMPROVEMENT OF CLASSIFICATION PERFORMANCE EVALUATION CRITERIA USING HYBRID CLUSTERING METHODS

Elif GÜLERYÜZ

Bursa Uludağ University
Graduate School of Natural and Applied Sciences
Department of Industrial Engineering

Supervisor: Assoc. Prof. Dr. Duygu YILMAZ EROĞLU

Today, with the development of technology, the amount of data produced is increasing rapidly. In order to obtain meaningful information from the produced data, the data must be processed. Making complex data meaningful shortens the process of obtaining information and making decisions. For this reason, researchers are looking for ways to obtain meaningful information. Data mining methods, which include techniques such as classification and clustering, also include processing the data and transforming it into information by making the necessary operations to give meaning to the data. The aim of the study carried out within the scope of the thesis is to compare the performances of classification algorithms directly applied to the datasets frequently used in the literature and the performances of the proposed methods and classification algorithms, and also to determine the method with the best performance. The medium and large data sets determined in the study were first preprocessed, and then only the preprocessed data were compared with the performance evaluation criteria obtained by applying the methods in three stages: the K-means clustering method, the proposed hybrid clustering method and the classification processes. At the end of each stage, improvement rates were increased by parameter optimization. It has been observed that the hybrid method proposed in the thesis provides improvement in a significant part of the data sets.

Key words: Classification, hybrid clustering, classification performance evaulation criterias, parameter optimization
2022, vii + 74 pages.

TEŞEKKÜR

Lisans ve Yüksek Lisans eğitimim süresince yanımda olan ve bu tez çalışmasını gerçekleştirirken ihtiyaç duyduğum her an tecrübesini paylaşmaktan çekinmeyen, beni hiçbir zaman geri çevirmeyen ve samimiyet duygusu ile çalışma motivasyonumu artıran saygıdeğer danışmanım Doç. Dr. Duygu YILMAZ EROĞLU'na teşekkürlerimi sunarım.

Hayatım boyunca attığım her adımda yanımda olup bana güç veren çok sevdiğim, kıymetli aileme sonsuz teşekkürlerimi sunarım.

Elif GÜLERYÜZ
28/12/2021

İÇİNDEKİLER

Sayfa

ÖZET i
ABSTRACT ii
TEŞEKKÜR iii
SİMGELER ve KISALTMALAR DİZİNİ v
ŞEKİLLER DİZİNİ vi
ÇİZELGELER DİZİNİ vii
1. GİRİŞ 1
2. KURAMSAL TEMELLER ve KAYNAK ARAŞTIRMASI 4
3. MATERYAL ve YÖNTEM 16
3.1. Çelik Levha Arızası Veri Seti 17
3.1.1. Ön işleme 18
3.1.2. Sınıflandırma algoritmaları 19
3.1.3. Performans değerlendirme kriterleri 23
3.1.4. Sadece ön işleme yapılmış veriler ile sınıflandırma işlemi 25
3.1.5. K-ortalamalar kümeleme yöntemi kullanılarak sınıflandırma işlemi 26
3.1.6. Melez kümeleme yöntemi kullanılarak sınıflandırma işlemi 29
3.1.7. Parametre optimizasyonu 30
3.2. Diğer veri setleri 30
4. BULGULAR 31
4.1. Çelik Levha Arızası Veri Seti 31
4.1.1. Ön işleme 31
4.1.2. Sadece ön işleme yapılmış veriler ile sınıflandırma işlemi 34
4.1.3. K-ortalamalar kümeleme yöntemi kullanılarak sınıflandırma işlemi 41
4.1.4. Melez kümeleme yöntemi kullanılarak sınıflandırma işlemi 43
4.2. Diğer Veri Setleri 48
5. TARTIŞMA ve SONUÇ 52
5.1. Çelik Levha Arızası Veri Seti 52
5.2. Tüm Veri Setleri 55
KAYNAKLAR 58
EKLER 64
EK 1 Cam Veri Seti İçin Sonuçlar 65
EK 2 Tohumlar Veri Seti İçin Sonuçlar 68
EK 3 Kullanıcı Bilgisi Veri Seti İçin Sonuçlar 71
ÖZGEÇMİŞ 74

SİMGELER ve KISALTMALAR DİZİNİ

Simgeler	Açıklama
x	Gözlem değeri
μ	Ortalama değer
σ	Standart sapma

Kısaltmalar	Açıklama
CART	Sınıflandırma ve regresyon ağacı algoritması
CV	Çapraz doğrulama
FN	Yanlış negatif
FP	Yanlış pozitif
KNN	K-en yakın komşu algoritması
MLP	Yapay sinir ağları algoritması
RF	Rassal orman algoritması
RS	Rassal arama
SVM	Destek vektör makinesi algoritması
TN	Doğru negatif
TP	Doğru pozitif
Weka	Waikato Environment for Knowledge Analysis

ŞEKİLLER DİZİNİ

Sayfa

Şekil 2.1. Veri madenciliği deneyimli ve deneyimsiz öğrenme yöntemleri. 6
Şekil 3.1. Çalışmada kullanılacak yöntemlerin genel akışı..... 17
Şekil 3.2. Adaboost algoritma adımları gösterimi 22
Şekil 3.3. K-katlı çapraz doğrulama gösterimi..... 25
Şekil 4.1. Kolmogorov Smirnov test sonuçları 31
Şekil 4.2. Aykırı veri analizi sonuçları..... 32
Şekil 4.3. Korelasyon matrisi sonuçları 33
Şekil 4.4. Elbow yöntemi ile küme sayısının belirlenmesi 41
Şekil 4.5. Melez kümeleme dendrogramı 44

ÇİZELGELER DİZİNİ

Sayfa

Çizelge 3.1. Çalışmada kullanılan veri setleri ve özellikleri....16
Çizelge 3.2. Sınıf etiketleri dağılımı....17
Çizelge 3.3. Sınıflandırma performans değerlendirme kriterleri....24
Çizelge 3.4. Çelik Levha Arızası veri seti Weka çalışma performans sonuçları....28
Çizelge 4.1. 19 öznitelik seçimi sonucu belirlenen öznitelikler....34
Çizelge 4.2. Eğitim/test veri setlerine uygulanan sınıflandırma algoritmaları için doğruluk oranları....36
Çizelge 4.3. Algoritma bazında en yüksek doğruluk oranları....36
Çizelge 4.4. Öznitelik seçimine göre doğruluk oranları (Rassal orman algoritması)....37
Çizelge 4.5. Rassal Orman algoritması ile sadece ön işleme ile sınıflandırma işlemi için doğruluk oranları ve iyileştirme oranları....38
Çizelge 4.6. Uygulanan sınıflandırma algoritması için diğer performans kriterleri sonuçları....39
Çizelge 4.7. Rassal orman parametreleri ve seviyeleri....39
Çizelge 4.8. Sadece ön işleme yapılmış hali ile Rassal Orman algoritması parametre optimizasyonu için doğruluk oranları ve iyileştirme oranları....40
Çizelge 4.9. Rassal orman algoritması ile K-ortalamalar kümeleme yöntemi kullanılarak sınıflandırma işlemi sonuçları....42
Çizelge 4.10. K-ortalamalar kümeleme yöntemine göre Rassal Orman algoritması parametre optimizasyonu ile doğruluk oranları ve iyileştirme oranları....43
Çizelge 4.11. Rassal orman algoritması ile melez kümeleme yöntemi kullanılarak sınıflandırma işlemi sonuçları (k = 3)....45
Çizelge 4.12. Melez kümeleme yöntemine göre Rassal Orman algoritması parametre optimizasyonu ile doğruluk oranları ve iyileştirme oranları (k = 3)....46
Çizelge 4.13. Rassal orman algoritması ile melez kümeleme yöntemi kullanılarak sınıflandırma işlemi sonuçları (k = 4)....47
Çizelge 4.14. Melez kümeleme yöntemine göre Rassal Orman algoritması parametre optimizasyonu ile doğruluk oranları ve iyileştirme oranları (k = 4)....48
Çizelge 4.15. Omurga veri seti Weka çalışma performans sonuçları....49
Çizelge 4.16. Omurga veri seti için algoritma bazında en yüksek doğruluk oranları....49
Çizelge 4.17. Omurga veri seti kesinlik, anma ve F-ölçüt sonuçları....50
Çizelge 4.18. Omurga veri seti parametreleri ve seviyeleri....50
Çizelge 4.19. Omurga veri seti için doğruluk oranları....51
Çizelge 5.1. Eğitim/test veri seti üzerinde Rassal Orman doğruluk oranı(%)....53
Çizelge 5.2. K-katlı çapraz doğrulama ile Rassal Orman doğruluk oranları(%)....54
Çizelge 5.3. Veri setleri için sınıflandırma performans göstergeleri iyileştirme oranları....55
Çizelge 5.4. Veri setleri için sonuçlar....56

1. GİRİŞ

Teknolojinin gelişmesiyle birlikte verinin üretimi, depolanması ve erişimi her geçen gün daha da kolaylaşmakta ve ucuzlamaktadır. Geçmişte kaydedilen verilerin biriktirilmesi, bu verilerden anlamlı bilginin çıkarılması ihtiyacını doğurmuştur. Verilerden anlamlı bilgi çıkarabilmek için de işlenmesi gerekmektedir. Belli bir amaç doğrultusunda işlenen veriler ile çok değerli çıkarımlar yapılabilir. Veri madenciliği de verilerin işlenmesi ile verilere anlam kazandırmak için gerekli işlemler yapılarak bilgiye dönüştürülmesi işlemidir. Veri analizi araçları kullanılarak veri kümeleri arasındaki gizli ilişkiler ortaya çıkarılabilmektedir (Tan, Steinbach ve Kumar, 2016). Veri madenciliği, istatistiksel modelleme tekniklerini, matematiksel algoritmaları ve makine öğrenmesine dayalı çeşitli teknikleri içermektedir (Seifert, 2004).

Sınıflandırma, yaygın olarak kullanılan bir veri madenciliği tekniğidir. Amacı, yeni bir veri kümesinin etiketini, denetimli öğrenme yöntemlerini kullanarak tahmin etmektir. Veri tabanlarının hızlı büyümesi nedeniyle, veri tabanlarından etkin sınıflandırma kurallarının çıkarılması oldukça önemlidir. Sınıflandırma tekniği için çeşitli algoritmalar kullanılmaktadır (Usama, Gregory ve Padhraic, 1996).

Genel olarak veri sınıflandırmada, sınıflandırma başarılarını değerlendirmek için doğruluk, kesinlik, anma ve F-ölçütü kullanılmaktadır. Bu değerlendirmelerde geçen kısaltma anlamları şunlardır; True Positive - Doğru Pozitif doğru sınıflandırılmış pozitif örnek sayısı, True Negative - Doğru Negatif doğru sınıflandırılmış negatif örnek sayısı, False Positive - Yanlış Pozitif yanlış sınıflandırılmış pozitif örnek sayısı, False Negative - Yanlış Negatif yanlış sınıflandırılmış negatif örnek sayılarını ifade eder. Doğruluk değeri modelde doğru tahmin edilen nesnelerin toplam veri kümesine oranı ile hesaplanmaktır (Haltaş ve Alkan, 2016).

Yaygın olarak kullanılan bir diğer veri madenciliği tekniği ise kümelemedir. Çok karmaşık ve büyük sorunları çözmek için genelde sorunları küçük parçalara ayırmakla başlanır. Daha sonra her bir alt sorun çözülür ve işlem sonunda tüm çözümler birleştirilir ve sonuç elde edilmiş olur. Kümeleme de benzer mantıkla, veri setindeki

benzer nesnelerin gruplandırılması ve homojen alt gruplara ayrılmasını sağlar. Kümeleme analizinde yöntem fark etmeksizin süreç aynı şekilde işler ve her nesne var olan kümelerle karşılaştırılır (Akbulut, 2006; Burn, Zrinji ve Kowalchulk, 1997).

En çok kullanılan kümeleme yöntemlerinden K-ortalamalar yöntemi, tutarlılığı ve hızı açısından öne çıkmaktadır ve hiyerarşik olmayan bir yapıya sahiptir. K-ortalamalar kümeleme yönteminde, benzer verilerin gruplanabilmesi için n adet veri, verilen k adet kümeye atanmaktadır. Diğer bir kümeleme yöntemi olan hiyerarşik kümeleme yönteminde ise, ağaç şeklinde bir kümeleme yapısı oluşturmak için veri kümelerinin farklı seviyelerde bölünmesi amaçlanmaktadır. Yukarıdan aşağıya (bölme) ve aşağıdan yukarıya (toplama) olmak üzere iki temel tekniği bulunmaktadır. Bölme tabanlı hiyerarşik kümeleme yönteminde, önce tüm veri kümesi bir küme olarak görülür ve daha sonra önceden tanımlanmış bir kurala göre gruplara ayırılır. Toplama hiyerarşik kümeleme yönteminde ise önce birçok başlangıç kümesi oluşturulur ve daha sonra bunlar farklı benzerlik hesaplama yöntemlerine göre birleştirilir. Hiyerarşik kümeleme yöntemlerinin sonuçları dendrogramlar ile gösterilebilmektedir (Wu, Peng, Lee, Leibnitz ve Xia, 2021).

Çeşitli teknikler kullanarak elde edilen bilgiler istenilen alanlarda kullanılabilir. Veri madenciliği istatistik, yapay zeka, makine öğrenmesi, örüntü tanımlama ve veri görselleştirmesi gibi pek çok teknik alan ile bağlantılı çok disiplinli bir alandır. Veri madenciliği üretim, kalite, bankacılık, biyoloji, finans, pazarlama, sigorta, tıp gibi birçok alanda uygulanmaktadır (Savaş, Topaloğlu ve Yılmaz, 2012).

Bu çalışmada, ilk olarak veri setleri için ön işleme yapılıp sonra kümeleme ve sınıflandırma yöntemleri uygulanarak sınıflandırma performans değerlendirme kriterlerinin iyileştirilmesi üzerine çalışılmıştır. Çalışma kapsamında, hem orta büyüklükte hem de daha büyük veri setlerinde önerilen yöntemler ile elde edilen sonuçlar karşılaştırılmıştır ve üç aşama halinde ilerlenilmiştir. İlk aşamada sınıflandırma algoritmaları veri setinin sadece ön işleme yapılmış haline uygulanmıştır. İkinci aşamada ele alınan veri setinin kaç kümeye ayrılması gerektiğine karar verilmiştir. Bu aşamada, literatürde kullanılan yöntemlerin kullanılması planlanmıştır (K-ortalamalar,

hiyerarşik kümeleme gibi). En yüksek kümeleme performansını sağlayan küme sayısı tespitinin ardından, her kümenin kendi içindeki sınıflandırma performansı gözlemlenmiştir. Sonraki aşamada ise K- ortalamalar ve hiyerarşik kümeleme yöntemlerinin birbirini besleyecek şekilde melez olarak kullanılması planlanmıştır. Bunun nedeni, K-ortalamalar kümeleme yönteminde önceden bilinemeyen küme sayısı ve başlangıç küme merkezleri dezavantajıdır. Melez (Hibrid) kümeleme yöntemiyle oluşan her kümeye sınıflandırma algoritmaları uygulanmıştır. Her aşama sonunda parametre optimizasyonu ile sınıflandırma performans kriterleri iyileştirilmeye çalışılmıştır. Bu işlemlerin sonucunda veri setini doğrudan sınıflandıran algoritmaların performansı ile diğer aşamaların sonucundaki sınıflandırma performansları karşılaştırılmıştır.

Tez kapsamında yapılan çalışmanın amacı literatürde sıklıkla kullanılan veri setlerine doğrudan uygulanan sınıflandırma algoritmalarının performansları ve önerilen yöntemler ile sınıflandırma algoritmalarının performanslarını karşılaştırmak, ayrıca en iyi performansı gösteren yöntemi tespit etmektir. Bu çalışmanın literatüre katkısı ise melez kümeleme yöntemi ile sınıflandırma algoritmalarının birlikte uygulanması olmuştur.

Çalışmanın geri kalan kısmı şu şekilde yapılandırılmıştır: İkinci bölümde kuramsal temeller ve kaynak araştırması, üçüncü bölümde çalışmada kullanılan materyal ve yöntemler, dördüncü bölümde bulgular, beşinci bölümde ise tartışma ve sonuç yer almaktadır.

2. KURAMSAL TEMELLER ve KAYNAK ARAŞTIRMASI

Veri tabanlarında bilginin elde edilme sürecinin bir aşaması olan Veri Madenciliği (Data Mining), anlamlı ve yeni bilgiyi büyük veri tabanlarından çıkarma işlemidir. Bu sebeple veri madenciliği, çeşitli algoritmalar kullanarak veri tabanından anlamlı bilgiler elde etmek için geniş bir araştırma alanı olarak kullanılmaktadır (Fayyad, Piatetsky-Shapiro ve Smyth, 1996).

Önceden veri taraması, veri yakalanması olarak da adlandırılan veri madenciliği kavramı 1960'larda bilgisayarların veri analizi amacıyla da kullanılmaya başlanmasıyla birlikte ortaya çıkmıştır ve veri madenciliği ile istenilen bilgilere ulaşılabileceğine inanılmıştır (Öğüt, 2005).

1980'li yıllarda veri tabanları, 1990'lı yıllarda ise verilerin depolanması için veri ambarları geliştirilmiştir (Altun, 2017). Yine 1990'lı yıllarda veri madenciliğine farklı yaklaşımlar getirilip çoklu ilişkilendirmeye başlanmıştır ve dinamik hale getirilmiştir. Bu yaklaşımların temeli pazarlama, otomasyon, istatistik, veri tabanlar ve makine öğrenimi gibi disiplinlere ve kavramlara dayanmaktadır (Ragothaman ve Sarojini, 2016). 2000'li yıllar ise veri madenciliğinin tüm alanlarda en yaygın kullanıldığı yıllar olmuştur (Altun, 2017).

Veri madenciliği, ürettikleri anlamlı bilgileri faaliyet alanına yönelik karar destek mekanizmaları için gerekli ön bilgi olarak kullanmaktadır (Fayyad ve diğerleri, 1996). Anlamlı bilgiyi elde etme süreci, veri girişi yapıldıktan sonra aşağıdaki üç ana adımdan oluşmaktadır. Bu adımlar sonunda bilgi elde edilmiş olur (Ragothaman ve Sarojini, 2016; García, Luengo ve Herrera, 2015).

- Veri ön işleme ve temizleme
- Veri madenciliği
- Veri sonrası işlem (Değerlendirme ve Yorumlama)

Ön işleme, veri madenciliğinde üç ana adımdan biridir (Velayutham ve Thangavel, 2011). Ön işleme görevleri veri temizleme, veri azaltma, veri dönüştürme ve veri entegrasyonu olarak dört kategoriye ayrılabilir (Öğüt, 2005).

Kurgan ve Musilek (2006), birkaç bilgi edinme çalışmasını gözden geçirdiğinde ön işleme için harcanan çaba ve zamanın yaklaşık %50-70'ini gerektirdiğini tespit etti. Veri ön işleme süreçleri güvenilir bir şekilde gerçekleştirildikten sonra veri madenciliği için daha doğru ve yararlı olarak tasarlanabilmiş bir veri seti beklenmektedir (Pyle, 1999).

