T.C.
KIRIKKALE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

VERİ KÜMESİNDEKİ DOĞAL YAPILANMALAR İLE MAKİNE ÖĞRENMESİ

BERGEN KARABULUT

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

DOKTORA TEZİ

DANIŞMAN
Doç. Dr. Halil Murat ÜNVER

KIRIKKALE-2022

Bergen KARABULUT tarafından hazırlanan "VERİ KÜMESİNDEKİ DOĞAL YAPILANMALAR İLE MAKİNE ÖĞRENMESİ" adlı tez çalışması aşağıdaki jüri tarafından OY BİRLİĞİ ile Kırıkkale Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalında DOKTORA TEZİ olarak kabul edilmiştir.

Danışman: Doç. Dr. Halil Murat ÜNVER

Bilgisayar Donanımı Anabilim Dalı, Kırıkkale Üniversitesi İmza……..

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

İkinci Danışman: Prof. Dr. Güvenç ARSLAN

Uygulamalı İstatistik Anabilim Dalı, Kırıkkale Üniversitesi İmza……..

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

Başkan: Prof. Dr. Necaattin BARIŞÇI

Bilgisayar Mühendisliği Anabilim Dalı, Gazi Üniversitesi İmza……..

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

Üye: Doç. Dr. Ahmet ARSLAN

Bilgisayar Donanımı Anabilim Dalı, Eskişehir Teknik Üniversitesi İmza……..

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

Üye: Doç. Dr. Bülent Gürsel EMİROĞLU

Bilgisayar Yazılımı Anabilim Dalı, Kırıkkale Üniversitesi İmza……..

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

Üye: Dr. Öğr. Üyesi Enes AYAN

Bilgisayar Bilimleri Anabilim Dalı, Kırıkkale Üniversitesi İmza……..

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

Tez Savunma Tarihi: 05/01/2022

Jüri tarafından kabul edilen bu tezin Doktora Tezi olması için gerekli şartları yerine getirdiğini onaylıyorum.

……..…….

Prof. Dr. Recep ÇALIN

Fen Bilimleri Enstitüsü Müdürü

Bu tezi sevgili anne ve babama ithaf ediyorum.

ETİK BEYANI

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

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

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

……

Bergen KARABULUT

05/01/2022

ÖZET

VERİ KÜMESİNDEKİ DOĞAL YAPILANMALAR İLE MAKİNE ÖĞRENMESİ

Kırıkkale Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı, Doktora Tezi
Danışman: Doç. Dr. Halil Murat ÜNVER
Ortak Danışman: Prof. Dr. Güvenç ARSLAN
Ocak 2022, 104 sayfa

Makine öğrenmesinde yaygın kullanılan öğrenme türlerinden birisi denetimli öğrenmedir. Teknolojik gelişmeler ve veri boyutlarındaki hızlı artışla birlikte mevcut denetimli öğrenme yöntemlerinin daha etkin hale getirilmesi ve yeni yöntemlerin geliştirilmesi yönündeki çalışmalar önem kazanmıştır. Bu doğrultuda, veri kümesinden daha etkin yararlanmayı amaçlayan çalışmalar dikkat çekmektedir. Bu çalışmaların bazılarında, kümeleme gibi denetimsiz öğrenme yöntemleriyle elde edilen doğal yapılanmaların, denetimli öğrenme sürecinde kullanımının araştırıldığı görülmektedir. Mevcut çalışmalar, veri kümesindeki doğal yapılanmaların tespit edilmesi ve bu yapılanmaların denetimli öğrenme sürecinde kullanımının etkin sonuçlar sağlayabildiğini göstermekte ve yeni çalışmaların gerekliliğini ortaya koymaktadır.