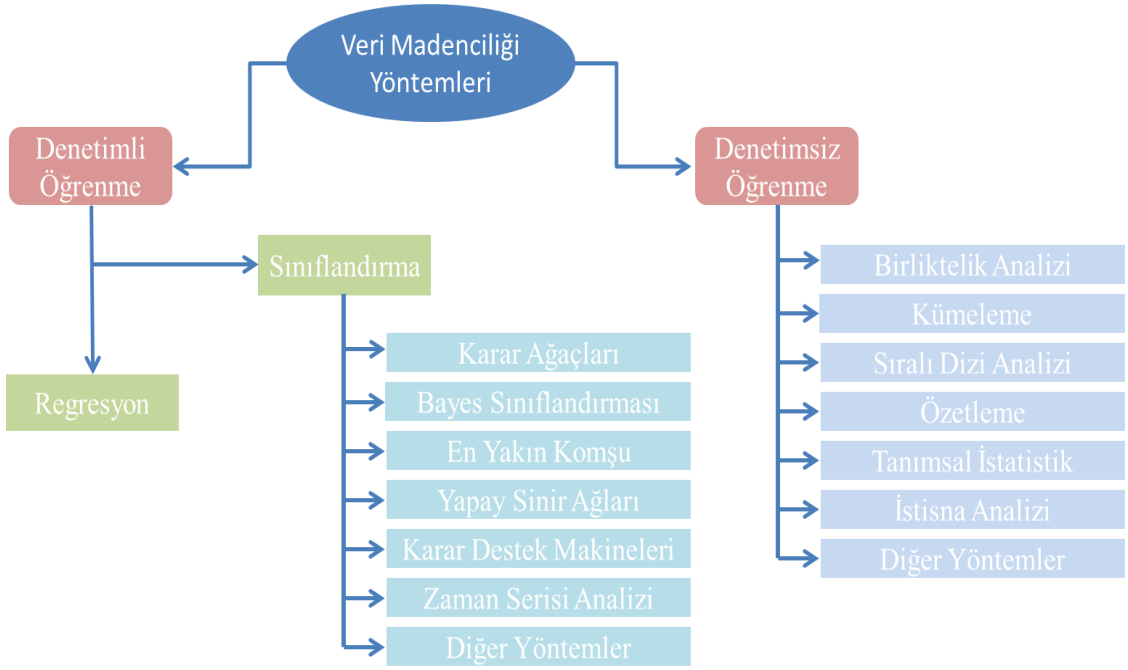
Veri setine uygun olarak ön işlemeler gerçekleştirildikten sonra veri madenciliği adımına geçilir. Veri madenciliği, denetimli ve denetimsiz öğrenme modelleri meydana getirip veriyi analiz eder. Ham verilerden anlamlı bilgileri çıkarmak için iki öğrenme yöntemi sunulmaktadır.

1. Denetimli Öğrenme: Bu öğrenme türünde veri seti girdi olarak verilir ve eğitmen eşliğinde istenildiği gibi çıktı alınır. Eğitmen genellikle giriş veri setini eğitir ve sınıflandırır. Yapay sinir ağları, karar ağaçları, karar destek makineleri vb. denetimli öğrenme yöntemlerine örnek olarak verilebilmektedir. Öngörü modelleme çalışması olarak da adlandırılabilen denetimli öğrenme sonucunda belirli bir konuda öngörü için diğer özelliklerin değerlerine dayanarak kullanılacak bir model oluşturmak amaçlanır. Denetimli öğrenme problemleri "regresyon" ve "sınıflandırma" olarak ikiye ayrılmaktadır. Regresyon yönteminde, sonuçlar sürekli bir değer tahmin edilmeye çalışılmaktadır, sınıflandırma yönteminde ise, sonuçlar ayrı kategorilerde tahmin edilmeye çalışılmaktadır (Zaimoğlu, 2018).

2. Denetimsiz Öğrenme: Veri tabanındaki analiz edilen verilerin sonucu, verinin mevcut durumunu ya da yaklaşık olarak neyi ifade ettiğini belirten yöntemleri kapsar. Henüz tespit edilmemiş ya da önceden keşfedilmemiş bilgiyi tespit etmek, bulmak için kullanılan bir yöntemdir. Denetimsiz modelde öğrenme işlemi sırasında istenilen sonuç sağlanmaz. Bu yöntem, sınıflardaki girdi

verilerini yalnızca istatistiksel özelliklerine dayalı olarak kümelemek için kullanılabilir (Zaimoğlu, 2018).

Şekil 2.1'de denetimli ve denetimsiz öğrenme yöntemleri verilmiştir.



Şekil 2.1. Veri madenciliği deneyimli ve deneyimsiz öğrenme yöntemleri (Zaimoğlu, 2018).

Sınıflandırma, veri madenciliğinde sıklıkla kullanılan bir tekniktir ve verinin içerdiği ortak özelliklere göre ayrıştırılması işlemidir. Sharma, Agrawal, Agarwal ve Sharma (2013)'ya göre ise sınıflandırma, daha gelişmiş ve karmaşık bir bilgi keşfetmek için verileri önceden tanımlanmış sınıflara veya gruplara ayırmak ve/veya tahmin etmek için veri madenciliğini kullanan bir makine öğrenimi metodolojisidir. Bu veri madenciliği tekniği birçok alanda uygulanmış, başarılı sonuçlar elde etmiş, bu sebeple her yerde araştırmacıların dikkatini çekmiştir.

Karar ağacı (Decision Tree) sınıflandırmada iyi performans göstermesi ve güçlü öğrenme yeteneği sayesinde klasik ve popüler makine öğrenme modellerinden biridir. Uyarlamalı artırma (Adaboost), Torbalama (Bagging) ve Rassal Orman (Random forest) gibi algoritmalar da karar ağaçlarına dayanmaktadır (Quinlan, 1987; Bahrudin, Alam ve

Haiyunnisa, 2016). Ağaç tabanlı topluluk öğrenme algoritmaları, tarım, ulaşım, enerji ve sağlık gibi çeşitli endüstri ve alanlarda uygulanabilmektedir (Hitziger ve Lies, 2014; Zhang ve Haghani, 2015; Papadopoulos, Azar, Woon ve Kontokosta, 2018).

Rassal Orman (RF), regresyon ve sınıflandırma için kullanılan popüler ağaç tabanlı topluluk yöntemlerinden biridir (Wei, Lu ve Song, 2015). Breiman tarafından önerilen algoritma, yeniden örnekleme yoluyla rastgele örnekler elde eder, rastgele örneklere dayalı olarak karar ağacını oluşturur, birden fazla karar ağaçlarının tahminlerini birleştirir ve oylama yoluyla son tahmin sonucunu yüksek bir tahmin doğruluğu ile elde eder. Ağaç yapısı serbest bir şekilde büyür ve budanmaya ihtiyaç duymaz, bu nedenle "aşırı uyum (over fitting)" dezavantajından etkili bir şekilde kaçınabilir. RF modelinin oluşturulması iki anahtar parametre içerir: karar ağaçlarının sayısı (ntree) ve bölünmüş düğümlerin sayısı (mtry) (J. Liu, Sun, Tan ve Y. Liu, 2020; Andrade, Faria, Silva, Chakraborty ve Curi, 2020).

RF'ler, hem kategorik hem de sayısal değişkenler için geçerlidir. RF'lerin özelliklerinden biri, bağımsız değişkenlerin sonuç değişkeniyle en etkili olandan en az ilişkili olana doğru sıralanmasını sağlayan yerleşik çapraz doğrulamadır. Bu, çok kaynaklı veri analizinde özellik çıkarımı için değer katar (Breiman, 2001).

Torbalama (bagging) algoritması ise Breiman (1996a) tarafından önerilmiştir. Bagging, öğrenme örneğinden yerine konulacak şekilde tekrar tekrar örnekler çekilerek yeni oluşturulan her torba örneğinde bir sınıflandırıcı oluşturmaya ve bunları birleştirilmiş bir tahmin edici elde etmek için birleştirmeye dayanır. Bagging, öğrenme setindeki küçük bir değişikliğin tahminlerde kayda değer bir değişikliğe neden olabileceği problemler için çok etkilidir (Breiman, 1996a, 1996b).

Bir diğer karar ağacı modeli olan Adaboost (uyarlamalı artırma) algoritması, Yoav Freund ve Robert Shapire (1995) tarafından bir dizi zayıf sınıflandırıcıdan güçlü bir sınıflandırıcı üretebilmek için önerilmiştir. Adaboost algoritması, eğitim verileri üzerinde bir ağırlık katsayısını koruyarak bir dizi zayıf öğrenen oluşturur ve bunları her zayıf öğrenme döngüsünden sonra değiştirilebilecek şekilde ayarlar. Mevcut zayıf

öğrenen tarafından yanlış sınıflandırılan eğitim örneklerinin ağırlıkları artırılırken doğru sınıflandırılan örneklerin ağırlıkları azaltılacaktır (Li, Wang ve Sung, 2005).

Sınıflandırma ve regresyon ağacı (CART) modeli, makine öğrenmesinde yaygın olarak kullanılan sınıflandırma yöntemlerindendir (Breiman, Friedman, Olshen ve Stone, 1984). CART modeli, sınıflandırma sonuçlarını basit ve sezgisel bir ağaç yapısı olarak sunarak sınıflandırma sürecinin anlaşılmasını ve yorumlanmasını kolay hale getirir. CART modeli genellikle ağaç oluşturma ve budamayı içeren özellik seçiminden oluşur ve sınıflandırma veya regresyon için kullanılabilir. Bir veri kümesinin bağımlı değişkeni sürekli bir değerse ağaç algoritması bir regresyon ağacıdır, ancak kesikli bir değerse ağaç algoritması bir sınıflandırma ağacıdır (Ronowicz, Thommes, Kleinebudde ve Krysi´nski, 2015).

CART modelini kullanmanın çeşitli avantajları vardır. İlk olarak, CART hem ayrık hem de sürekli veri kümelerinden gelen veriler için uygulanabilmektedir. İkinci olarak, CART, basitçe optimal sınıflandırma öznitelik modelini seçen ve düğümün optimal özniteliği olarak en küçük Gini dizinini kullanır. Üçüncüsü, CART'ın bağımsız değişkenler arasındaki doğrusallık sorununu dikkate almasına gerek yoktur ve bu da aykırı değerlerle başa çıkmasını kolaylaştırır (Ronowicz ve diğerleri, 2015).

Destek vektör makinesi ise (SVM), Yapısal Risk Minimizasyonu teorisinden geliştirilmiştir. Bununla birlikte, pratikte uygulanan sınıflandırma performansı genellikle teorik olarak beklenenden uzaktır. Gerçek SVM'nin sınıflandırma performansını iyileştirmek için, bazı araştırmacılar geleneksel Bagging ve Adaboost gibi topluluk yöntemlerini kullanmaya çalışırlar (Kim ve Pang, 2005). Ancak Adaboost algoritmasının SVM'lerin performansını her zaman iyileştirmesi beklenmemektedir, performansını kötüleştirdiği de olabilmektedir (Buciu, 2006; Jeevani, 2001). Bu da SVM'nin özünde kararlı ve güçlü bir sınıflandırıcı olduğunun göstergesidir.

K-en yakın komşu (KNN) sınıflandırıcısı basitliği, etkinliği ve sezgiselliği nedeniyle birçok alanda yaygın olarak çalışılan ve kullanılan klasik bir sınıflandırma algoritmasıdır (Cover ve Hart, 1967; Wu, Kumar, Quinlan, Ghosh, Yang ve Motoda,

2008; Yiwei, Zhibin, Yikun ve Wei, 2020). KNN sınıflandırıcısının temel fikri, sorgunun sınıf etiketini eğitim kümesindeki en yakın komşularına göre belirlemektir. Spesifik olarak, sorguyu, çoğunluk oylama kuralına göre k-en yakın komşuları arasında en sık görünen sınıfa atayabilmektedir (Li, Chen ve Chen, 2008).

Genel olarak sınıflandırıcının performansını ölçen araçlar değerlendirme metriği olarak tanımlanabilir. Farklı metrikler, sınıflandırma algoritması tarafından oluşturulan sınıflandırıcının farklı özelliklerini değerlendirir (Hossin ve Sulaiman, 2015).

Doğruluk (accuracy) veya hata oranı (error rate), birçok araştırmacı tarafından sınıflandırıcıların performansını değerlendirmek için kullanılan en yaygın metriklerden biridir. Eğitilmiş sınıflandırıcı, görünmeyen verilerle test edildiğinde eğitilmiş sınıflandırıcı tarafından doğru bir şekilde tahmin edilen örneklerin toplamını ifade eden toplam doğruluğa dayalı olarak ölçülür (Lavesson ve Davidsson, 2008).

Doğruluk veya hata oranının avantajları, bu metriğin daha az karmaşıklıkla hesaplanmasının kolay olmasıdır. Çok sınıflı ve çok etiketli problemler için de geçerlidir ve insanlar tarafından anlaşılması kolaydır (Huang ve Ling, 2007). Doğruluk oranı ile hata oranı birbirlerinin tersidir. Diğer performans metriklerinden olan kesinlik ve anma ölçütlerinin daha doğru sonuçlar verilebilmesi ikisinin bir arada olduğu F-ölçütü kullanılır. F-ölçütü ve anma değerlerinin harmonik ortalaması olarak hesaplanır (Haltaş ve Alkan, 2016).

Son derece yüksek boyutlu veriler, mevcut öğrenme yöntemlerine, yani boyutluluk sebebiyle ciddi zorluklar sunmuştur. Çok sayıda özniteliğin varlığı ile bir öğrenme modeli, performanslarının dejenere olmasına neden olarak aşırı uyum (over fitting) sağlama eğilimindedir. Öznitelik seçimi, uygulayıcılar arasında boyutluluğu azaltmak için yaygın olarak kullanılan bir tekniktir. Genellikle daha iyi öğrenme performansına (örneğin, sınıflandırma için daha yüksek öğrenme doğruluğu), daha düşük hesaplama maliyetine ve daha iyi model yorumlanabilirliğine yol açan belirli uygunluk değerlendirme kriterlerine göre orijinal özelliklerden ilgili özelliklerin küçük bir alt

kümesini seçmeyi amaçlar (Liu ve Motoda, 2007; Hastie, Tibshirani ve Friedman, 2001).

İdeal olarak, öznitelik seçme yöntemleri özniteliklerin alt kümelerini araştırır ve bazı değerlendirme işlevlerine göre yarışan 2^m aday alt kümeler arasından en iyisini bulmaya çalışır (Dash ve Liu, 1997).

Eğitim aşamasından önce, veriler üzerinde makul bir süre içinde en iyi performansı arşivleyen bir dizi hiperparametre değeri bulma işlemine ise hiperparametre optimizasyonu veya ayarlama denir. Hiperparametre optimizasyonu makine öğrenimi algoritmalarının tahmin doğruluğunda önemli bir rol oynar. Temel olarak iki tür hiperparametre optimizasyonu yöntemi vardır, bunlar manuel arama ve otomatik arama yöntemleridir. Manuel arama, hiperparametre setlerini elle dener. Sonuçlar üzerinde daha büyük etkisi olan önemli parametreleri belirleyebilen ve ardından görselleştirme araçlarıyla belirli parametreler ile nihai sonuçlar arasındaki ilişkiyi belirleyebilen uzman kullanıcıların temel sezgilerine ve deneyimine bağlıdır (Aarshay, 2018).

Manuel aramanın dezavantajlarının üstesinden gelmek için, Izgara arama veya Kartezyen hiperparametre arama gibi otomatik arama algoritmaları önerilmiştir. Izgara aramanın (Grid search) ilkesi kapsamlı aramadır. Izgara arama, eğitim setindeki olası hiperparametre değerlerinin her bir kombinasyonu ile bir makine öğrenimi modelini eğitir ve performansı bir çapraz doğrulama setinde önceden tanımlanmış bir ölçüye göre değerlendirir. Son olarak, Izgara araması, en iyi performansı elde eden hiperparametreleri verir. Bu yöntem otomatik ayarlamayı başarmasına ve teorik olarak optimizasyon amaç fonksiyonunun global optimal değerini elde etmesine rağmen, boyutsallık açısından dezavantajlıdır, yani ayarlanan hiperparametrelerin sayısı ve değer aralığına bağlı olarak algoritmanın verimliliği hızla azalır (Bergstra ve Bengio, 2012).

Rassal arama (Random Search, RS) yöntemi, sınırlı yürütme süresi ve kaynakları göz önüne alındığında, arama alanında hiper parametre kombinasyonlarını rastgele seçen başka bir teorik yöntemdir (James ve Yoshua, 2012). Izgara arama yönteminde pahalı maliyet problemini çözmek için, çoğu veri seti için hiperparametrelerden sadece

birkaçının gerçekten önemli olduğunu bulan rassal arama algoritması önerilmiştir (Bergstra ve Bengio, 2012). Aramanın önemi olmayan hiperparametrelere indirgenmesiyle genel verimlilik arttırılabilir ve son olarak optimizasyon fonksiyonunun yaklaşık çözümü elde edilir. Rassal arama yöntemi, bir dizi değerin rastgele kombinasyonlarını dener. Izgara arama yöntemi ile karşılaştırıldığında, rassal arama yöntemi yüksek boyutlu uzayda daha verimlidir (Bergstra, Bardenet, Bengio ve Kégl, 2011).

Veri madenciliği aşamasında literatürde sıklıkla uygulan tekniklerden biri kümeleme yöntemidir. Kümeler genellikle, etiketlenmemiş bilgilerin veri çeşitliliğindeki problemlerle ilgilenen, en başta gelen denetimsiz öğrenme problemi olarak düşünülür (Soni ve Ganatra, 2012).

Hedef, küme içi benzerliği en üst düzeye çıkarmaya ve kümeler arası benzerliği en aza indirmeye çalışmak ve bu da yüksek küme tutarlılığını korumak için denetimsiz bir şekilde (yani herhangi bir ön bilgi olmadan) ayrıştırmaktır. Kümeleme yöntemi, veri örneklerini, farklı örnekler farklı gruplara aitken benzer örneklerin birlikte gruplanacağı şekilde alt kümelere ayırır (“Wikipedia”, 2013; Elavaras, 2011).

En popüler kümeleme yöntemlerden biri olan K-ortalamalar, MacQueen tarafından 1967 yılında önerilmiştir. Veri kümesini, k'nin önceden belirlendiği k ayrık alt kümeye bölmektedir. Algoritma, kümelere hiçbir yeni nesne ataması yapılamayana kadar nesnelerin en yakın geçerli küme ortalamasına atanmasını ayarlamaya devam etmektedir (Voorhees, 1986).

Literatür incelendiğinde K-ortalamalar ile ilgili pek çok çalışma yapıldığı görülmektedir. 1967'den 1998'e kadar olan zaman diliminde, tüm araştırma çalışmaları, kümeleme alanında K-ortalamalar yönteminin tanıtılmasıyla ilgiliydi. Bundan sonra, K-ortalamalar kümeleme yönteminde tüm değişiklikler ve iyileştirmeler başlatıldı (Shukla ve Naganna, 2014).

Küme sayısını önceden belirlemek, K- ortalamalar kümeleme yaklaşımı için her zaman zor bir problem olduğu için başlangıçta doğru küme sayısını belirlemek oldukça iyi sonuçlar verecektir. Yapılan bazı çalışmalarda küme sayısının veri setinde bulunan sınıf sayısına göre belirlendiği tespit edilmiştir. (Abbas, 2008). Tez kapsamında ele alınan veri setleri için, önerilen kümeleme yöntemindeki küme sayısı ile sınıf etiketi sayısı karşılaştırılmıştır.

Pham, Dimov ve Nguyen (2005), K- ortalamalar kümeleme yönteminde kullanılan k sayısı üzerinde çalışmıştır. Farklı veri kümeleri için farklı sayıda küme oluşturdular. Cheung (2003), orijinal olarak Voorhees (1986) tarafından verilen K-ortalamalar kümeleme yönteminin genelleştirilmiş bir yolunu vermiştir. Bu algoritma ile başlangıçta bilinen sayıda küme olmadan doğru kümeleme oluşturulabilir.

Yaygın olarak kullanılan diğer kümeleme yöntemi ise hiyerarşik kümelemedir. Hiyerarşik kümeleme algoritmaları, bir veri kümesini bir dizi iç içe bölüm halinde böler veya birleştirir. İç içe bölümlerin hiyerarşisi, toplayıcı (aşağıdan yukarıya) veya bölücü (yukarıdan aşağıya) olabilir. Toplayıcı (Aglomeratif) yöntemde kümeleme, her nesnenin kendi başına küme olması ile başlayıp tüm nesneler tek bir kümede bir araya gelene kadar en yakın küme çiftlerini kümelemeye devam etmektedir. Bölücü hiyerarşik kümeleme yöntemi ise, toplayıcı yöntemin tam tersine tek bir kümedeki tüm nesnelerle başlayıp tüm nesneler birim kümelere ayrılana kadar daha büyük kümeleri daha küçük kümelere bölmeye devam eder (Rani ve Rohil, 2013; Voorhees, 1986). Her iki hiyerarşik yöntem de dendrogram ile kümeleri temsil etmektedir. Hiyerarşik kümeleme yöntemi, hiyerarşinin istenilen düzeyde kesilmesine olanak sağlaması yönünden avantajlıdır.

Hiyerarşik kümeleme yöntemleri içerisinde genellikle Ward yöntemi en iyi sonuç veren yöntem olarak kabul görmektedir (Hands ve Everitt, 1987; Ferreira ve Hitchcock, 2009). Ward yöntemi, aglomeratif kümeleme yöntemleri arasında, klasik kareler toplamı kriterine dayalı olarak her ikili füzyonda grup içi dağılımı(küme içindeki varyans) minimize ederek kümelerin oluşmasını sağlayan tek yöntemdir. Uzaklık

ölçüleri arasında en sık kullanılan yöntem ise Öklid uzaklığıdır (Murtagh ve Legendre, 2014).

Chen, Tai, Harrison ve Pan (2005)'in yaptığı çalışmada mevcut hiyerarşik ve hiyerarşik olmayan kümeleme yöntemlerinin dezavantajlarını azaltmak veya ortadan kaldırmak amacıyla birden fazla kümeleme yönteminin birlikte kullanılması melez kümeleme olarak adlandırılmıştır. Bu yaklaşımda hiyerarşik kümeleme ile tespit edilen küme sayısı ve küme merkezleri, K- ortalamalar yöntemi için başlangıç bilgilerini oluşturmaktadır.

Şchiopu (2010)'nun yaptığı çalışmada iki adımlı kümeleme, esasen büyük veri kümelerini analiz etmek için tasarlanmış bir algoritmadır. Algoritma, yaklaşım kriterini kullanarak kümelerdeki gözlemleri gruplandırır. Ön kümeleme, aykırı veri analizi ve kümeleme adımlarından oluşmaktadır. Ön kümeleme adımında, veri kaydını tek tek tarar ve mevcut kaydın daha önce oluşturulmuş kümelerden birine eklenip eklenemeyeceğine karar verilir veya mesafe (Öklid ve log-olasılık mesafesi) kriterine göre yeni bir küme başlatılır. Kümeleme aşaması, girdi olarak ön kümeleme adımında oluşturulan alt kümelere sahiptir ve bunları istenen sayıda kümede gruplandırabilir. Prosedür, aglomeratif hiyerarşik kümeleme yöntemi kullanır. Klasik kümeleme analizi yöntemleriyle karşılaştırıldığında, hem sürekli hem de kategorik özelliklere olanak sağlar. Ayrıca, yöntem en uygun sayıda kümeyi otomatik olarak belirleyebilir.

İki adımlı kümeleme yönteminin aynı zamanda hiyerarşik olmayan kümeleme yöntemlerinden K-ortalamalar ve hiyerarşik yöntemlerden Ward yönteminin birleştirilmesi ile oluşan bir melez kümeleme tekniği olduğu belirtilebilir. İki adımlı algoritma kendi içerisinde daha benzer kümeler sağladığından farklı çalışmalarda birçok araştırmacı tarafından uygulanmıştır (Ceylan, Gürsev ve Bulkan, 2017).

Aydın ve Seven (2015), nüfus müdürlüklerini iş yoğunlukları bazında melez kümeleme yöntemi ile kümelendirmiştir. Küme sayısına karar verirken Silhouette endeksinden yararlanılmıştır. Yapılan çalışma sonucunda, altı farklı küme halinde benzer iş yoğunluğuna sahip illerden oluşan yapı ortaya çıkarılmıştır.

Onan’ın (2018) yaptığı çalışmada, firma başarısızlıklarının tahmin edilebilmesi için sınıflandırma algoritmaları kullanılmıştır. Geliştirilen yöntemde, K-ortalamalar algoritması kullanılarak farklı eğitim alt kümeleri oluşturulmaktadır. Bu eğitim alt kümelerine dayalı olarak her bir temel öğrenme algoritması eğitilmekte ve temel öğrenme yöntemlerinin çıktıları çoğunluk oylaması aracılığıyla birleştirilmektedir.

Kim, Jo ve Shin (2016) tarafından gerçekleştirilen çalışmada, firma başarısızlıklarının tahmin edilmesi için kümelemeye ve yapay sinir ağlarına dayalı bir model önerisi önerilmiştir. Yapılan çalışmada, veri dengesizliğini ortadan kaldırmak amacıyla kümeleme yöntemi kullanılmıştır. Kümeleme analizi uygulanarak uygun veri nesnelerinin seçilmesi, veri dengesizliğinin kaldırılması ile doğru sınıflandırma performansının artırılması amaçlanmıştır.

Çalışkan ve Soğukpınar (2008) tarafından gerçekleştirilen çalışmada veri madenciliği yöntemlerinden K-ortalamalar ve K en yakın komşu yöntemlerinin iyileştirilmesi amacıyla nüfus tespiti için K-ortalamalar ve K en yakın komşu yöntemlerini bir arada kullanan melez bir yapı geliştirilmiştir. Geliştirilen uygulamada en hızlı sonucu veren K-ortalamalar uygulaması ile test kümesi daha küçük alt kümelere ayrılarak K en yakın komşu yönteminin zaman karmaşası ve bellek gereksinimi azaltılmıştır.

Literatür incelediğinde K-ortalamalar ve hiyerarşik kümeleme yöntemlerinin dezavantajlarını yok edebilmek için bu iki yöntemi bir arada uygulayan melez kümeleme çalışmaları görülmüştür. Ancak melez kümeleme ve sınıflandırma yöntemlerini bir arada kullanan herhangi bir çalışmaya rastlanmamıştır. Bu sebeple yapılan çalışma kapsamında melez kümeleme ve sınıflandırma yöntemlerini bir arada kullanılması önerilmiştir. Önerilen yöntem ile doğruluk oranlarının iyileştirilmesi hedeflenmiştir.

Nkonyana, Sun, Twala ve Dogo (2019), Steel Plates Fault veri setini kullanarak eğitim / test veri setinde Rassal Orman algoritması uygulamıştır. K-katlı çapraz doğrulama uyguladıkları veri setine ise Izgara arama parametre optimizasyonu yapılmıştır.

Tez kapsamında önerilen algoritmada UCI veritabanında kullanılan diğer veri setlerinin yanı sıra Steel Plates Fault veri seti de kullanılıp Nkonyana ve diğerlerinin (2019) yaptığı çalışma ile karşılaştırılmıştır.

3. MATERYAL ve YÖNTEM

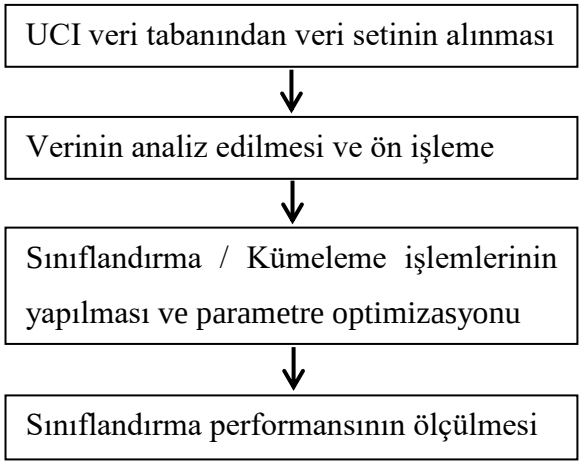
Bu bölümde ilk olarak literatürde sıklıkla atıf alan ve çalışmada da kullanılan veri setleri ile ilgili bilgiler aktarılmıştır. UCI veri tabanında incelenen veri setleri arasından çalışmaya dahil edilenler ve bu veri setlerinin veri sayısı, öznitelik sayısı, sınıf etiketi sayısı özellikleri Çizelge 3.1'de verilmiştir.

Çizelge 3.1. Çalışmada kullanılan veri setleri ve özellikleri

Veri Seti	Veri Sayısı	Öznitelik Sayısı	Sınıf Etiketi Sayısı
Çelik Levha Arızası	1941	28	7
Cam	214	10	7
Omurga	310	7	2
Tohumlar	210	8	3
Kullanıcı Bilgisi	403	6	4

Çalışma kapsamında, hem orta büyüklükte hem de daha büyük veri setlerinde önerilen yöntemler Pyhton programında uygulanmıştır. Python, 1990 yılında Guido van Rossum tarafından tasarlanan güçlü, genel amaçlı, etkileşimli ve nesne tabanlı bir programlama dilidir. Liste ve diziler, dinamik yazma ve dinamik bağlama, modüller, sınıflar, otomatik bellek yönetimi vb. gibi üst düzey veri yapıları sağlar (Miranda, 1999). Pyhton programında önerilen yöntemlerin cevapları karşılaştırılmış ve üç farklı aşama halinde ilerlenip göstergeler veri setinin sadece ön işleme yapılmış halindeki göstergeleri ile karşılaştırılmıştır.

İlk olarak veri setleri için sadece ön işleme yapılıp sonra kümeleme ve sınıflandırma yöntemleri de uygulanarak sınıflandırma performans değerlendirme kriterlerinin iyileştirilmesi üzerine çalışılmıştır. Gerçekleştirilmiş olan çalışmanın genel akışı Şekil 3.1'de verilmiştir.



Şekil 3.1. Çalışmada kullanılacak yöntemlerin genel akışı

3.1. Çelik Levha Arızası Veri Seti

Önerilen algoritmanın literatürdeki yerini doğru tespit edebilmek için sıklıkla kullanılan yöntem, literatürde önerilen başka bir algoritma ile aynı veri setini kullanmaktır. Bu amaçla, Nkonyana ve diğerlerinin (2019) yaptığı çalışmada odaklanılan Çelik Levha Arızası (Steel Plates Fault) veri seti öncelikli olarak incelenmiştir. Bu veri setinde çelik levhalarda bulunan kalite hatalarının türü tahmin edilmeye çalışılmıştır. Steel Plates Fault veri setinde 7 sınıf etiketi, 27 öznitelik bulunmaktadır. İlgili veri setindeki sınıf etiketlerinin isimleri ve her bir sınıf etiketine ait kaç veri bulunduğu Çizelge 3.2'de verilmiştir. Öznitelikler ise tamsayı biçiminde bulunmaktadır.

Çizelge 3.2. Sınıf etiketleri dağılımı

Sınıf Etiketi	Sınıf Etiketi Adı	Veri Sayısı
1	Pastry	158
2	Z_Scratch	190
3	K_Scatch	391
4	Stains	72
5	Dirtiness	55
6	Bumps	402
7	Other_Faults	673

3.1.1. Ön işleme

Veri analizi çalışması gerçekleştirilip veri setinde bu çalışmaya uygun ön işleme yöntemleri belirlenmiştir ve bu sayede maksimum seviyede iyileştirme yapılması sağlanmıştır.

Veri seti incelendiğinde aşağıdaki ön işleme işlemlerinin yapılmasına karar verilmiştir.

- Eksik veri analizi
- Aykırı veri analizi
- Normallik testi
- Korelasyon matrisi
- Öznitelik seçimi (Feature selection)

Eksik veri analizinde eksik verileri tespit edip bu eksik verilerin neden ortaya çıktığını anlamak gerekmektedir.

Aykırı değer, bir veri setindeki gözlemlerin geri kalanından büyük ölçüde farklı olan herhangi bir veri noktasıdır. Aykırı veri analizinde yaygın olarak kullanılan yöntemlerden biri Z-skoru yöntemidir. Z-skor, normal dağılıma sahip bir veri setinde, bir gözlemin ortalamadan ne kadar standart sapma saptığını göstermektedir. Herhangi bir gözlem için Z-skoru, gözlemden ortalama çıkartıldıktan sonra standart sapmaya bölünmesiyle hesaplanmaktadır. Z-skorun hesaplanmasına ait formülasyon Denklem 3.1'de verilmiştir.

$$Z = \frac{x - \mu}{\sigma} \tag{3.1}$$

Z-skor ile aykırı veri analizinin yapılabilmesi için ilk olarak veri setinin normal dağılıma uyup uymadığı kontrol edilmiştir.

Veri setinin normal dağılıma uyup uymadığı, normallik sınaması için yaygın olarak kullanılan Kolmogorov Smirnov testi ile analiz edilmiştir. Kolmogorov Smirnov testi için oluşturulan hipotezler aşağıdaki gibidir.

H_0: Değişken normal dağılıma sahiptir, p-değer > 0,05
H_1: Değişken normal dağılıma sahip değildir, p-değer < 0,05

Veri seti için bir diğer ön işleme işlemi ise korelasyon matrisi oluşturulması olarak belirlenmiştir. Korelasyon matrisi özniteliklerin birbirleri ile olan ilişkilerini ortaya koymaktadır. Korelasyon matrisinde her bir 2'li öznitelik ilişkisi için -1 ile 1 arasında değer verilmektedir. Bu değer 1'e yakınsa korelasyonuna bakılan 2 öznitelik arasında güçlü bir doğru orantı, -1'e yakınsa güçlü bir ters orantı, 0 ise lineer bir ilişki olmadığı söylenebilmektedir (Dursun, 2018).

Ön işleme olarak uygulanmasına karar verilen son işlem ise öznitelik seçimidir. Öznitelik seçimi, boyutluluğu azaltmak için yaygın olarak kullanılan bir tekniktir. Veri seti için uygulanacak her bir algoritma ve her öznitelik sayısı için öznitelik seçimi yapılması planlanmıştır. Veri setleri için seçilen öznitelikler içerisinde yüksek korelasyona sahip öznitelikler bulunuyor ise, birbirlerini kötü etkileme ihtimalinin önüne geçebilmek için öznitelik seçiminde elde edilecek önem sırasına göre daha önemsiz olan öznitelik, seçilen öznitelikler arasından çıkartılarak algoritma sonucu tekrar elde edilmiştir.

3.1.2. Sınıflandırma algoritmaları

Ön işleme yöntemleri belirlendikten ve uygulandıktan sonra önerilen yöntemler ile veri madenciliği aşamasına geçilmiştir. Pyhton programında yapılan çalışmada uygulanan sınıflandırma algoritmaları aşağıda verilmiştir.

i. Rassal Orman,
ii. Sınıflandırma ve regresyon ağacı (CART),
iii. Torbalama (Bagging),

iv. Uyarlamalı artırma (Adaboost),
v. Destek vektör makinesi (SVM),
vi. K- en yakın komşu (KNN)

Rassal Orman algoritmasında amaç tek bir karar ağacı üretmek yerine her biri farklı eğitim kümelerinde eğitilmiş olan çok sayıda, çok değişkenli ağacın kararlarını birleştirmektir. Algoritmanın aşamaları aşağıdaki gibidir.

1. Algoritma, sağlanan veri setinden rastgele örnekler seçecektir.
2. Algoritma, seçilen her örnek için bir karar ağacı oluşturacaktır. Oluşturulan her karar ağacından bir tahmin sonucu elde edilecektir.
3. Daha sonra tahmin edilen her sonuç için sınıflandırma problemi için mod kullanacaktır.
4. Son tahmin olarak en çok oylanan tahmin sonucunu seçecektir.

CART algoritması ikili ağaç yapısına sahip olup ana düğümden iki yavru düğüm oluşturmaktadır. Algoritmada amaç homojen bir ağaç yapısı elde etmektir. Algoritmada 3 temel adım vardır.

1. Maksimum ağaç oluşturmak
2. Ağacın derinliğini belirlemek
3. Test verilerini ağaca uygulamak

Maksimum ağacı oluşturmadaki asıl amaç en saf, en iyi bölme durumunu elde etmektir (Bozan, 2010).

Torbalama (Bagging) algoritması gürültülü bir veri kümesindeki varyansı azaltmak için yaygın olarak kullanılan topluluk öğrenme yöntemidir. Torbalamada, bir eğitim setindeki rastgele bir veri örneği seçilir. Veri noktaları birden fazla kez seçilebileceği için veri kümesi her zaman tamamıyla işleme alınmaktadır. Bagging algoritmasının adımları aşağıda verilmiştir.

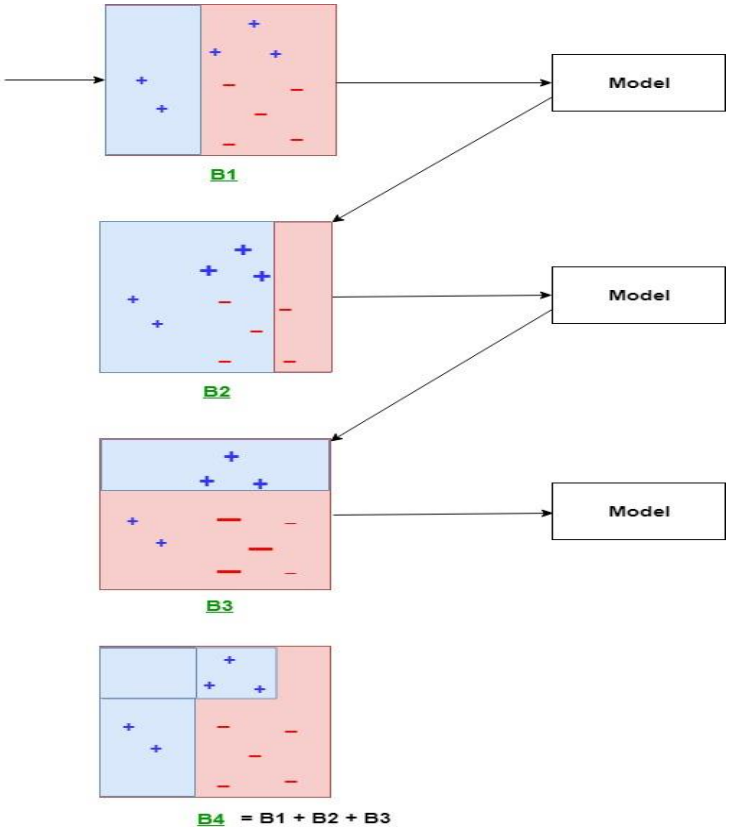
1. Veri kümesinden çoklu alt kümeler oluşturulur.
2. Bu alt kümelerin her birinde bir temel model (zayıf model) oluşturulur.
3. Modeller paralel ve birbirinden bağımsız olarak çalıştırılır.
4. Nihai tahminler, tüm modellerden gelen tahminler birleştirilerek belirlenir.

Diğer bir sınıflandırma algoritması olan Adaboost algoritmasının adımları aşağıda verilmiştir.

1. n adet veri noktaların her birine eşit ağırlık (1/n) atanır.
2. 1.adımdaki ağırlıklar modele girdi olarak sağlanır ve yanlış sınıflandırılmış veri noktaları belirlenir.
3. Yanlış sınıflandırılmış veri noktalarının ağırlığı arttırılır.
4. Gerekli sonuçlar elde edildiyse algoritma tamamlanır, elde edilmediyse 2. Adıma gidilir. Son model, tüm modellerin ağırlıklı ortalamasıdır.