Bu çalışmada, veri kümesindeki yapısal bilginin (yani doğal yapılanmaların) elde edilmesi ve bu bilginin denetimli öğrenme sürecinde kullanılması için iki farklı yaklaşım araştırılmıştır. İlk olarak Benzerlik Tabanlı Doğal Kümeler (SNC) olarak adlandırılan yeni bir kümeleme algoritması önerilmiştir. SNC kümeleme algoritması ile elde edilen yapısal kümeler, sınıflandırma sürecine adapte edilerek yeni algoritmalar araştırılmıştır. Bu şekilde, Doğal Kümeler Tabanlı En Benzer Örnekler (NC-MSI), Doğal Kümeler Tabanlı Destek Vektör Makinesi (NC-SVM) ve Doğal Kümeler Tabanlı Destek Vektör Makinesi-Sınırlar (NC-SVM-B) sınıflandırma algoritmaları önerilmiştir. Bu sınıflandırma algoritmalarının her birinde yapısal kümeler farklı şekilde kullanılmıştır. Önerilen algoritmalar, literatürde yer alan benzer yöntemlerle çeşitli gerçek hayat veri kümeleri üzerinde karşılaştırmalı olarak analiz edilmiştir. Elde edilen sonuçlar, önerilen algoritmaların, özellikle bazı veri kümelerinde önemli örnekleri yani yapısal bilgiyi başarılı şekilde tespit edebildiğini ve veri kümesinden daha etkin yararlanabildiğini göstermektedir.

İkinci olarak, veri kümesinden doğal yapılanmaların elde edilmesi için CURE kümeleme algoritması kullanılmıştır. CURE algoritması ile veri kümesinin yapısal bilgisini elde eden ve bu bilgiyi eğitim kümesi yerine denetimli öğrenme sürecinde kullanan Temsili Noktalar Tabanlı Destek Vektör Makinesi (RP-SVM) algoritması

önerilmiştir. Bu algoritmada, SVM yöntemi tüm eğitim kümesi yerine daha az örnek içeren temsili noktalar kümesi ile eğitilmektedir. RP-SVM yöntemi, çeşitli gerçek hayat veri kümeleri üzerinde standart SVM, KMSVM, *K*NN ve CART yöntemleri ile karşılaştırmalı olarak analiz edilmiştir. Elde edilen sonuçlar, RP-SVM yönteminin standart SVM yöntemi ile benzer doğruluk elde ederken eğitim kümesi boyutunu önemli ölçüde azalttığını ve daha az destek vektörü kullanılmasını sağlayabildiğini göstermektedir. Ayrıca RP-SVM yöntemi, *K*NN ve CART yöntemlerine kıyasla daha az eğitim örneği kullanarak daha iyi doğruluk elde edebilmektedir. Bununla birlikte, RP-SVM yöntemi KMSVM yöntemine göre daha az veri azaltma sağlamakta ancak RP-SVM yönteminin tüm veri kümelerinde doğruluk açısından iyi sonuçlar elde ederek KMSVM yönteminden daha kararlı olduğu görülmektedir.

Bu çalışma kapsamında elde edilen sonuçlar, veri kümesinden elde edilen doğal yapılanmaların denetimli öğrenme sürecine katkı sağlayabileceğini göstermektedir. Önerilen yöntemler, geliştirilebilir ve farklı makine öğrenmesi yöntemlerine adapte edilebilir niteliktedir. Ayrıca önerilen yaklaşımlar, büyük veri çalışmaları için motivasyon sağlayabilir.

Anahtar kelimeler: CURE algoritması, kümeleme, doğal yapılanmalar, temsili noktalar, destek vektör makineleri, yapısal bilgi, sınıflandırma.

ABSTRACT

MACHINE LEARNING WITH NATURAL STRUCTURES IN THE DATA SET

Kırıkkale University

Graduate School of Natural and Applied Sciences

Department of Computer Engineering, Ph. D. Thesis

Supervisor: Assoc. Prof. Dr. Halil Murat ÜNVER

Co-Supervisor: Prof. Dr. Güvenç ARSLAN

January 2022, 104 pages

One of the commonly used learning types in machine learning is supervised learning. With the technological developments and the rapid increase in data sizes, studies aimed at making existing supervised learning methods more effective and at developing new methods have gained importance. In this direction, studies aiming to make more effective use of the data set draw attention. In some of these studies, it is seen that the use of natural structures obtained by unsupervised learning methods, such as clustering, in the supervised learning process has been investigated. Existing studies show that detecting the natural structures in the data set and using these structures in the supervised learning can provide effective results, and reveal the necessity of new studies.