Şekil 3.2'deki şema ile örnek verilecek olursa;

- B1, 5'i artı ve 5'i eksi olmak üzere iki türden toplam 10 veri noktasından oluşur ve her birine başlangıçta eşit ağırlık (1/10) atanmıştır. İlk model, veri noktalarını sınıflandırmaya çalışır ve dikey bir ayırıcı çizgi oluşturur, ancak 3 artıyı yanlış olarak eksi olarak sınıflandırır.
- B2, mevcut modelin bu artıları daha doğru sınıflandırmaya çalışması için yanlış sınıflandırılan 3 artının daha fazla ağırlıklandırıldığı önceki modelden 10 veri noktasından oluşur. Bu model, daha önce yanlış sınıflandırılan artıları doğru olarak sınıflandıran bir dikey ayırıcı çizgi oluşturur, ancak bu girişimde, iki eksiyi yanlış sınıflandırır.
- B3, yanlış sınıflandırılan 3 eksinin daha fazla ağırlıklandırıldığı önceki modelden 10 veri noktasından oluşur, böylece mevcut model bu eksileri doğru sınıflandırmaya çalışır. Bu model, daha önce yanlış sınıflandırılan eksileri doğru bir şekilde sınıflandıran yatay bir ayırıcı çizgi oluşturur.
- B4, kullanılan herhangi bir bireysel modelden çok daha iyi olan güçlü bir tahmin modeli oluşturmak için B1, B2 ve B3'ü bir araya getirir (Bhadauria, 2021).



Şekil 3.2. Adaboost algoritma adımları gösterimi (Bhadauria, 2021)

Torbalamada zayıf öğrenenler paralel olarak eğitilir, ancak artırmada (Adaboost) sırayla öğrenirler. Bu, bir dizi modelin oluşturulduğu ve her yeni model yinelemesinde, önceki modelde yanlış sınıflandırılan verilerin ağırlıklarının arttığı anlamına gelmektedir.

Destek vektör makinesi (SVM) bir uzayda optimal bir hiper düzlemi (bir veri kümesini doğrusal olarak ayıran ve sınıflandıran bir çizgi) tahmin etmek için eğitim veri setini kullanan ikili bir sınıflandırma tekniğidir. Hiper düzlemin boyutu, özelliklerin sayısına bağlıdır. Girdi özelliklerinin sayısı iki ise, hiper düzlem sadece bir çizgidir. En iyi hiper düzlem olarak makul olan seçim, iki sınıf arasındaki en büyük ayrımı veya marjı temsil eden seçimdir. Böylece, her iki taraftaki en yakın veri noktasına olan uzaklığı maksimize edilen hiper düzlem seçilmektedir. En iyi yerden geçebilecek hiper düzlemin yaratılmasına destek olmaları sebebiyle karar sınırına en yakın noktalara veya değerlere destek (*support*) denir. Eğer bu noktaların konumu değişirse en iyi durum değişeceği için karar sınırları da değişecektir. Boşluğu en iyilemek ve sınıflar arasına bir hiper

düzlem yerleştirmek, yeni gelen değerlerin doğru sınıf kategorisine girme şansını büyük ölçüde artırabilir. SVM'nin amacı, görünmeyen yeni nesneleri belirli bir kategoriye atayan bir model oluşturmaktır. Bunu, özellik alanının iki kategoriye doğrusal bir bölümünü oluşturarak yapmaya çalışmaktadır. Yeni görünmeyen nesnelerdeki özelliklere dayanarak, bir kategoriyi hiper düzlemin üstüne veya altına yerleştirir (Sasidharan, 2021; Şener 2017).

K-en yakın komşu algoritmasında ise amaç sorgulanan sınıf etiketini eğitim kümesindeki en yakın komşularına göre belirlemektir. Algoritmanın adımları aşağıda verilmiştir.

1. Seçilen K sayısına göre algoritma başlatılır.
2. Her bir veri için başlangıç komşu arasındaki uzaklık hesaplanır ve listeye eklenir.
3. Uzaklıklara göre nesneler küçükten büyüğe yeniden gruplandırılır ve yeni K en yakın komşuları bulunur.
4. Sınıflandırma için eğitim veri kümesinden K-en yakın komşuların çoğuna atanan bir sınıf etiketi, yeni veri noktası için tahmin edilen bir sınıf olarak kabul edilir.

3.1.3. Performans değerlendirme kriterleri

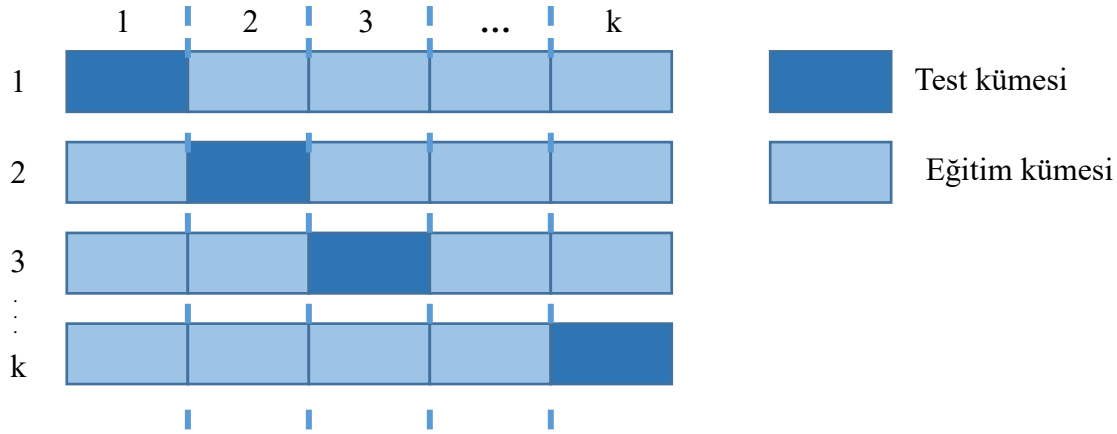
Uygulanan algoritmaların performansını belirleyebilmek için sınıflandırma performans değerlendirme kriterleri kullanılmıştır. Çizelge 3.3'te sınıflandırma performans değerlendirme kriterleri verilmiştir.

Çizelge 3.3. Sınıflandırma performans değerlendirme kriterleri

Performans Kriteri	**Formülü**	**Özellik**
Doğruluk (acc)	$\frac{TP + TN}{TP + FP + TN + FN}$	Doğru tahmin edilen değerlerin toplam veri kümesine oranıdır.
Hata oranı (err)	$\frac{FP + FN}{TP + FP + TN + FN}$	Yanlış tahmin edilen değerlerin toplam veri kümesine oranıdır.
Kesinlik (p)	$\frac{TP}{TP + FP}$	Pozitif olarak tahmin edilen değerlerin gerçekte kaç adedinin Pozitif olduğunu göstermektedir.
Anma (r)	$\frac{TP}{TP + FN}$	Pozitif olarak tahmin edilmesi gereken işlemlerin ne kadarının Pozitif olarak tahmin edildiğini gösteren bir metriktir.
F-ölçütü (FS)	$\frac{2 * p * r}{p + r}$	Kesinlik ve anma değerlerinin harmonik ortalamasını göstermektedir.

Çapraz doğrulama ise modelin yüksek performansının rastgele olup olmadığını gösterir. K-katlı çapraz doğrulamanın temel yaklaşımı aşağıdaki gibidir.

1. Veri seti ilk olarak k farklı alt kümeye bölünür.
2. Verileri eğitmek ve son alt kümeyi test verisi olarak bırakmak için k-1 adet alt küme kullanılır.
3. k adet deney sonucunda ortaya çıkan ortalama hata değeri modelin geçerliliğini belirtmektedir.



Şekil 3.3. K-katlı çapraz doğrulama gösterimi

Çalışma kapsamında k değeri 10 olarak alınmıştır.

3.1.4. Sadece ön işleme yapılmış veriler ile sınıflandırma işlemi

Sadece ön işleme yapılmış veriler ile sınıflandırma işlemi aşamasında, ön işlemeden sonra belirlenen algoritmalar veri seti üzerinde doğrudan çalıştırılmıştır.

Karşılaştırma yapılan Nkonyana ve diğerlerinin (2019) yaptığı çalışmada eğitim/test verisi üzerinde çalışmalar yapılmıştır. Sınıflandırma yapılırken veri seti eğitim (train) ve test setleri olarak ayrılmıştır. Eğitim veri seti üzerinden model oluşturulur ve test veri seti üzerinde yapılan tahminleri test edilir. Nkonyana ve diğerlerinin (2019) yaptığı çalışma ile eşit bir kıyaslama yapılabilmesi için ilk olarak çalışma eğitim/test verisi üzerinde uygulanmıştır. Ancak eğitim/ test veri seti ayrımı rastgele yapılmamış olabilir, bu da aşırı uyum (overfitting) sorununa sebep olabilir. Bu sorunu çözebilmek için veri seti için K-katlı çapraz doğrulama da uygulanmıştır.

Öznitelik seçimi ile her bir öznitelik sayısı için belirlenen özniteliklere algoritmalar karşılaştırma yapılan Nkonyana ve diğerlerinin (2019) yaptığı çalışmada eğitim/test verisi üzerinde çalışmalar yapılmıştır. Sınıflandırma yapılırken veri seti eğitim (train) ve test setleri olarak ayrılmıştır. Eğitim veri seti üzerinden model oluşturulur ve test veri

seti üzerinde yapılan tahminleri test edilir. Nkonyana ve diğerlerinin (2019) yaptığı çalışma ile eşit bir kıyaslama yapılabilmesi için ilk olarak çalışma eğitim/test verisi üzerinde uygulanmıştır. Ancak eğitim/ test veri seti ayrımı rastgele yapılmamış olabilir, bu da aşırı uyum (overfitting) sorununa sebep olabilir. Bu sorunu çözebilmek için veri seti için K-katlı çapraz doğrulama da uygulanmıştır.

Öznitelik seçimi ile her bir öznitelik sayısı için belirlenen özniteliklere algoritmalar uygulanmıştır. Böylelikle en yüksek performans değerlendirme kriterlerini veren algoritma ve öznitelik sayısı tespit edilmiştir.

En yüksek performans değerlendirme kriterlerini (özellikle doğruluk oranı) veren öznitelikler içerisinde yüksek korelasyon değerine sahip olan öznitelikler olabilmektedir. Bu durumda öznitelik seçimine göre önem sırasında daha az önemli olan özniteliğin, seçili özniteliklerden çıkartılmış haline algoritmalar yeniden uygulanmıştır. Algoritmalar uygulandıktan sonra sınıflandırma performans göstergeleri hesaplanmıştır.

3.1.5. K-ortalamalar kümeleme yöntemi kullanılarak sınıflandırma işlemi

Sadece ön işleme yapılmış veriler ile sınıflandırma işleminden sonraki bu aşamada kümelemenin sınıflandırma performansının etkisini araştırmak amacıyla, K-ortalamalar kümeleme yöntemi kullanılmıştır. K-ortalamalar kümeleme yönteminin temel yaklaşımı aşağıda verilmiştir (Kijsipongse ve U-ruekolan, 2012).

1. Veri kümesinden k tane rastgele merkez seçilerek bir ilk kümeleme oluşturulur.
2. Her veri noktasının, tüm merkezlere olan uzaklığı hesaplanır ve ilgili nokta en yakın merkeze atanır.
3. Yeni küme ağırlık merkezlerini kümelere atanan tüm veri noktalarının ortalamasına göre yeniden hesaplanır.
4. Yakınsamaya kadar 2. adımı tekrarlanır.

Veri kümeleri K- ortalamalar yöntemi ile belirlenen küme sayısında çeşitlendirilmiş alt kümelere ayrıştırılmıştır. Uygun K sayısını seçmek için çeşitli metotlar bulunmaktadır. Elbow metodu yaygın olarak kullanılan metotlardandır. Her bir noktanın farklı k değerleri için küme merkezine uzaklıklarının karesi toplamı hesaplanmaktadır. Bu değerlere göre her K değeri için grafik çizilir. Grafik üzerinde toplamlar arasındaki farkın azalmaya başladığı noktaya dirsek noktası denmektedir. Bu dirsek noktası ise en uygun K değeri olarak belirlenmektedir. En yüksek kümeleme performansını sağlayan küme sayısı tespitinin ardından, alt kümeler oluşturulmuş ve her kümenin kendi içindeki sınıflandırma performansı gözlemlenmiştir. Sadece ön işleme yapılmış veriler ile sınıflandırma işleminde elde edilen en yüksek performansa ait algoritma K-ortalamalar kümeleme yöntemi kullanılarak sınıflandırma işleminde de kullanılmıştır.

Sadece ön işleme yapılmış veriler ile sınıflandırma işleminde olduğu gibi yüksek korelasyona sahip öznitelikler için gerekli olan işlem K-ortalamalar kümeleme yöntemi kullanılarak sınıflandırma işleminde de gerçekleştirilmiştir.

K-ortalamalar kümeleme yönteminin dezavantajının k küme sayısının önceden bilinememesi olduğundan bahsedilmişti. Her ne kadar dirsek yöntemi ile çözüm bulunmaya çalışılsa da, literatürde kullanılan diğer bir yöntem olan Hiyerarşik kümeleme (Murtagh ve Legendre, 2014) kullanarak doğrudan elde edilen sonuçların performansını ortaya koyabilmek için ilave çalışma yapılmış ve bunun için de Weka açık kaynak yazılımı kullanılmıştır. Weka, veri madenciliği için popüler bir açık kaynak aracı olup ön işleme ve analiz araçlarını içeren zengin özelliklere sahip bir araç kutusunda bir araya getirilmiş bir veri madenciliği ve makine öğrenimi algoritmaları koleksiyonunu içermektedir. Weka, kullanıcılara bir Grafik Kullanıcı Arayüzü ve Java tabanlı bir Uygulama Programlama Arayüzü sağlamaktadır (Hall, Frank, Holmes, Pfahringer, Reutemann ve Witten, 2009). Weka'da yapılan çalışmada Rassal orman, Adaboost, Bagging, SVM ve KNN algoritmaları kullanılmıştır.

Çizelge 3.4'te Çelik Levha Arızası (Steel Plates Fault) veri seti için Weka çalışma sonuçları verilmiştir. Çizelge 3.4'te verilen;

- “Ham hali” sütunu UCI veri tabanından alınan veri setine belirlenen algoritmaların doğrudan uygulanmasıyla elde edilen doğruluk oranlarını göstermektedir.
- “K-means ortalama” sütunu veri setini K-ortalamalar kümeleme yöntemi ile alt kümelere ayrıldığında her küme ile elde edilen doğruluk oranlarının ortalamasının alınmış halini göstermektedir.
- “Hiyerarşik ortalama” sütunu veri setini hiyerarşik kümeleme yöntemi ile alt kümelere ayrıldığında her küme ile elde edilen doğruluk oranlarının ortalamasının alınmış halini göstermektedir.
- “K-means iyileştirme” sütunu K-ortalamalar yöntemi ile elde edilen doğruluk oranlarının ortalamasının ham halinde elde edilen doğruluk oranına göre iyileştirmesini göstermektedir.
- “Hiyerarşik iyileştirme” sütunu hiyerarşik kümeleme yöntemi ile elde edilen doğruluk oranlarının ortalamasının ham halinde elde edilen doğruluk oranına göre iyileştirmesini göstermektedir.

Çizelge 3.4. Çelik Levha Arızası veri seti Weka çalışma performans sonuçları

	Çapraz doğrulama				
Algoritma	**Ham Hali**	**K-means ortalama**	**Hiyerarşik ortalama**	**K-means iyileştirmesi**	**Hiyerarşik iyileştirmesi**
Rassal Orman	47,96%	50,05%	47,64%	4,36%	-0,68%
Adaboost	44,77%	53,86%	51,86%	20,30%	15,84%
Bagging	34,67%	34,89%	35,75%	0,62%	3,12%
SVM	72,59%	72,25%	72,30%	-0,47%	-0,40%
KNN	64,04%	63,91%	60,08%	-0,20%	-6,18%
Ortalama iyileştirme				4,92%	2,34%

Yapılan çalışma sonucu 5 veri seti için de;

- Kümeleme yöntemleri ile alt kümeler oluşturulduktan sonra sınıflandırma algoritmaları uygulandığında, sadece ön işleme yapılmış halinde elde edilen sonuçlara kıyasla daha yüksek doğruluk oranı elde edilmiştir.
- K-ortalamalar kümeleme yöntemi ile ve hiyerarşik kümeleme yöntemi ile sınıflandırma işlemlerinin performansları kıyaslandığında K-ortalamalar

kümeleme yöntemi ile sınıflandırma işleminde daha yüksek doğruluk oranı elde edilmiştir.

Bu nedenle bu aşamada Pyhton programında hiyerarşik kümeleme yöntemi ile sınıflandırma işleminin gerçekleştirilmesi yerine iki yöntem birleştirilerek melez bir yaklaşım ile performans kriterlerinin seviyesinin irdelenmesine karar verilmiştir.

3.1.6. Melez kümeleme yöntemi kullanılarak sınıflandırma işlemi

K-ortalamalar kümeleme yöntemi kullanılarak sınıflandırma işlemi uygulandıktan sonra veri setleri K-ortalamalar ve hiyerarşik kümeleme yöntemlerinin birlikte kullanıldığı melez kümeleme yöntemleri ile belirlenen küme sayısında çeşitlendirilmiş alt kümelere ayrıştırılmıştır. Bunun nedeni, K-ortalamalar kümeleme yönteminin önceden bilinemeyen küme sayısı ve başlangıç küme merkezleri dezavantajıdır. Melez kümeleme yönteminin temel yaklaşımı aşağıda verilmiştir.

1. Veri setine hiyerarşik kümeleme yöntemi uygulanır.
2. Oluşan dendrograma göre küme sayısı belirlenir ve k kümelerine ayrılır.
3. Her kümenin küme merkezi hesaplanır.
4. İlk küme merkezleri olarak 3.adımda elde edilen küme merkezleri kümesi dikkate alınarak K-ortalamalar uygulanır.

Melez kümelemenin ilk aşaması olan hiyerarşik kümeleme yönteminde Ward yöntemi ve Öklid mesafesi kullanılmıştır. Melez kümeleme yöntemiyle oluşan her kümeye sınıflandırma algoritmaları uygulanıp her kümenin kendi içindeki sınıflandırma performansı gözlemlenmiştir. Sadece ön işleme yapılmış veriler ile sınıflandırma işleminde elde edilen en yüksek performansa ait algoritma melez kümeleme yöntemi kullanılarak sınıflandırma işleminde de kullanılmıştır. Bu işlemlerin sonucunda veri kümesini doğrudan sınıflandıran algoritmaların göstergeleri ile önerilen yöntemler sonucundaki sınıflandırma göstergeleri karşılaştırılmıştır. Önceki iki aşamada olduğu gibi yüksek korelasyona sahip öznitelikler için gerekli olan işlem bu aşamada da gerçekleştirilmiştir.

3.1.7. Parametre optimizasyonu

Sadece ön işleme yapılmış veriler ile ve önerilen yöntemler ile sınıflandırma işlemleri sonucu elde edilen sınıflandırma performansını iyileştirebilmek için her aşama sonunda parametre optimizasyonu yapılmıştır. Izgara arama parametre optimizasyonu global optimal değerini elde etmesine rağmen boyutsal açısından dezavantajlı olması sebebiyle çalışma kapsamında rassal arama parametre optimizasyonu uygulanmıştır.

Her algoritmanın ve her veri setinin/ kümesinin parametreleri farklı olup parametre optimizasyonu, önceki üç aşamada uygulanan ve en yüksek performansa sahip olan algoritma için yapılmıştır.

Çalışmanın performansının ortaya konulabilmesi için ilk olarak Steel Plates Fault veri setine uygulanan bu çalışma, Nkonyana ve diğerlerinin (2019) yaptığı çalışma sonucu ile kıyaslanmıştır.

3.2. Diğer veri setleri

UCI veri tabanından alınan diğer veri setleri için de Steel Plates Fault veri seti için uygulanan benzer çalışmalar gerçekleştirilerek bu çalışmaların ne kadar fayda sağladığı ölçülüp çalışmanın faydası ortaya çıkarılmıştır.

4. BULGULAR

4.1. Çelik Levha Arızası Veri Seti

Çelik Levha Arızası (Steel Plates Fault) veri seti için yapılan çalışmalar sonucunda elde edilen bulgulardan bu bölümde bahsedilmiştir.