In this study, two different approaches have been investigated in order to obtain the structural information (that is, the natural structures) in the data set and to use this information in the supervised learning process. Firstly, a new clustering algorithm called Similarity-based Natural Clusters (SNC) was proposed. Structural clusters obtained with the SNC algorithm were adapted to the classification and new algorithms were investigated. In this way, the Natural Clusters-based Most Similar Instances (NC-MSI), Natural Clusters-based Support Vector Machine (NC-SVM) and Natural Clusters-based Support Vector Machine-Boundaries (NC-SVM-B) classification algorithms were proposed. Structural clusters were used differently in each of these algorithms. The proposed algorithms were analyzed comparatively on various real-life data sets with similar methods in the literature. The results show that the proposed algorithms can successfully detect the important instances, i.e., the structural information, especially in some data sets, and can utilize the data set more effectively.

Secondly, the CURE clustering algorithm was used to obtain the natural structures from the data set. A Representative Points-based Support Vector Machine (RP-SVM) algorithm was proposed, which obtains the structural information of the data set with the CURE algorithm and uses this information in the supervised learning instead of the training set. In this algorithm, the SVM method is trained with a set of

representative points containing fewer samples instead of the entire training set. The RP-SVM method was analyzed comparatively with the standard SVM, KMSVM, *K*NN and CART methods on various real-life data sets. The results show that the RP-SVM method can achieve similar accuracy to the standard SVM method, while significantly reducing the training size and using fewer support vectors. In addition, the RP-SVM method can obtain better accuracy by using fewer training samples compared to the *K*NN and CART methods. Moreover, while the RP-SVM method achieves less data reduction than the KMSVM method, it is seen that the RP-SVM method is more stable than the KMSVM method by obtaining good results in terms of accuracy in all data sets.

The results obtained within the scope of this study show that the natural structures obtained from the data set can contribute to the supervised learning process. The proposed methods can be improved and adapted to different machine learning methods. In addition, the proposed approaches can provide motivation for big data studies.

Key words: CURE algorithm, clustering, natural structures, representative points, support vector machines, structural information, classification.

TEŞEKKÜR

Tezimin hazırlanması esnasında hiçbir yardımını esirgemeyen, bilgi ve deneyimi ile tez çalışmama yön veren ve değerli katkılar sağlayan tez yöneticisi hocam, Sayın Doç. Dr. Halil Murat ÜNVER'e ve tez konumun belirlenmesinde yardımcı olan ve tüm süreçte yardımını gördüğüm hocam, Sayın Prof. Dr. Güvenç ARSLAN'a teşekkürlerimi sunarım.

Ayrıca, tez komitemde yer alan Prof. Dr. Necaattin BARIŞÇI, Doç. Dr. Bülent Gürsel EMİROĞLU, Doç. Dr. Ahmet ARSLAN ve Dr. Öğr. Üyesi Enes AYAN'a değerli katkılarından dolayı teşekkür ederim.

Son olarak, hayatım boyunca beni destekleyen ve daima yanımda olan aileme sonsuz teşekkür ederim.

İÇİNDEKİLER DİZİNİ

Sayfa

ÇİZELGELER DİZİNİ

Sayfa

ŞEKİLLER DİZİNİ

ŞEKİL Sayfa

SİMGELER VE KISALTMALAR

Simgeler

Σ	Toplam fonksiyonu
ϕ	Çekirdek fonksiyonu

Kısaltmalar

AbDG	Özellik Tabanlı Karar Çizgesi / Attribute-based Decision Graph
CART	Sınıflandırma ve Regresyon Ağaçları / Classification and Regression Tree
CBGSVM	Kümeleme Tabanlı Geometrik Destek Vektör Makineleri / Clustering-Based Geometric Support Vector Machines
CB-SVM	Kümeleme Tabanlı Destek Vektör Makinesi / Clustering-Based Support Vector Machine
CR	Sıkıştırma Oranı / Compression Rate
CURE	Temsilcileri Kullanarak Kümeleme / Clustering Using Representatives
CV	Çapraz Doğrulama / Cross Validation
FCM	Bulanık c-ortalamalar / Fuzzy C-means
KA-SVM	Destek Vektör Makinesi için K-ortalamalar tabanlı Aktif Öğrenme / K-means based on Active Learning for Support Vector Machine
KEEL	Knowledge Extraction based on Evolutionary Learning
KMSVM	K-ortalamalar Destek Vektör Makinesi / K-means Support Vector Machine
*K*NN	*K*-En Yakın Komşuluk / K-Nearest Neighbor

KS-SVM	K-uzamsal Medyanlar SVM / K-spatial Medians SVM
NC-MSI	Doğal Kümeler Tabanlı En Benzer Örnekler / Natural Clusters-based Most Similar Instances
NC-SVM	Doğal Kümeler Tabanlı Destek Vektör Makinesi / Natural Clusters-based Support Vector Machine
NC-SVM-B	Doğal Kümeler Tabanlı Destek Vektör Makinesi-Sınırlar / Natural Clusters-based Support Vector Machine-Boundaries
nCV	İç İçe Çapraz Doğrulama / Nested Cross Validation
RP-SVM	Doğal Kümeler Tabanlı Destek Vektör Makinesi / Representative Points-based Support Vector Machine
RBF	Radyal Tabanlı Fonksiyon / Radial Basis Function
RVMs	İlgililik Vektör Makineleri / Relevance Vector Machines
SNC	Benzerlik Tabanlı Doğal Kümeler / Similarity-based Natural Clusters
SVM	Destek Vektör Makinesi / Support Vector Machine
SVMs	Destek Vektör Makineleri / Support Vector Machines
UCI	University of California Irvine

1. GİRİŞ

Son yıllarda, hızla yeni teknolojilerin gelişmesi veri toplamanın benzeri görülmemiş bir şekilde büyümesine yol açmıştır. Pratik uygulamaların yaygınlaşmasıyla birlikte bilgi işlem gücü ve veri toplamadaki üstel büyümeyle desteklenen makine öğrenmesi, günümüzde stratejik öneme sahip bir alan haline gelmiştir (Gambella vd., 2021). İnsan, analiz yaparken veya çoklu özellikler arasında ilişkiler kurmaya çalışırken sıklıkla hata yapmaya meyillidir (Kotsiantis vd., 2007). Bu da belirli problemlerin insanlar tarafından çözülmesini zorlaştırmaktadır. Çeşitli algoritmalar barındıran makine öğrenmesi, bu tarz problemlerin başarılı bir şekilde çözülmesini sağlamaktadır. Makine öğrenmesi algoritmaları, veri kümelerini analiz etmek ve problem karmaşıklığı nedeniyle algoritmik bir çözümün mümkün olmadığı durumlarda karar verme sistemleri oluşturmak için giderek daha fazla kullanılmaktadır (Verbraeken vd., 2020). Makine öğrenmesi algoritmaları, bilgiyi doğrudan veri üzerinden öğrenmek için kullanılan hesaplamalı yöntemlerdir. Bu yöntemler, model olarak önceden tanımlanmış eşitliğe dayanmazlar. Veri üzerinde kavrama sağlayan ve daha iyi kararlar ya da tahminler yapılmasını sağlayan verideki doğal örüntüleri tespit etmeye çalışırlar.