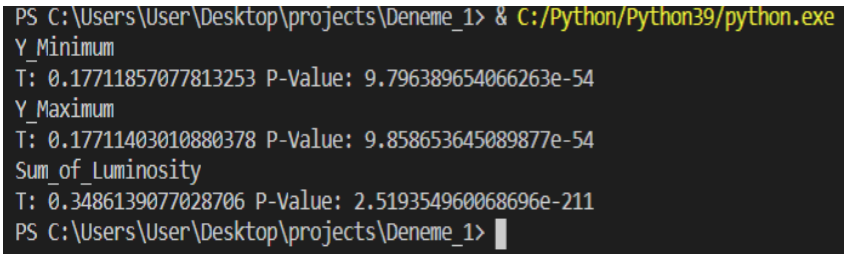
4.1.1. Ön işleme

Eksik veri analizi için Steel Plates Fault veri seti incelenmiştir ve eksik herhangi bir veri bulunmamıştır.

Steel Plates Fault veri seti içi aykırı veri analizi yapabilmek için ilk olarak veri setinin normal dağılıma uyup uymadığı Kolmogorov Smirnov testi ile analiz edilmiştir. Kolmogorov Smirnov testi için oluşturulan hipotezler aşağıdaki gibidir.

H_0: Değişken normal dağılıma sahiptir, p-değer > 0,05
H_1: Değişken normal dağılıma sahip değildir, p-değer < 0,05

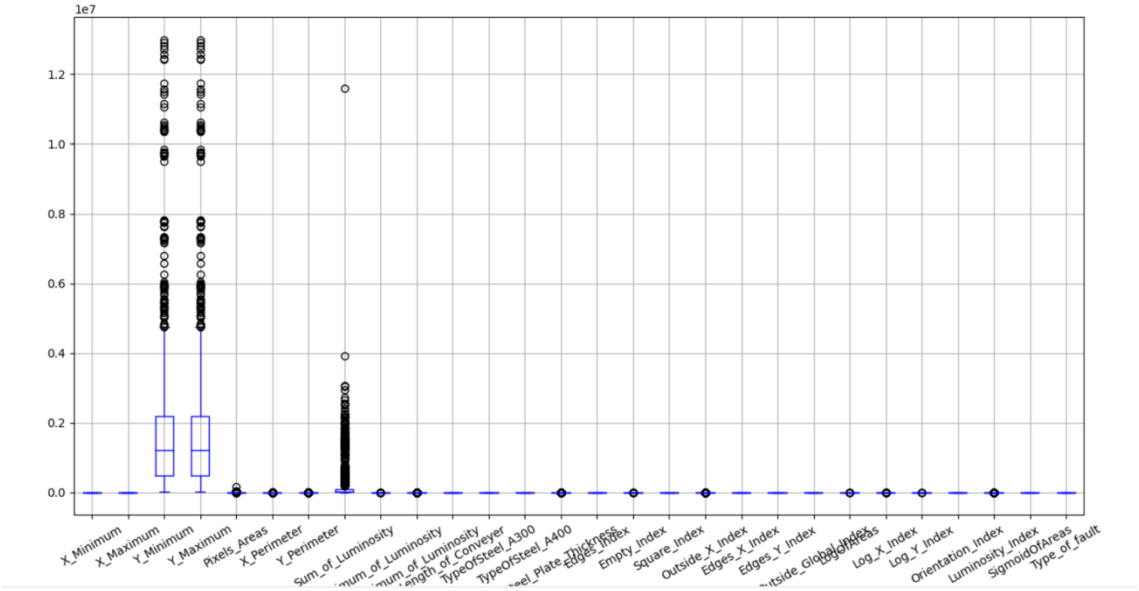
Uygulanan Kolmogorov Smirnov test sonuçları Şekil 4.1'de verilmiştir. Bu sonuca göre p-değerleri 0,05'ten büyük olduğu için H_0 hipotezi kabul edilir ve veri setinin normal dağılıma uyduğu söylenebilmektedir.



Şekil 4.1. Kolmogorov Smirnov test sonuçları

Kolmogorov Smirnov test sonucuna göre veri seti normal dağılıma uymaktadır, bu nedenle Z-skor yöntemi aykırı veri analizi için kullanılabilecektir. Veri seti normal dağılıma uyduğu için aykırı veri analizi için Z-skor kullanılabilmiştir.

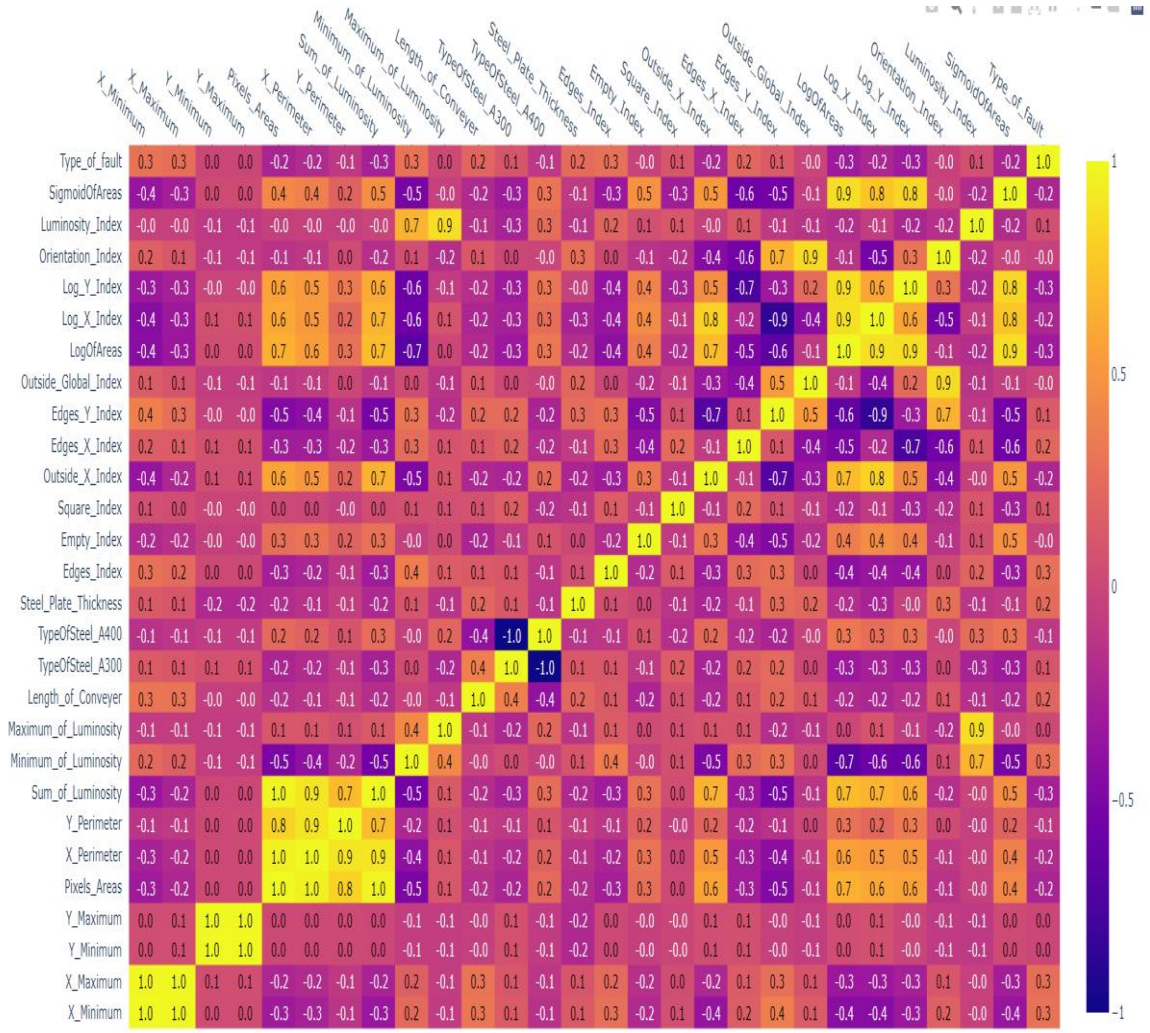
Şekil 4.2'de görüldüğü üzere Z-skor ile aykırı veri analizi yapıldığında veri seti içerisinde özellikle 3 öznitelikte (Y_Minimum, Y_Maksimum ve Sum_of_Luminosity) aykırı verilerin olduğu gözlemlenmiştir.



Şekil 4.2. Aykırı veri analizi sonuçları

UCI veri tabanından alınan veri seti için yapılan aykırı veri analizi ve normallik testi sonucuna rağmen veri setinde çıkarılacak herhangi bir aykırı verinin önem derecesi bilinmediği için aykırı verilerin veri setinden çıkarılmamasının daha doğru olacağına karar verilmiştir.

Veri setinde öznitelikler arası ilişkiyi görebilmek için yapılan korelasyon matrisine göre öznitelikler arası korelasyon oranları Şekil 4.3'te verilmiştir.



Şekil 4.3. Korelasyon matrisi sonuçları

Öznitelikler arası korelasyon oranı %90 değerinin üzerinde olan öznitelikler yüksek korelasyona sahip öznitelikler olarak nitelendirilmiştir. Veri seti için elde edilen korelasyon matrisine göre aşağıda verilenler öznitelikler arasında %91 ve üzeri korelasyon bulunmaktadır. Çalışma kapsamında sonraki adımda yapılacak olan öznitelik seçimine göre en iyi öznitelikler belirlendiğinde yüksek korelasyona sahip öznitelikler aynı anda bulunuyor ise bu özniteliklerden öznitelik seçimine göre daha önemsiz olan öznitelik çıkartılarak algoritma sonuçları tekrar elde edilmiştir.

- Y_Minimum ve Y_Maximum,
- X_ Minimum ve X_Maximum,
- Pixels_Areas ve Sum_of_Luminosity,

- Pixels_Areas ve X_Perimeter,
- X_Perimeter ve Sum_of_Luminosity,
- X_Perimeter ve Y_Perimeter,
- TypeOfSteel_A300 ve TypeOfSteel_A400

Ön işleme olarak uygulanan son işlem ise öznitelik seçimidir. Her bir öznitelik sayısı için en iyi öznitelikler seçilmiştir. Öznitelik sayısı 19 için seçilen öznitelikler önem sırasıyla Çizelge 4.1'de verilmiştir.

Çizelge 4.1. 19 öznitelik seçimi sonucu belirlenen öznitelikler

Öznitelik Seçimi Sonrası Seçilen Öznitelikler (k=19)		
1. Log_X_Index	8. Log_Y_Index	15. Minimum_of_Luminosity
2. LogOfAreas	9. X_Minimum	16. X_Perimeter
3. Edges_Y_Index	10. Pixels_Areas	17. Edges_Index
4. Outside_X_Index	11. SigmoidOfAreas	18. Steel_Plate_Thickness
5. TypeOfSteel_A300	12. Orientation_Index	19. Square_Index
6. TypeOfSteel_A400	13. X_Maximum	
7. Sum_of_Luminosity	14. Length_of_Conveyer	

4.1.2. Sadece ön işleme yapılmış veriler ile sınıflandırma işlemi

Nkonyana ve diğerlerinin (2019) yaptığı çalışma ile eşit bir kıyaslama yapılabilmesi için;

- İlk olarak çalışma, eğitim/test verisi üzerinde, daha sonra K-katlı çapraz doğrulama üzerine uygulanmıştır.
- Eğitim/test veri seti için uygulanırken referans makaledeki Rassal Orman, MLP ve SVM algoritmaları uygulanmıştır.

Eğitim/test veri seti için uygulanan algoritmalar, hem verinin veri tabanından çekildiği ham haline, hem de öznitelik seçimi ile elde edilen verilere uygulanmıştır. Elde edilen sonuçlar Çizelge 4.2'de verilmiştir.

Çizelge 4.2'de;

- "Tüm öznitelikler" sütunu, algoritmaların tüm özniteliğe uygulandığında elde edilen sonuçları göstermektedir.
- "19 öznitelik" sütunu, uygulanan üç algoritma içerisinde en yüksek doğruluk oranını veren Rassal orman algoritmasında en yüksek doğruluk oranını veren öznitelik sayısı için elde edilen sonuçları göstermektedir.
- En yüksek doğruluk oranını veren 19 öznitelik içerisinde Xminimum ile yüksek korelasyona sahip olan Xmax özniteliği bulunmaktadır. "(-Xmax)" sütunu da, Xminimum özniteliği Çizelge 4.1'e göre daha önemli olduğu için 19 öznitelik içerisinden Xmax özniteliğinin çıkartılmış halini göstermektedir.
- En yüksek doğruluk oranını veren 19 öznitelik içerisinde X_Perimeter ve Sum_of_Luminosity öznitelikleri ile yüksek korelasyona sahip olan Pixel Area özniteliği bulunmaktadır. "(-Pixel Area)" sütunu da, diğer iki öznitelik de Çizelge 4.1 e göre daha önemli olduğu için 19 öznitelik içerisinden Pixel Area özniteliğinin çıkartılmış halini göstermektedir.
- En yüksek doğruluk oranını veren 19 öznitelik içerisinde Sum_of_Luminosity ile yüksek korelasyona sahip olan X_Perimeter özniteliği bulunmaktadır. "(-X Perimeter)" sütunu da, Sum_of_Luminosity özniteliği Çizelge 4.1'e göre daha önemli olduğu için 19 öznitelik içerisinden X_Perimeter özniteliğinin çıkartılmış halini göstermektedir.
- En yüksek doğruluk oranını veren 19 öznitelik içerisinde TypeOfSteel_A300 ile yüksek korelasyona sahip olan TypeOfSteel_A400 özniteliği bulunmaktadır. "(-A 400)" sütunu da, TypeOfSteel_A300 özniteliği Çizelge 4.1'e göre daha önemli olduğu için 19 öznitelik içerisinden TypeOfSteel_A400 özniteliğinin çıkartılmış halini göstermektedir.

Çizelge 4.2. Eğitim/test veri setlerine uygulanan sınıflandırma algoritmaları için doğruluk oranları

	Tüm Öznitelik	**19 Öznitelik**	**(-Xmax)**	**(-Pixel Area)**	**(-X Perimeter)**	**(- A400)**
SVM	54,55%	53,86%	54,72%	52,32%	51,80%	53,86%
MLP	50,94%	54,72%	56,60%	-	-	55,23%
RF	78,39%	78,39%	78,22%	**80,10%**	78,04%	79,25%

Çizelge 4.2'de doğruluk oranlarına bakıldığında en yüksek doğruluk oranına Rassal Orman ve 18 öznitelik (19 öznitelik- Pixel Area) ile ulaşıldığı görülmektedir (% 80,10).

Literatürdeki sonuçlar ile karşılaştırma yapabilmek için eğitim/test veri seti üzerinde çalışılmış olup elde edilen sonuçlar sonraki bölümde karşılaştırma yapabilmek amacıyla kullanılmıştır. Eğitim/veri seti üzerinde çalışıldıktan sonra yapılan tüm çalışmalar K-katlı çapraz doğrulama ile yapılmıştır. Bu doğrultuda K-katlı çapraz doğrulama ile uygulanan algoritma bazında elde edilen en yüksek doğruluk oranları Çizelge 4.3'te verilmiştir.

Çizelge 4.3. Algoritma bazında en yüksek doğruluk oranları

Algoritma	**En Yüksek Doğruluk Oranı (%)**	**En iyi Öznitelik Sayısı**
Rassal Orman	**%66,36**	19
KNN	%53,12	5
Bagging	%64,34	24
Adaboost	%48,69	23
CART	%59,86	22
SVM	%55,59	5

En yüksek doğruluk oranı Rassal Orman algoritması için elde edilmiştir. Bu nedenle, bundan sonraki işlemler Rassal Orman algoritması kullanılarak yapılmıştır. Çizelge 4.4'te ise Rassal Orman algoritmasında her bir öznitelik sayısı için seçilen öznitelikler için doğruluk oranları ve öznitelik seçimi yapılmadığı yani tüm öznitelikleri içeren

haline göre iyileştirme yüzdeleri verilmiştir. Hiçbir işlem yapılmadan sadece algoritma uygulandığında elde edilen doğruluk oranı % 65,48'dir.

Çizelge 4.4. Öznitelik seçimine göre doğruluk oranları (Rassal orman algoritması)

Rassal Orman		
Öznitelik sayısı	**Rassal Orman Sonucu**	**Öznitelik seçimsiz haline göre iyileştirme**
1	47.03%	-28.18%
2	47.03%	-28.18%
3	49.20%	-24.86%
4	52.24%	-20.22%
5	57.13%	-12.75%
6	57.75%	-11.81%
7	57.49%	-12.20%
8	57.86%	-11.64%
9	61.15%	-6.61%
10	61.51%	-6.06%
11	61.51%	-6.06%
12	61.72%	-5.74%
13	62.03%	-5.27%
14	64.19%	-1.97%
15	63.99%	-2.28%
16	64.50%	-1.50%
17	65.53%	0.08%
18	65.89%	0.63%
19	66.36%	1.34%
20	66.35%	1.33%
21	66.10%	0.95%
22	65.84%	0.55%
23	66.15%	1.02%
24	66.15%	1.02%
25	66.20%	1.10%
26	65.63%	0.23%
27	65.48%	0.00%

Rassal Orman algoritması için en yüksek doğruluk oranını veren 19 öznitelik içinde yüksek korelasyon değerine sahip özniteliklere dikkat edilip gerekli işlem uygulanmıştır. Çizelge 4.5'te tüm öznitelikler için, en iyi öznitelik sayısı için ve en iyi öznitelik sayısı içerisinde yüksek korelasyona sahip özniteliklere göre işlem uygulandığında elde edilen doğruluk oranları ve bu doğruluk oranlarının tüm özniteliklerde elde edilen doğruluk oranına göre iyileştirme oranları verilmiştir.

İyileştirme oranı tüm özniteliklerin doğruluk oranı baz alınarak hesaplanmıştır. Çizelge 4.2'de oluşturulan gösterimde olduğu gibi Çizelge 4.5'te de,

- 18 Öznitelik (-Xmax) satırı, 19 öznitelik içerisinden öznitelikler arası korelasyona dikkat edilerek Xmaximum özniteliğinin çıkartılmış halini göstermektedir.
- 18 Öznitelik (-Pixel Area) satırı, 19 öznitelik içerisinden öznitelikler arası korelasyona dikkat edilerek Pixel Area özniteliğinin çıkartılmış halini göstermektedir.
- 18 Öznitelik (-X Perimeter) satırı, 19 öznitelik içerisinden öznitelikler arası korelasyona dikkat edilerek X Perimeter özniteliğinin çıkartılmış halini göstermektedir.
- 18 Öznitelik (-A400) satırı, 19 öznitelik içerisinden öznitelikler arası korelasyona dikkat edilerek TypeOfSteel_A400 özniteliğinin çıkartılmış halini göstermektedir.

Çizelge 4.5. Rassal Orman algoritması ile sadece ön işleme ile sınıflandırma işlemi için doğruluk oranları ve iyileştirme oranları

Öznitelikler	RF Doğruluk Oranı(%)	İyileştirme Oranı(%)
Tüm Öznitelik	65,48%	-
19 Öznitelik	66,36%	1,34%
18 Öznitelik (-Xmax)	**66,98%**	2,29%
18 Öznitelik (-Pixel Area)	65,27%	-0,32%
18 Öznitelik (-X Perimeter)	65,69%	0,32%
18 Öznitelik (-A400)	65.73%	0,38%

Tüm öznitelikler için, 19 öznitelik için ve 18 öznitelik (19 öznitelik- Pixel Area) için kesinlik, anma ve F-ölçütü sonuçları ise Çizelge 4.6'da verilmiştir. Elde edilen kesinlik, anma ve F- ölçüt değerleri daha sonra Sonuç bölümünde karşılaştırma amacıyla kullanılmıştır.