Hayatın her alanına ait büyük ölçekli verilerin öneminin artması, makine öğrenmesi algoritmaları temelli yeni talepleri ortaya çıkarmaktadır (Jordan ve Mitchell, 2015). Veri, makine öğrenmesinde önemli bir rol oynamaktadır. Veri örüntüleri ise öğrenme sonuçlarını ve etkilerini belirlemektedir (Meng vd., 2020). Tüm makine öğrenmesi yöntemleri girdi olarak veri almakta ve veri üzerinde işlemler yapmaktadır. Ancak uygulanan probleme göre yöntemler ve çıktılar farklılık gösterebilmektedir. Son dönemlerde, makine öğrenmesinde veriden daha etkin yararlanmaya yönelik çalışmalar dikkat çekmektedir. Herhangi bir makine öğrenmesi algoritmasının performansı, modeli eğitmek için kullanılan verilere bağlıdır (Camacho vd., 2018). Bu nedenle, özellikle veriden daha etkin yararlanmak ve bu şekilde denetimli öğrenmenin verimliliğini artırmak için farklı yöntemler araştırıldığı görülmektedir. Örneğin; Lopes vd. (2009) eğitim veri kümesinin topolojik yapısını çıkaran optimal K-ilişkili ağ (optimal K-associated network) temelli bir sınıflandırma yöntemi geliştirmişlerdir.

Kayaalp ve Arslan (2014) ise Bulanık c-ortalamalar (Fuzzy c-means, FCM) algoritması yardımıyla örnekleri kümeleyerek veri kümesinin yapısı hakkındaki bilgiyi kullanan yeni bir algoritma geliştirmişlerdir. Bununla birlikte, veri kümesini alt parçalara ayırarak yerel çözümler üzerinden daha etkin yöntemler araştıran çalışmalar da bulunmaktadır. Örneğin; Gu ve Han (2013), parçala ve yönet mantığına dayalı bir Kümelenmiş Destek Vektör Makinesi (Clustered Support Vector Machine, CSVM) yöntemi önermişlerdir. Bu yöntem, K-ortalamalar kümeleme (K-means clustering) yöntemi ile veriyi birkaç küme halinde gruplamakta ve ardından veriyi yerel olarak ayırmak için her bir kümede bir lineer destek vektör makinesi eğitmektedir. Örneklenen bu çalışmalarda, daha verimli makine öğrenmesi yöntemleri geliştirilebilmek için verinin yapısal bilgisinden veya verideki yerel bilgiden yararlanıldığı yani verinin daha etkin kullanımının araştırıldığı görülmektedir.

Veri kümesinden elde edilen yapısal bilginin kullanıldığı yöntemlerden birisi Destek Vektör Makineleridir (Support Vector Machines, SVMs). Denetimli öğrenme yöntemlerinden biri olan destek vektör makineleri zamanla yapay sinir ağları gibi geleneksel metotlara göre genelleme yeteneğinin mükemmelliğinden dolayı makine öğrenmesi topluluğunda ünlü ve popüler olmuştur (Widodo ve Yang, 2007). Ancak SVM yönteminin de bazı kısıtlamaları bulunmaktadır. Bunlardan birisi, hem eğitim hem de test aşamasındaki hız ve boyut problemidir (Byun ve Lee, 2002). Eğitim kümesindeki örnek sayısı arttıkça SVM yönteminin performansı önemli ölçüde azalmaktadır (Almasi ve Rouhani, 2016; Sayed ve Hassanien, 2017). Ayrıca milyonlarca destek vektörü ile çok büyük veri kümeleri için eğitim işlemi çözülememiş bir problemdir (Burges, 1998). Bu problemler dikkate alındığında, eğitim veri kümesinin yapısal bilgisinden faydalanarak veri kümesini özetleyen en anlamlı bilginin elde edilmesi ve bunun denetimli öğrenme sürecinde kullanılması önem kazanmaktadır. Literatürde, farklı yaklaşımlarla veriden elde edilen yapısal bilgiyi SVM yönteminde kullanan ve bu şekilde yöntemin performansını korurken hız ve boyut problemlerine çözüm arayan veya yöntemin performansını artırmaya çalışan çeşitli çalışmalar yer almaktadır (Yu vd., 2003; Wang vd., 2005; Chen ve Pan, 2010; Bang vd., 2010; Horng vd., 2011; Chitrakar ve Chuanhe, 2012; Yao vd., 2013; Bang ve Jhun, 2014; Almasi ve Rouhani, 2016; Gan vd., 2017). Bu çalışmalarda, bir denetimsiz öğrenme yöntemi olan ve bir dizi varlığın "doğal gruplara" ayrıştırılmasıyla eş anlamlı olarak tanımlanan (Gaertler, 2005) kümelemeden

yararlanıldığı görülmektedir. Kümeleme ile etiketsiz veriden doğal yapılanmalar/yapısal bilgi tespit edilmekte ve elde edilen yapısal bilgi farklı yaklaşımlarla SVM yöntemine adapte edilmektedir. Mevcut çalışmalar incelendiğinde ve giderek artan veri boyutları dikkate alındığında veri kümesinden daha etkin yararlanmak için daha fazla araştırmaya ihtiyaç olduğu görülmektedir.