Çizelge 4.6. Uygulanan sınıflandırma algoritması için diğer performans kriterleri sonuçları

Öznitelik	Kesinlik(%)	Anma(%)	F-ölçüt (%)
Tüm öznitelikler	%78	%78	%78
19 öznitelik	%79	%78	%78
19 öznitelik- Pixel Area	%80	%80	%80

Sınıflandırma doğruluk oranını daha fazla iyileştirebilmek amacıyla parametre optimizasyonu uygulanabilmesi için ilk olarak Rassal Orman algoritmasının parametreleri incelenmiştir.

Rassal Orman algoritmasında yaygın olarak kullanılan dört ana parametre bulunmaktadır. Bunlar "n_estimators, min_samples_split, max_depth ve max features" tır. Çalışma kapsamında ayrıca öznitelik seçimi yapıldığı için bu parametrelerden max features incelenmemiştir.

- n_estimators (Varsayılan: 10): Oluşturmak istenilen ağaç sayısını göstermektedir. Bu değer yükseldikçe programın süresi de uzayacaktır.
- max_depth (Varsayılan: None) : Oluşturulacak ağaçların derinliğini (Ağacın kökü ve yapraklar arasındaki maksimum bağlantı sayısı) göstermektedir. Aşırı uyumu önleyebilmek için ayarlanması gerekmektedir.
- min_samples_split (Varsayılan: 2): Bir bölünmenin gerçekleşmesi için bulunması gereken minimum örnek sayısını göstermektedir.

Steel Plates Fault veri seti için yapılan denemeler sonucunda, bu parametreler için belirlenen değerler ve değer aralıkları Çizelge 4.7'de verilmiştir. Ayrıca model içerisinde K-katlı çapraz doğrulama için k 10 olarak alınmıştır.

Çizelge 4.7. Rassal orman parametreleri ve seviyeleri

Parametre	n_estimator	max_depth	min_samples_split
Seviye	[100,200,300,400,500,1000]	(10,30)	(2,20)

Parametre optimizasyonu; tüm özniteliklere, Rassal Orman algoritması için en yüksek doğruluk oranını veren 19 özniteliğe ve 19 öznitelik içinde yüksek korelasyon değerine sahip özniteliklere dikkat edilip daha az öneme sahip öznitelikler 19 öznitelik içerisinden çıkartılarak kalan 18 öznitelik için de uygulanmıştır.

Yapılan parametre optimizasyonlarına göre elde edilen doğruluk oranları Çizelge 4.8'de verilmiştir. Çizelge 4.8'de verilen;

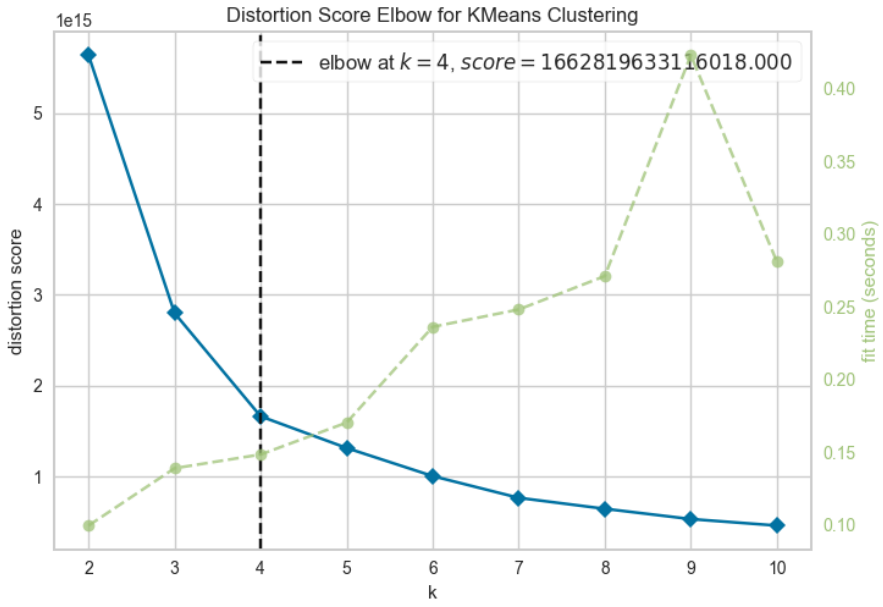
- "RF Doğruluk Oranı(%)" sütunu farklı öznitelikler için parametre optimizasyonu ile elde edilen Rassal orman algoritması doğruluk oranlarını göstermektedir.
- "İyileştirme (%)(CV parametre optimizasyonu)" sütunu parametre optimizasyonu ile elde edilen tüm öznitelikler için doğruluk oranı (% 68,41) baz alınarak elde edilen iyileştirme oranlarını göstermektedir.
- "İyileştirme (%)(CV ilk hali)" sütunu ise parametre optimizasyonu yapılmadan tüm öznitelikler ile elde edilen (%65,48) doğruluk oranı baz alınarak elde edilen iyileştirme oranlarını göstermektedir.

Çizelge 4.8. Sadece ön işleme yapılmış hali ile Rassal Orman algoritması parametre optimizasyonu için doğruluk oranları ve iyileştirme oranları

Algoritma	**RF Doğruluk Oranı(%)**	**İyileştirme (%) (CV parametre optimizasyonu)**	**İyileştirme (%) (CV ilk hali)**
Tüm Öznitelik	68,41%	-	4,47%
19 Öznitelik	68,58%	0,25%	4,73%
18 Öznitelik (-Xmax)	**68,76%**	0,26%	5,01%
18 Öznitelik (-Pixel Area)	68,35%	-0,60%	4,38%
18 Öznitelik (-X Perimeter)	68,70%	0,51%	4,92%
18 Öznitelik (-A400)	68,59%	-0,16%	4,75%

4.1.3. K-ortalamalar kümeleme yöntemi kullanılarak sınıflandırma işlemi

Sadece ön işleme yapılmış veriler ile sınıflandırma işlemi sonuçlarından sonra veri kümeleri K- ortalamalar yöntemi ile belirlenen küme sayısında çeşitlendirilmiş alt kümelere ayrıştırılmıştır. Şekil 4.4'te verilen Elbow yöntemi grafiğinde mavi çizgi küme merkezine uzaklıklarının karesi toplamları arasındaki farkı, siyah kesikli çizgi dirsek noktasını, yeşil kesikli çizgi ise zamanı göstermektedir. Küme sayısını belirlemek için kullanılan Elbow yöntemine göre küme merkezine uzaklıklarının karesi toplamları arasındaki farkın azalmaya başladığı dirsek noktasına göre küme sayısı 4 olmalıdır.



Şekil 4.4. Elbow yöntemi ile küme sayısının belirlenmesi

Dört küme sayısına göre veri setine K-ortalamalar kümeleme yöntemi uygulanıp 4 alt küme elde edilmiştir. Sadece ön işleme yapılmış veriler ile sınıflandırma işleminde Rassal Orman algoritması için K-katlı çapraz doğrulama ile uygulanılan tüm işlemler, elde edilen her bir alt küme için de uygulanmıştır. Her aşamada 4 alt küme için elde edilen doğruluk oranları ortalaması alınarak ortalama doğruluk oranları elde edilmiştir. Alt küme bazında elde edilen doğruluk oranları, alt kümelerin ortalama doğruluk oranları ve sadece ön işleme ile sınıflandırma işleminde tüm öznitelikler için elde edilen doğruluk oranı (%65,48) baz alınarak iyileştirme yüzdeleri Çizelge 4.9'da verilmiştir.

Çizelge 4.9. Rassal orman algoritması ile K-ortalamalar kümeleme yöntemi kullanılarak sınıflandırma işlemi sonuçları

Öznitelikler	Küme	RF Doğruluk Oranı(%)	Ortalama(%)	İyileştirme(%)
Tüm Öznitelikler	Küme 0	72,91%	68,14%	4,06%
	Küme 1	72,73%		
	Küme 2	61,88%		
	Küme 3	65,03%		
19 Öznitelik	Küme 0	72,05%	67,95%	3,78%
	Küme 1	72,73%		
	Küme 2	59,90%		
	Küme 3	67,13%		
18 Öznitelik (Xmaxsız)	Küme 0	72,48%	69,74%	6,51%
	Küme 1	79,55%		
	Küme 2	59,41%		
	Küme 3	67,53%		
18 Öznitelik (Pixel Areasız)	Küme 0	72,19%	69,15%	5,60%
	Küme 1	75,00%		
	Küme 2	61,88%		
	Küme 3	67,53%		
18 Öznitelik (Xperimetersız)	Küme 0	71,47%	**69,86%**	6,69%
	Küme 1	79,55%		
	Küme 2	60,89%		
	Küme 3	67,53%		
18 Öznitelik(A400süz)	Küme 0	71,90%	68,12%	4,02%
	Küme 1	72,73%		
	Küme 2	59,90%		
	Küme 3	67,93%		

Oluşturulan her bir kümenin özelliklerine göre uyarlanan parametre seviyeleri ile gerçekleştirilen parametre optimizasyonu sonrası elde edilen doğruluk oranları Çizelge 4.10'da verilmiştir.

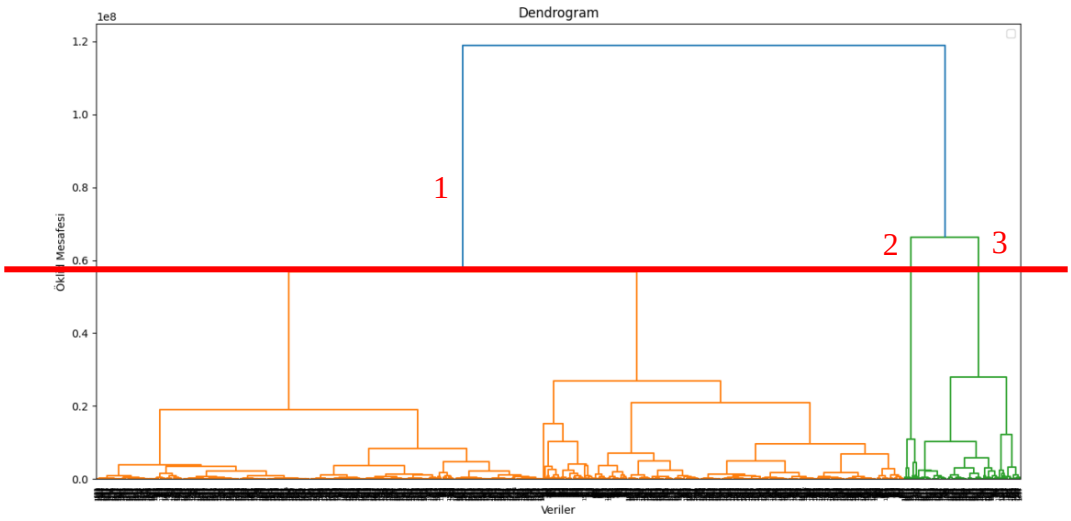
Çizelge 4.10. K-ortalamalar kümeleme yöntemine göre Rassal Orman algoritması parametre optimizasyonu ile doğruluk oranları ve iyileştirme oranları

Öznitelikler	Küme	RF Doğruluk Oranı(%)	Ortalama(%)	İyileştirme(%)
Tüm Öznitelikler	Küme 0	77,14%	74,97%	14,50%
	Küme 1	82,50%		
	Küme 2	70,32%		
	Küme 3	69,93%		
19 Öznitelik	Küme 0	76,17%	75,09%	14,67%
	Küme 1	82,50%		
	Küme 2	70,41%		
	Küme 3	71,26%		
18 Öznitelik (Xmaxsız)	Küme 0	76,28%	**75,52%**	15,34%
	Küme 1	85,00%		
	Küme 2	69,77%		
	Küme 3	71,04%		
18 Öznitelik (Pixel Areasız)	Küme 0	75,69%	75,50%	15,30%
	Küme 1	85,00%		
	Küme 2	70,38%		
	Küme 3	70,92%		
18 Öznitelik (Xperimetersız)	Küme 0	75,97%	74,96%	14,47%
	Küme 1	82,50%		
	Küme 2	70,44%		
	Küme 3	70,92%		
18 Öznitelik(A400süz)	Küme 0	76,33%	75,14%	14,75%
	Küme 1	85,00%		
	Küme 2	68,74%		
	Küme 3	70,48%		

4.1.4. Melez kümeleme yöntemi kullanılarak sınıflandırma işlemi

K-ortalamalar kümeleme yöntemi kullanılarak sınıflandırma işlemi uygulamasının sonuçlarından sonra başlangıç küme merkezlerinin rastgele belirlenmesinin önüne geçebilmek için Phyton programı kullanılarak melez kümeleme yöntemi ile sınıflandırma işlemi gerçekleştirilmiştir. Veri setleri melez kümeleme yöntemi ile belirlenen başlangıç kümeleri ile belirlenen küme sayısında çeşitlendirilmiş alt kümelere ayrıştırılmıştır. Melez kümelemenin ilk aşaması olan hiyerarşik kümeleme sonucunda elde edilen dendrogram Şekil 4.5'te verilmiştir. Dendrogram ile küme sayısı, başka bir çizgi ile kesilmeyen en uzun dikey uzaklık kullanılarak belirlenir. Belirlenen noktadan yatay çizgi çizilir ve kesiştiği dikey çizgiler sayılır. Bu sayıların toplamı olması gereken

küme sayısını gösterir. Bu doğrultuda Şekil 4.5'teki dendrogram incelendiğinde küme sayısı 3 olarak belirlenmiştir.



Şekil 4.5. Melez kümeleme dendrogramı

Hiyerarşik kümeleme yönteminden küme merkezlerinin belirlenmesi için de yararlanılmıştır. Hiyerarşik kümeleme yönteminde elde edilen başlangıç küme merkezleri, K-ortalamalar kümeleme yönteminde başlangıç küme merkezleri olarak kullanılmıştır.

Üç küme sayısına göre veri setine melez kümeleme yöntemi uygulanıp 3 alt küme elde edilmiştir. Önceki aşamalarda K-katlı çapraz doğrulama ile uygulanılan tüm işlemler, elde edilen her bir alt küme için uygulanmıştır. Her aşamada 3 alt kümenin ortalaması alınarak ortalama doğruluk oranları elde edilmiştir. Çizelge 4.11'de doğruluk oranı sonuçları verilmiştir.

Çizelge 4.11. Rassal orman algoritması ile melez kümeleme yöntemi kullanılarak sınıflandırma işlemi sonuçları (k = 3)

Öznitelikler	Küme	RF Doğruluk Oranı(%)	Ortalama (%)	İyileştirme (%)
Tüm Öznitelikler	Küme 0	69,73%	71,12%	8,61%
	Küme 1	67,54%		
	Küme 2	76,09%		
19 Öznitelik	Küme 0	70,79%	70,75%	8,04%
	Küme 1	67,54%		
	Küme 2	73,91%		
18 Öznitelik (Xmaxsız)	Küme 0	70,64%	**71,37%**	8,99%
	Küme 1	67,37%		
	Küme 2	76,09%		
18 Öznitelik (Pixel Areasız)	Küme 0	69,89%	71,23%	8,79%
	Küme 1	67,72%		
	Küme 2	76,09%		
18 Öznitelik (Xperimetersız)	Küme 0	71,17%	70,99%	8,41%
	Küme 1	67,89%		
	Küme 2	73,91%		
18 Öznitelik (A400süz)	Küme 0	70,26%	71,12%	8,61%
	Küme 1	67,01%		
	Küme 2	76,09%		

Oluşturulan her bir kümenin özelliklerine göre uyarlanan parametre seviyeleri ile gerçekleştirilen parametre optimizasyonu sonrası elde edilen doğruluk oranları Çizelge 4.12'de verilmiştir.

Çizelge 4.12. Melez kümeleme yöntemine göre Rassal Orman algoritması parametre optimizasyonu ile doğruluk oranları ve iyileştirme oranları (k = 3)

Öznitelikler	Küme	RF Doğruluk Oranı(%)	Ortalama(%)	İyileştirme(%)
Tüm Öznitelikler	Küme 0	71,82%	74,66%	14,01%
	Küme 1	70,15%		
	Küme 2	82.00%		
19 Öznitelik	Küme 0	72,27%	74,43%	13,67%
	Küme 1	67,03%		
	Küme 2	84,00%		
18 Öznitelik (Xmaxsız)	Küme 0	72,69%	**75,50%**	15,31%
	Küme 1	67,82%		
	Küme 2	86,00%		
18 Öznitelik (Pixel Areasız)	Küme 0	72,19%	74,98%	14,50%
	Küme 1	67,24%		
	Küme 2	85,50%		
18 Öznitelik (Xperimetersız)	Küme 0	72,61%	75,29%	14,98%
	Küme 1	67,25%		
	Küme 2	86,00%		
18 Öznitelik (A400süz)	Küme 0	71,91%	75,24%	14,91%
	Küme 1	67,82%		
	Küme 2	86,00%		

Elbow yöntemine göre ise küme sayısı 4 olarak belirlenmiştir. Hiyerarşik kümeleme aşamasında küme sayısı 4 olarak verilip yeni küme merkezleri elde edilmiştir. Elde edilen 4 küme merkezi kullanılarak K-ortalamalar kümeleme yöntemi uygulanmıştır.

Dört küme sayısına göre veri setine melez kümeleme yöntemi uygulanıp 4 alt küme elde edilmiştir. Önceki aşamalarda K-katlı çapraz doğrulama ile uygulanılan tüm işlemler her bir alt küme için de uygulanmıştır. Her aşamada 4 alt kümenin ortalaması alınarak ortalama doğruluk oranları elde edilmiştir. Çizelge 4.13'te doğruluk oranı sonuçları verilmiştir.

Çizelge 4.13. Rassal orman algoritması ile melez kümeleme yöntemi kullanılarak sınıflandırma işlemi sonuçları (k = 4)

Öznitelikler	Küme	RF Doğruluk Oranı(%)	Ortalama(%)	İyileştirme(%)
Tüm Öznitelikler	Küme 0	63,32%	70,55%	7,74%
	Küme 1	72,46%		
	Küme 2	80,56%		
	Küme 3	65,85%		
19 Öznitelik	Küme 0	62,81%	71,50%	9,19%
	Küme 1	72,03%		
	Küme 2	83,33%		
	Küme 3	67,81%		
18 Öznitelik (Xmaxsız)	Küme 0	63,32%	**72,43%**	10,61%
	Küme 1	72,46%		
	Küme 2	86,11%		
	Küme 3	67,81%		
18 Öznitelik (Pixel Areasız)	Küme 0	62,81%	72,01%	9,97%
	Küme 1	71,30%		
	Küme 2	86,11%		
	Küme 3	67,81%		
18 Öznitelik (Xperimetersız)	Küme 0	63,32%	71,39%	9,03%
	Küme 1	71,01%		
	Küme 2	83,33%		
	Küme 3	67,91%		
18 Öznitelik (A400süz)	Küme 0	63,32%	71,49%	9,18%
	Küme 1	71,30%		
	Küme 2	83,33%		
	Küme 3	68,01%		

Oluşturulan her bir kümenin özelliklerine göre uyarlanan parametre seviyeleri ile gerçekleştirilen parametre optimizasyonu sonrası elde edilen doğruluk oranları Çizelge 4.14'te verilmiştir.

Çizelge 4.14. Melez kümeleme yöntemine göre Rassal Orman algoritması parametre optimizasyonu ile doğruluk oranları ve iyileştirme oranları (k = 4)

Öznitelikler	Küme	RF Doğruluk Oranı(%)	Ortalama(%)	İyileştirme(%)
Tüm Öznitelikler	Küme 0	73,33%	76,14%	16,28%
	Küme 1	75,22%		
	Küme 2	85,00%		
	Küme 3	71,00%		
19 Öznitelik	Küme 0	75,00%	77,32%	18,09%
	Küme 1	74,41%		
	Küme 2	88,33%		
	Küme 3	71,55%		
18 Öznitelik (Xmaxsız)	Küme 0	75,00%	**77,43%**	18,25%
	Küme 1	74,73%		
	Küme 2	88,33%		
	Küme 3	71,66%		
18 Öznitelik (Pixel Areasız)	Küme 0	75,00%	76,46%	16,77%
	Küme 1	74,41%		
	Küme 2	85,00%		
	Küme 3	71,44%		
18 Öznitelik (Xperimetersız)	Küme 0	73,89%	76,35%	16,60%
	Küme 1	74,73%		
	Küme 2	85,00%		
	Küme 3	71,77%		
18 Öznitelik (A400süz)	Küme 0	73,89%	76,24%	16,43%
	Küme 1	74,73%		
	Küme 2	85,00%		
	Küme 3	71,33%		

Tartışma ve Sonuç bölümünde Çizelge 4.5, Çizelge 4.8, Çizelge 4.9, Çizelge 4.10, Çizelge 4.11, Çizelge 4.12, Çizelge 4.13 ve Çizelge 4.14 birleştirilip karşılaştırma yapılmıştır.

4.2. Diğer Veri Setleri

Steel Plates Fault veri setine K-katmanlı çapraz doğrulama ile uygulanan sadece ön işleme yapılmış veriler ile, K-ortalamalar kümeleme ile ve önerilen melez kümeleme yöntemi ile sınıflandırma işlemleri ve parametre optimizasyonları UCI veri tabanından alınan diğer veri setleri için de gerçekleştirilmiştir.

Omurga (Vertebral Column) veri seti için Weka programında yapılan çalışma sonuçları Çizelge 4.15'te verilmiştir.

Çizelge 4.15. Omurga veri seti Weka çalışma performans sonuçları

Vertebral Column					
	Cross validation Fold				
	Ham hali	**K-means ort.**	**Hiyerarşik ort.**	**K-means iyileştirmesi**	**Hiyerarşik iyileştirmesi**
Rassal Orman	67,74%	71,02%	68,03%	4,84%	0,42%
Adaboost	67,42%	70,75%	68,30%	4,93%	1,30%
Bagging	67,74%	70,75%	68,03%	4,44%	0,42%
SVM	78,71%	72,67%	67,38%	-7,67%	-14,39%
KNN	81,61%	71,30%	67,75%	-12,64%	-16,98%
Ortalama iyileştirme				**-1,22%**	**-5,85%**

Vertebral Column veri seti için Pyhton programında belirlenen algoritmalar kullanılarak gerçekleştirilen sadece ön işleme ile sınıflandırma işlemi ile algoritmalar için elde edilen en yüksek doğruluk oranı ve öznitelik sayısı Çizelge 4.16'da verilmiştir.

Çizelge 4.16. Omurga veri seti için algoritma bazında en yüksek doğruluk oranları

Algoritma	**En Yüksek Doğruluk Oranı (%)**	**En iyi Öznitelik Sayısı**
Rassal Orman	%83,23	5
KNN	%82,90	4
Bagging	**%83,87**	5
Adaboost	%80,97	6
CART	%79,68	6
SVM	%80,32	6

Çizelge 4.16'ya göre en yüksek doğruluk oranı Bagging algoritması ve 5 öznitelik seçimi ile elde edilmiştir. Bu nedenle, bundan sonraki aşamalarda yapılan çalışmalar da Bagging algoritması kullanılarak yapılmıştır. Vertebral Column veri seti için hazırlanan

korelasyon matrisine göre korelasyon oranı %90 üzerinde bulunan öznitelikler bulunmamıştır.

Vertebral Column veri seti için gerçekleştirilen sadece ön işleme ile sınıflandırma işlemi sonucu elde edilen kesinlik, anma ve F-ölçüt sonuçları Çizelge 4.17'de verilmiştir.

Çizelge 4.17. Omurga veri seti kesinlik, anma ve F-ölçüt sonuçları

Öznitelik	**Kesinlik(%)**	**Anma(%)**	**F-ölçüt (%)**
Tüm öznitelikler	%83	%84	%83
5 öznitelik	%84	%84	%84

Bagging algoritmasında önemli olan parametreler "n_estimators" ve "max_samples" dır.

- n_estimators (varsayılan = 10) Topluluktaki temel tahmin edicilerin sayısını göstermektedir.
- max_samples (varsayılan = 1.0) Her bir temel tahminciyi eğitmek için veri setinden çekilecek örnek sayısını göstermektedir. Genelde veri setindeki veri sayısının yarı miktarına kadar ayarlanabilmektedir.

Vertebral Column veri setinde kullanılan bagging algoritması için belirlenen parametre değerleri Çizelge 4.18'de verilmiştir.

Çizelge 4.18. Omurga veri seti parametreleri ve seviyeleri

Parametre	**n_estimator**	**max_samples**
Seviye	[5,10,20,30,50,100]	(1,2,3,10,20,50,75,100,155)

Vertebral Column veri seti için Pyhton programı kullanılarak sadece ön işleme yapılmış hali ile, K-ortalamalar kümeleme yöntemi ile ve melez kümeleme yöntemi ile sınıflandırma işlemleri sonucu elde edilen doğruluk oranları Çizelge 4.19'da verilmiştir.

Çizelge 4.19'da üç aşamanın sonuçlarının içerisinde sadece öznitelik seçimi ile elde edilen ve parametre optimizasyonu ile elde edilen doğruluk oranları verilmiştir.

Çizelge 4.19. Omurga veri seti için doğruluk oranları

Vertebral Column (Bagging algoritması)						
Yöntem	Sadece Ön İşleme		K-ortalamalar		Melez kümeleme	
Küme Sayısı	1		2		2	
İşlem	Özn. Seçimi	Parametre optimizasyonu	Özn. Seçimi	Parametre optimizasyonu	Özn. Seçimi	Parametre optimizasyonu
Tüm Öznitelik	83,55%	86,76%	87,59%	89,22%	87,15%	**89,22%**
5 Öznitelik	83,87%	85,67%	87,84%	88,39%	87,15%	88,67%
İyileştirme	0,38%	3,84%	5,13%	6,79%	4,31%	6,79%

Steel Plates Fault ve Vertebral Column veri setlerine uygulanan sadece ön işleme yapılmış hali ile, K-ortalamalar kümeleme yöntemi ile ve melez kümeleme yöntemi ile sınıflandırma işlemleri ve parametre optimizasyonları diğer veri setleri için de uygulanmıştır. Weka programı çalışma sonuçları, sadece ön işleme ile sınıflandırma işlemi için algoritma bazında en yüksek doğruluk oranları, Pyhton kesinlik, anma ve F-ölçüt sonuçları ve elde edilen doğruluk oranları sırasıyla;

- Cam (Glass) veri seti için EK 1'de,
- Tohumlar (Seeds) veri seti için EK 2'de,
- Kullanıcı Bilgisi (User Knowledge) veri seti için EK 3'te verilmiştir.

5. TARTIŞMA ve SONUÇ

Bu çalışmada, ilk olarak veri setleri için ön işleme yapılıp sonra kümeleme ve sınıflandırma yöntemleri uygulanarak sınıflandırma performans değerlendirme kriterlerinin iyileştirilmesi üzerine çalışılmıştır. Çalışma kapsamında, hem orta büyüklükte hem de daha büyük veri setlerinde önerilen yöntemlerin cevapları karşılaştırılmış ve üç farklı aşama halinde ilerlenip göstergeler veri setinin sadece ön işleme yapılmış halindeki göstergeler ile karşılaştırılmıştır. İlk aşamanın ilk kısmında sınıflandırma algoritmaları veri setinin sadece ön işleme yapılmış haline uygulanmıştır. İlk aşamanın ikinci kısmında sadece ön işleme yapılmış haline parametre optimizasyonu yapılmıştır. İkinci aşama olan K-ortalamalar kümeleme yöntemi ile sınıflandırma işleminin ilk kısmında her bir veri kümesinin kaç kümeye ayrılması gerektiğine karar verilmiştir. En yüksek kümeleme performansını sağlayan küme sayısı tespitinin ardından, her kümenin kendi içindeki sınıflandırma performansı gözlemlenmiştir. İkinci aşamanın ikinci kısmında da K-ortalamalar kümeleme yöntemi ile sınıflandırma işleminde parametre optimizasyonu yapılmıştır. Üçüncü aşamanın ilk kısmında ise kümeleme yöntemi olarak K-ortalamalar ve hiyerarşik kümeleme yöntemlerinin birbirini besleyecek melez kümeleme yöntemi kullanılmıştır. Melez (hibrid) kümeleme yöntemiyle oluşan her kümeye farklı sınıflandırma algoritmaları uygulanmıştır. Üçüncü aşamanın ikinci kısmında da melez kümeleme yöntemi ile sınıflandırma işleminde parametre optimizasyonu yapılmıştır. Bu işlemlerin sonucunda veri kümesini doğrudan sınıflandıran algoritmaların performans kriterleri ile önerilen yöntemler sonucundaki sınıflandırma performans kriterleri karşılaştırılmıştır.

Bu çalışmanın literatüre katkısı ise melez kümeleme yöntemi ile sınıflandırma algoritmalarının birlikte uygulanması olmuştur.

5.1. Çelik Levha Arızası Veri Seti

Çalışma kapsamında karşılaştırma yapılan Nkonyana ve diğerlerinin (2019) yaptığı çalışmada eğitim/test veri seti üzerine uygulanan Rassal Orman algoritma sonucu elde edilen doğruluk oranı %77,80, K-katlı çapraz doğrulama üzerine uygulanan Izgara arama parametre optimizasyonuyla elde edilen doğruluk oranı ise %75,50 olmuştur.

Pyhton programında eğitim/test veri seti üzerinde Rassal Orman uygulandığında ise elde edilen doğruluk oranları ise Çizelge 5.1'de verilmiştir. Çizelge 5.1 ile Nkonyana ve diğerlerinin (2019) yaptığı çalışma sonuçları karşılaştırıldığında sadece ön işleme yöntemi ile doğruluk oranı % 80,10'a artırılarak % 2, 96 oranında bir iyileştirme elde edilmiştir. K-katlı çapraz doğrulama üzerine yapılan çalışmaya göre ise doğruluk oranı melez kümeleme yöntemi ile sınıflandırma işlemine parametre optimizasyonu yapılarak %77,43'e artırılmıştır ve % 2,56 oranında iyileştirme elde edilmiştir.

Çizelge 5.1. Eğitim/test veri seti üzerinde Rassal Orman doğruluk oranı(%)

Öznitelikler	**Sadece Ön işleme ile RF Doğruluk Oranı(%)**
Tüm Öznitelik	78,39%
19 Öznitelik	78,39%
18 Öznitelik (-Xmax)	78,22%
18 Öznitelik (-Pixel Area)	**80,10%**
18 Öznitelik (-X Perimeter)	78,04%
18 Öznitelik (-A400)	79,25%

Eğitim/test veri seti üzerinde yapılan çalışmalarda elde edilen bu iyileştirmeler ön işleme aşamasında uygulanılan öznitelik seçimi ve korelasyon matrisinin başarısını göstermektedir.

Nkonyana ve diğerlerinin (2019) yaptığı çalışmaya göre iyileştirme elde edildikten sonra sadece ön işleme ile sınıflandırma işlemi aşamasında K-katlı çapraz doğrulama ile Rassal Orman algoritması kullanılarak gerçekleştirilen uygulama sonucu kesinlik, anma ve F-ölçüt performans göstergeleri için % 2,56 oranında bir iyileşme sağlandığı görülmektedir. Sınıflandırma işlemi için gerçekleştirilen üç aşama sonunda elde edilen doğruluk oranları ise Çizelge 5.2'de verilmiştir.

Çizelge 5.2. K-katlı çapraz doğrulama ile Rassal Orman doğruluk oranları(%)

Yöntem	Ham Hali		K-ortalamalar		Melez Kümeleme			
Küme Sayısı	1		4		3		4	
İşlem	Feature selection	Parametre opt.	Feature selection	Parametre opt.	Feature selection	Parametre opt.	Feature selection	Parametre opt.
Tüm Öznitelik	65,48%	68,41%	68,14%	74,97%	71,12%	74,66%	70,55%	76,14%
19 Öznitelik	66,36%	68,58%	67,95%	75,09%	70,75%	74,43%	71,50%	77,32%
18 Öznitelik (-Xmax)	66,98%	68,76%	69,74%	75,52%	71,37%	75,50%	72,43%	**77,43%**
18 Öznitelik (-Pixel Area)	65,27%	68,35%	69,15%	75,50%	71,23%	74,98%	72,01%	76,46%
18 Öznitelik (-X Perimeter)	65,69%	68,70%	69,86%	74,96%	70,99%	75,29%	71,39%	76,35%
18 Öznitelik (-A400)	65,73%	68,59%	68,12%	75,14%	71,12%	75,24%	71,49%	76,24%
İyileştirme (En iyiler)	2,29%	5,01%	6,69%	15,33%	9,00%	15,30%	10,61%	18,25%

Çizelge 5.2 incelendiğinde tespit edilen sonuçlar aşağıdaki gibidir.

- Eğitim/test veri seti üzerinde uygulanan çalışmada olduğu gibi K-katlı çapraz doğrulama ile çalışma yapıldığında da öznitelik seçimi ve korelasyon matrisi işlemlerinin doğruluk oranında iyileştirmeye sebep olduğu görülmektedir.
- Uygun parametre aralıkları girilip parametre optimizasyonu yapıldığında her üç yöntemde de daha yüksek doğruluk oranları elde edilmiştir.
- K-ortalamalar kümeleme yöntemi ile yapılan sınıflandırma işlemleri sonuçlarının sadece ön işleme yapılmış veriler ile yapılan sınıflandırma işlemleri sonuçlarından daha iyi olduğu görülmektedir.
- Sadece ön işleme yapılmış veriler ile, K-ortalamalar kümeleme yöntemi ile ve önerilen melez kümeleme yöntemi ile sınıflandırma işlemleri doğruluk oranları kıyaslandığında en iyi sonuçların önerilen melez kümeleme yöntemi ile sınıflandırma işlemleri sonucunda elde edildiği tespit edilmektedir. Bu da önerilen yöntemin başarısını göstermektedir.

5.2. Tüm Veri Setleri

Tüm veri setleri için yapılan sadece ön işleme yapılmış veriler ile sınıflandırma işlemi aşamasında elde edilen kesinlik, anma ve F-ölçüt sonuçları karşılaştırıldığında her veri seti için iyileştirme oranları Çizelge 5.3'te verilmiştir.

Çizelge 5.3. Veri setleri için sınıflandırma performans göstergeleri iyileştirme oranları

Veri Seti	Kesinlik İyileştirme(%)	Anma İyileştirme (%)	F-ölçüt İyileştirme(%)
Steel Plates Fault	%2,56	%2,56	%2,56
Glass	%0	%0	%0
Vertebral Column	%1,20	%0	%1,20
Seeds	%2,20	%2,20	%2,20
User Knowledge	%3,23	%3,23	%3,23

Glass veri seti için herhangi bir iyileştirme elde edilememiş olma sebebi en yüksek oranları tüm öznitelikler için uygulandığında almış olmasıdır. Genel değerlendirme yapıldığında ise sadece ön işleme yapılmış veriler ile yapılan sınıflandırma işlemlerinde korelasyon matrisi ve öznitelik seçimi ile kesinlik, anma ve F-ölçüt performans göstergelerinde iyileştirme yapılmıştır.

Çalışma yapılan veri setleri için uygulanan üç aşama sonucunda elde edilen en yüksek doğruluk oranı, uygulanılan algoritma, öznitelik sayısı, ilk hallerine göre iyileştirme oranları ve elde edildiği yöntem Çizelge 5.4'te verilmiştir.

Çizelge 5.4. Veri setleri için sonuçlar

Veri Seti	**Algoritma**	**Doğruluk Oranı(%)**	**Öznitelik Sayısı**	**İyileştirme Oranı(%)**	**Yöntem**
Steel Plates Fault	Rassal Orman	% 77,43	18	% 18,25	Melez + Param. opt.
Glass	Rassal Orman	% 87,48	10	% 14,85	Melez + Param. opt.
Vertebral Column	Bagging	% 89,22	7	% 6,79	Melez + Param. opt.
Seeds	Bagging	% 97,10	8	% 6,20	Melez + Param. opt.
User Knowledge	Rassal Orman	% 97,24	2	% 4,50	Sadece ön işleme + Param. opt.

Çalışma kapsamında uygulama yapılan 5 adet veri seti içerisinde 4 veri setinin melez kümeleme yöntemi ile sınıflandırma işlemi uygulanarak daha yüksek doğruluk oranlarına ulaşılmıştır. Bu da önerilen yöntemin başarısını göstermektedir. Ayrıca parametre optimizasyonu uygulanması sayesinde 5 veri seti için de daha yüksek doğruluk oranı elde edildiği görülmektedir. Uygulama yapılan veri setleri için, önerilen kümeleme yönteminde belirlenen küme sayıları ile sınıf etiketi sayısının çoğunlukla aynı olmadığı tespit edilmiştir.

Yapılan çalışma sonucunda sınıflandırma algoritmaları performans göstergelerini iyileştirebilmek için melez kümeleme ile sınıflandırma işleminin gerçekleştirilmesinin alternatif olarak düşünülebileceği ortaya konulmuştur.

KAYNAKLAR

Aarshay, J. (2018). Complete guide to parameter tuning in gradient boosting (GBM) in python [Blog yazısı].
Erişim adresi: https://www.analyticsvidhya.com/blog/2016/02/complete-guide-parameter-tuning-gradient-boosting-gbm-python/

Abbas, O. A. (2008). Comparisons between data clustering algorithms. *The international Arab journal of information technology,* 5, 3.

Akbulut, S. (2006). Veri madenciliği teknikleri ile bir kozmetik markanın ayrılan müşteri analizi ve müşteri segmentasyonu. *Gazi Üniversitesi Fen Bilimleri Enstitüsü, Ankara.*

Altun, M. (2017). *Veri madenciliği ve uygulama alanları.* Doktora Semineri Raporu, Akdeniz Üniversitesi, Eğitim Bilimleri.

Andrade, R., Faria, W.M., Silva, S., Chakraborty, S. ve Curi, N. (2020). Prediction of soil fertility via portable X-ray fluorescence (pXRF) spectrometry and soil texture in the Brazilian coastal plains. *Geoderma,* 357.

Aydın, N. ve Seven, A.N. (2015). İl nüfus ve vatandaşlık müdürlüklerinin iş yoğunluğuna göre hibrid kümeleme ile sınıflandırılması. *Yönetim ve Ekonomi Araştırmaları Dergisi,* 13(2), 181-201.

Bahrudin, H. S., Alam ve Haiyunnisa, T. (2016). Computational fluid dynamic simulation of pipeline irrigation system based on ansys. *International Journal of Technology and Engineering Studies,* 2, 6, 189-193.

Bergstra, J. ve Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research,* 13, 1, 281-305.

Bergstra, J., Bardenet, R., Bengio, Y. ve Kégl, B. (2011). Algorithms for hyper-parameter optimization. *in Proc. of the 24th Intl. Conf. on Neural Information Processing Systems,* 2546-2554.

Bhadauria, R. (2021). Boosting in Machine Learning | Boosting and AdaBoost [Blog yazısı]. Erişim adresi: geeksforgeeks.org

Bozan, F. (2010). CART(Classification and Regression Tree) [Blog yazısı]. Erişim adresi: http://www.farukbozan.com/2010/01/cartclassification-and-regression-tree

Breiman, L. (1996a). *Bagging predictors.* Machine Learning, 24 (2), 123–140 .

Breiman, L. (1996b). Heuristics of instability and stabilization in model selection. *The Annals of statistics,* 24 (6), 2350–2383 .

Breiman, L. (2001*). Random forests.* Mach Learn 45:5–32.

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

Buciu, I. (2006). Demonstrating the stability of support vector machines for classification. *Signal Processing,* 86: 2364-2380.

Burn, D. H., Zrinji, Z. ve Kowalchulk, M. (1997). Regionalization of catchments for regional flood frequency analysis. *Journal of Hydrologic Engineering*, 2(2), 76–82.

Ceylan, Z., Gürsev, S. ve Bulkan, S. (2017). İki aşamalı kümeleme analizi ile bireysel emeklilik sektöründe müşteri profilinin değerlendirilmesi. *Bilişim Teknolojileri Dergisi,* 10(4), 475-485.

Chen, B., Tai, P.C., Harrison, R ve Pan, Y. (2005). Novel hybrid hierarchical K-means clustering method (HK-means) for microarray analysis. *IEEE Computational Systems Bioinformatics Conference,* California-USA.

Cheung, Y.M. (2003). k∗-Means: A new generalized k-means clustering algorithm. *Pattern Recognition Letters*, 24, 15, 2883-2893.

Cover, T. ve Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Trans. Inform. Theory* 13 (1) 21–27.

Çalışkan, S. ve Soğukpınar, İ. (2008). K-KNN: K-Means ve K en yakın komşu yöntemleri ile ağlarda nüfuz tespiti. *TmmobEmo 2.Ağ Ve Bilgi Güvenliği Ulusal Sempozyumu.*

Dash, M. ve Liu, H. (1997). Feature selection for classification. *Intelligent data analysis,* 1(1-4):131–156.

Dursun, F. (2018). Temel istatistik işlemleri [Blog yazısı]. Erişim adresi: https://fatihdursunn.wordpress.com.

Elavaras, S. A. (2011). A survey on partitional clustering algorithm. *International Journal of Enterprise Computing and Business Systems,* 1, 1.

Fayyad, U., Piatetsky-Shapiro, G. ve Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI Magazine,* 17, 37-54

Ferreira, L. ve Hitchcock, D. B. (2009). A comparison of hierarchical methods for cluster functional data. *Communications in Statistics-Simulation and Computation,* 38(9), 1925-1949.

Freund, Y. ve Shapire, R. (1995). A decision-theoretic generalization of on-line learning and application to boosting. *Proceedings of the Second European Conference on Computaional Learning Theory,* 23-27.

García, S., Luengo, J. ve Herrera, F. (2015). Tutorial on practical tips of the most influential data preprocessing algorithms in data mining. *Knowledge-Based Systems.*

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. ve Witten, I.H. (2009). The weka data mining software: an update. *SIGKDD Explor Newsl*, 11(1):10–18.

Haltaş, A. ve Alkan, A. (2016). Medline veritabanı üzerinde bulunan tıbbi dokümanların kanser türlerine göre otomatik sınıflandırılması. *Bilişim Teknolojileri Dergisi,* 9-2.

Hands, S. ve Everitt, B. (1987). A Monte Carlo study of the recovery of cluster structure in binary data by hierarchical cluster techniques. *Multivar.Behav. Res.* 22, 235-243.

Hastie, T. , Tibshirani, R., ve Friedman, J. (2001). *The Elements of statistical learning*. Springer.

Hitziger, M. ve Lies, M. (2014). Comparison of three supervised learning methods for digital soil mapping: application to a complex terrain in the ecuadorian andes. *Appl. Environ. Soil Sci.*

Hossin, M. ve Sulaiman, M.N. (2015). A Review on evaluation metrics for data classification evaluations. *International Journal of Data Mining & Knowledge Management Process (IJDKP),* 5, 2.

Huang, J. ve Ling, C. X. (2007). Constructing new and better evaluation measures for machine learning. *in R. Sangal, H. Mehta and R. K. Bagga (Eds.) Proc. of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007),* San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 859-864.

James, B. ve Yoshua, B. (2012). Random search for hyper-parameter optimization. *J. Mach. Learn. Res.* 13 (1) 281–305.

Jeevani, W. (2001). Performance degradation in boosting. *In conf. MCS :multiple classifier systems,11*-21.

Kijsipongse, E. ve U-ruekolan, S. (2012). Dynamic load balancing on GPU clusters for large-scale K-Means clustering. *IEEE International Joint Conference on Computer Science and Software Engineering (JCSSE),* 346, 350.

Kim, H. J., Jo, N. O. ve Shin, K. S. (2016). Optimization of cluster-based evolutionary undersampling for the artificial neural networks in corporate bankruptcy prediction. *Expert Systems with Applications,* 59, 226-234.

Kim, S. ve Pang, M.Je. (2005). Constructing support vector machine ensemble. *Pattern Recognition*, 36:2757-2767

Kurgan, L.A. ve Musilek, P. (2006). A survey of knowledge discovery and data mining process models. *Knowl. Eng. Rev.*, 21 (01), 1.

Lavesson, N. ve Davidsson, P. (2008). Generic methods for multi-criteria evaluation. *in Proc. of the Siam Int. Conference on Data Mining,* Atlanta, Georgia, USA: SIAM Press, 541-546.

Li, B. , Chen, Y. ve Chen, Y. (2008). The nearest neighbor algorithm of local probability centers. *IEEE Trans. Syst. Man, Cybern. B, Cybern.* 38 (1) 141–154.

Li, X., Wang, L. ve Sung, E. (2005). A study of adaboost with SVM based weak learners. *Proceedings of International Joint Conference on Neural Network.*

Lin, G-F. ve Chen, L-H. (2006). Identification of homogeneous regions for regional frequency analysis using the self-organizing map. *Journal of Hydrology,* 324, 1-9.

Liu, H. ve Motoda, H. (2007). *Computational methods of feature selection.* Chapman and Hall/CRC Press.

Liu, J., Sun, S., Tan, Z. ve Liu, Y. (2020). Non-destructive detection of sunset yellow in cream based on near-infrared spectroscopy and interval random forest. *Spectrochim. Acta Part A Mol. Biomol. Spectrosc.* 242.

Murtagh, F. ve Legendre, P. (2014). Ward's hierarchical agglomerative cluster method: Which algorithms implement ward's criterion?. *Journal of Classification,* 31(3), 274 295.

Nkonyana, T., Sun, Y., Twala, B. ve Dogo, E. (2019). Performance evaluation of data mining techniques in steel manufacturing industry. *2nd International Conference on Sustainable Materials Processing and Manufacturing,* 35, 623-628.

Onan, A. (2018). Firma başarısızlığının tahmin edilmesi için kümelemeye dayalı bir sınıflandırıcı topluluğu yaklaşımı. *The Journal of Operations Research, Statistics, Econometrics and Management Information Systems,* 6, 2.

Öğüt, M. (2005). *Örneklere dayalı bir sınıflandırma algoritma tasarımı ve uygulaması.* Yüksek Lisans Tezi, Selçuk Üniversitesi, Fen Bilimleri Enstitüsü.

Papadopoulos, S., Azar, E., Woon. W.L. ve Kontokosta CE. (2018). Evaluation of tree-based ensemble learning algorithms for building energy performance estimation. *J Build Perform Simul 11*:322–32.

Pham, D. T., Dimov, S. S. ve Nguyen, C. D. (2005). Selection of K in K-means clustering. *Proc. IMechE 219 Part C: J. Mechanical Engineering Science, IMechE.*
Pyle, D. (1999). *Data preparation for data mining.* Morgan Kaufmann Publishers Inc.

Quinlan, J. R. (1987). Simplifying decision trees. *International Journal of Man-Machine Studies,* 27, 3, 221-234.

Ragothaman, B. ve Sarojini, B. (2016). A multi-objective non-dominated sorted artificial bee colony feature selection algorithm for medical datasets. *Indian J. Sci. Technol.,* 9 (45).

Rani, Y. ve Rohil, H. (2013). A study of hierarchical clustering algorithm. *International Journal of Information and Computation Technology.* ISSN 0974-2239, 3, 11, 1225-123

Ronowicz, J., Thommes, M., Kleinebudde, P., Krysi´nski, J. (2015). A data mining approach to optimize pellets manufacturing process based on a decision tree algorithm. *Eur. J. Pharm. Sci.* 73, 44–48.

Sanner, M. F. (1999). Python: a programming language for soware integration and development. *The Scripps Research Institute.*

Sasidharan, A. (2021). Support Vector Machine Algorithm [Blog yazısı]. Erişim adresi: geeksforgeeks.org

Savaş, S., Topaloğlu, N. ve Yılmaz, M. (2012). Veri madenciliği ve Türkiye'deki uygulama örnekleri. *İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi,* 11, 21.

Seifert, J.W. (2004). Data mining and the search for security: challenges for connecting the dots and databases. *Government Information Quarterly,* 21 (4), 461 - 480.

Sharma, S., Agrawal, J., Agarwal, S. ve Sharma, S. (2013). Machine learning techniques for data mining: a survey. *In: 2013 IEEE International Conference on Computational Intelligence and Computing Research, IEEE ICCIC 2013. IEEE Computer Society.6*

Shukla, S. ve Naganna, S. (2014). A review on K-means data clustering approach. *International Journal of Information & Computation Technology,* 4, 17, 1847-1860.

Soni, N. ve Ganatra, A. (2012). Comparitive study of several clustering algorithms. *International Journal of Advanced Computer Research,* 2, 4, 6.

Şchiopu, D. (2010). Applying TwoStep cluster analysis for identifying bank customers' profile. *Buletinul,* 62, 66-75.

Şener, Y. (2017). Destek Vektör Makineleri (Support Vector Machine | SVM) Çalışma Mantığı ve Python Uygulaması [Blog yazısı]. Erişim adresi: https://yigitsener.medium.com/

Tan, P.N., Steinbach, M. ve Kumar, V. (2016). *Introduction to data mining.* Pearson Education Hindistan.

UCI Machine Learning Repository. Datasets [Veri seti]. Erişim adresi: https://archive.ics.uci.edu/ml/datasets.php

Usama, F. , Gregory, P. ve Padhraic, S. (1996). *Knowledge discovery and data mining.* AAAI Press / MIT Press, Cambridge.

Velayutham, C. ve Thangavel, K. (2011). Unsupervised quick reduct algorithm using rough set theory. *Journal of Electronic Science and Technology*, 9, 3.

Voorhees, E.M. (1986). Implementing agglomerative hierarchical clustering algorithms for use in document retrieval. *Information Processing and Management*, 22(6):465–476.

Wei, P., Lu, Z. ve Song, J. (2015). Variable importance analysis: A comprehensive review. *Reliab Eng Syst Saf*, 142:399–432.

Wikipedia (2013). Erişim adresi: http://en.wikipedia.org/wiki

Wu, C., Peng, Q., Lee, J., Leibnitz, K. ve Xia, Y. (2021). Effective hierarchical clustering based on structural similarities in nearest neighbor graphs. *Knowledge-Based Systems*, 228.

Wu, X., Kumar, V., Quinlan, J.R., Ghosh, J., Yang, Q. ve Motoda, H. (2008). Top 10 algorithms in data mining. *Knowl. Inform. Syst.* 14 (1) 1–37.

Yiwei, P., Zhibin, P., Yikun, W. ve Wei, W. (2020). A new fast search algorithm for exact k-nearest neighbors based on optimal triangleinequality- based check strategy. *Knowl. Based Syst.*

Zaimoğlu, E. A. (2018). *Veri madenciliği teknikleri kullanılarak sosyal ağlar aracılığı ile bilgisayar ve bilişim mühendisliği mezun öğrenci profillerinin belirlenmesi.* Yüksek Lisans Tezi, Sakarya Üniversitesi, Fen Bilimleri Enstitüsü.

Zhang, Y. ve Haghani, A. (2015). A gradient boosting method to improve travel time prediction. *Transp Res Part C Emerg Technol* 58:308–24.

EKLER

EK 1 Cam Veri Seti İçin Sonuçlar
EK 2 Tohumlar Veri Seti İçin Sonuçlar
EK 3 Kullanıcı Bilgisi Veri Seti İçin Sonuçlar

EK 1 Cam Veri Seti İçin Sonuçlar

Weka çalışması performans sonuçları

Glass					
	Cross validation Fold				
	Ham hali	**K-means ort.**	**Hiyerarşik ort.**	**K-means iyileştirmesi**	**Hiyerarşik iyileştirmesi**
Random Forest	46,73%	48,20%	50,34%	3,15%	7,73%
Adaboost	41,12%	50,61%	53,60%	23,07%	30,34%
Bagging	42,52%	45,74%	41,75%	7,56%	-1,82%
SVM	52,33%	70,18%	59,71%	34,11%	14,09%
KNN	55,61%	73,68%	61,08%	32,49%	9,84%
Ortalama iyileştirme				**20,08%**	**12,03%**

EK 1 Devamı

Sadece ön işleme ile sınıflandırma işlemi için algoritma bazında en yüksek doğruluk oranları

Algoritma	**En Yüksek Doğruluk Oranı (%)**	**En iyi Öznitelik Sayısı**
Rassal Orman	**%76,17**	9
KNN	%64,95	6
Bagging	%70,56	4
Adaboost	%51,87	4
CART	%63,55	9
SVM	%58,41	3

Sadece ön işleme ile sınıflandırma işlemi için diğer performans kriterleri sonuçları

Öznitelik	**Kesinlik(%)**	**Anma(%)**	**F-ölçüt (%)**
Tüm öznitelikler	%76	%76	%76

EK 1 Devamı

Pyhton doğruluk oranları

Glass (Rassal orman algoritması)								
Yöntem	Sadece Ön İşleme		K-ortalamalar		Melez Kümeleme			
Küme Sayısı	1		3		3		4	
İşlem	Özn. Seçimi	Param. opt.	Özn. Seçimi	Param. opt.	Özn. Seçimi	Param. opt.	Özn. Seçimi	Param. opt.
Tüm Öznitelik	76,17%	79,95%	78,96%	86,86%	80,96%	**87,48%**	77,27%	80,82%
İyileştirme	0,00%	4,96%	3,66%	14,03%	6,29%	14,85%	1,44%	6,10%

EK 2 Tohumlar Veri Seti İçin Sonuçlar

Weka çalışması performans sonuçları

Seeds					
	Cross validation Fold				
	Ham hali	**K-means ort.**	**Hiyerarşik ort.**	**K-means iyileştirmesi**	**Hiyerarşik iyileştirmesi**
Random Forest	62,38%	61,46%	59,35%	-1,47%	-4,86%
Adaboost	33,33%	34,15%	35,48%	2,46%	6,45%
Bagging	48,57%	48,78%	50,34%	0,43%	3,63%
SVM	67,62%	69,76%	62,01%	3,16%	-8,30%
KNN	39,05%	41,95%	39,73%	7,43%	1,74%
Ortalama iyileştirme				**2,40%**	**-0,27%**

EK 2 Devamı

Sadece ön işleme ile sınıflandırma işlemi için algoritma bazında en yüksek doğruluk oranları

Algoritma	**En Yüksek Doğruluk Oranı (%)**	**En iyi Öznitelik Sayısı**
Rassal Orman	%92,38	7
KNN	%86,67	6
Bagging	**%92,86**	6
Adaboost	%86,19	3
CART	%90,48	3
SVM	%90	7

Sadece ön işleme ile sınıflandırma işlemi için diğer performans kriterleri sonuçları

Öznitelik	**Kesinlik(%)**	**Anma(%)**	**F-ölçüt (%)**
Tüm öznitelikler	%91	%91	%91
6 öznitelik	%93	%93	%93

EK 2 Devamı

Pyhton doğruluk oranları

Seeds (Bagging algoritması)								
Yöntem	Sadece Ön İşleme		K-means		Hibrid Yöntem			
Küme Sayısı	1		3		4		3	
İşlem	Özn. Seçimi	Parametre optimizasyonu	Özn. Seçimi	Parametre optimizasyonu	Özn. Seçimi	Parametre optimizasyonu	Özn. Seçimi	Parametre optimizasyonu
Tüm Öznitelik	91,43%	94,18%	93,68%	96,11%	95,80%	96,71%	93,80%	**97,10%**
6 Öznitelik	92,86%	93,07%	94,51%	95,64%	95,80%	95,46%	93,80%	96,15%
5 Öznitelik (-a)	92,38%	93,63%	94,51%	96,11%	93,38%	94,83%	94,62%	96,21%
5 Öznitelik (-b)	92,38%	93,07%	93,65%	96,59%	94,69%	95,67%	93,80%	96,57%
5 Öznitelik (-d)	91,43%	92,54%	94,51%	97,06%	94,69%	96,71%	93,31%	96,63%
5 Öznitelik (-e)	91,90%	94,68%	93,65%	96,11%	95,35%	96,92%	93,80%	96,15%
5 Öznitelik (-g)	85,24%	89,82%	88,12%	93,18%	90,65%	94,46%	89,29%	93,31%
İyileştirme	1,56%	3,55%	3,37%	6,16%	4,78%	6,00%	3,49%	6,20%

EK 3 Kullanıcı Bilgisi Veri Seti İçin Sonuçlar

Weka çalışması performans sonuçları

User Knowledge					
	Cross validation Fold				
	Ham hali	**K-means ort.**	**Hiyerarşik ort.**	**K-means iyileştirmesi**	**Hiyerarşik iyileştirmesi**
Random Forest	83,38%	69,78%	77,40%	-16,32%	-7,18%
Adaboost	33,50%	45,32%	36,89%	35,28%	10,10%
Bagging	80,40%	68,55%	75,99%	-14,75%	-5,49%
SVM	73,70%	65,27%	66,78%	-11,45%	-9,39%
KNN	72,46%	66,21%	64,62%	-8,63%	-10,83%
			Ortalama iyileştirme	**-3,17%**	**-4,56%**

EK 3 Devamı

Sadece ön işleme ile sınıflandırma işlemi için algoritma bazında en yüksek doğruluk oranları

Algoritma	**En Yüksek Doğruluk Oranı (%)**	**En iyi Öznitelik Sayısı**
Rassal Orman	**%96,03**	2
KNN	%96,03	2
Bagging	%95,29	2
Adaboost	%69,98	1
CART	%93,55	2
SVM	%95,53	2

Sadece ön işleme ile sınıflandırma işlemi için diğer performans kriterleri sonuçları

Öznitelik	**Kesinlik(%)**	**Anma(%)**	**F-ölçüt (%)**
Tüm öznitelikler	%93	%93	%93
2 öznitelik	%96	%96	%96

EK 3 Devamı

Pyhton doğruluk oranları

User Knowledge (Rassal orman algoritması)						
Yöntem	Sadece Ön İşleme		K-means		Hibrid Yöntem	
Küme Sayısı	1		5		5	
İşlem	Özn. Seçimi	Parametre optimizasyonu	Özn. Seçimi	Parametre optimizasyonu	Özn. Seçimi	Parametre optimizasyonu
Tüm Öznitelik	93,05%	94,23%	93,31%	95,16%	93,32%	94,83%
2 Öznitelik	96,03%	**97,24%**	94,04%	96,30%	94,46%	95,52%
İyileştirme	3,20%	4,50%	1,06%	3,49%	1,52%	2,65%

ÖZGEÇMİŞ

Adı Soyadı : Elif GÜLERYÜZ
Doğum Yeri ve Tarihi :
Yabancı Dil : İngilizce, Almanca

Eğitim Durumu
Lise : Bursa Ulubatlı Hasan Anadolu Lisesi
Lisans : Uludağ Üniversitesi Endüstri Mühendisliği
Yüksek Lisans : Uludağ Üniversitesi Endüstri Mühendisliği

Çalıştığı Kurumlar : Faurecia (06.2021 - Halen)
Yeşim Tekstil (01.2019 - 08.2019)
TOFAŞ (10.2017 - 06.2018)
Çimtaş (10.2017 - 06.2018)

İletişim (e-posta) : eliffguleryuzz@gmail.com

Akademik çalışmalar :

Güleryüz, E., Yıldız, P., Yılmaz Eroğlu D., Gündüz, T. (2018). *Capacity planning for support parts for pipe production.* International Conference on Applied Mathematics in Engineering (ICAME)