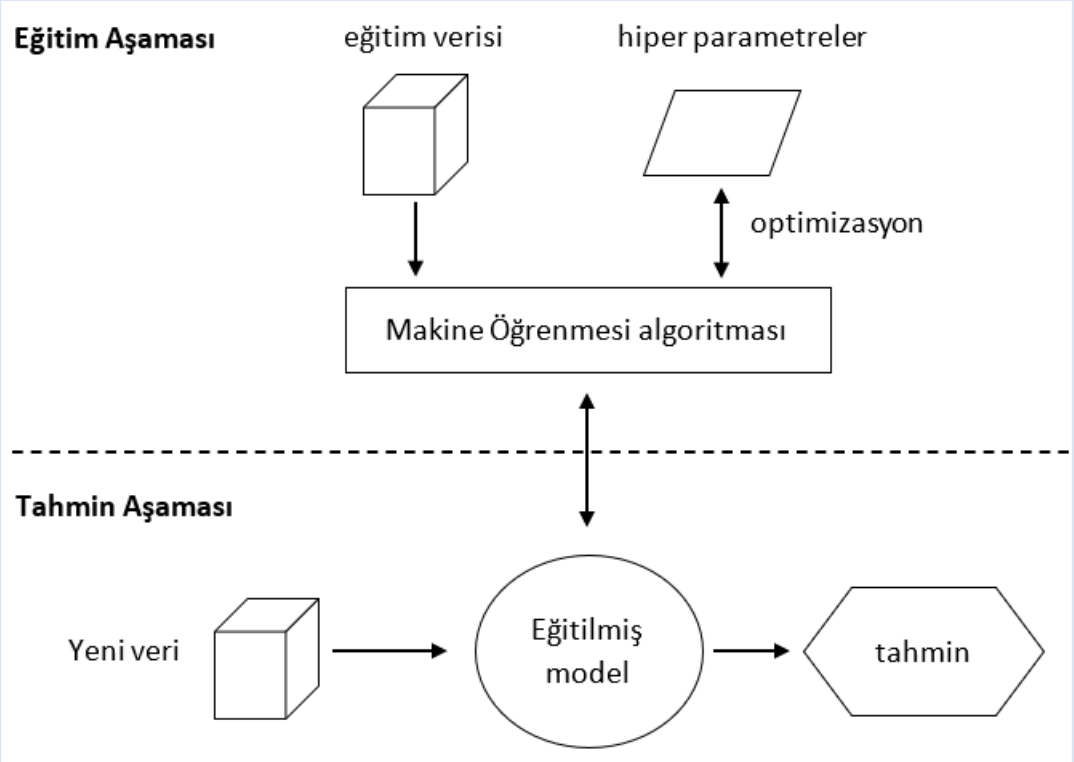
1.1. Makine Öğrenmesi

Makine öğrenmesi, bilgisayarın her problemi kapsamlı ve açık bir şekilde programlamadan sağlanan verilerle "öğrenmesini" sağlayan bir tekniktir. Veri girdilerindeki derin ilişkileri modellemeyi amaçlar ve bir bilgi şemasını yeniden yapılandırır. Öğrenmenin sonucu ise kestirim, tahmin ve sınıflandırma için kullanılabilir (Meng vd., 2020). Yapay zekânın bir alt kümesi olan makine öğrenmesi, görevi gerçekleştirmek için açıkça programlanmadan tahminler veya kararlar almak amacıyla "eğitim verileri" olarak bilinen örnek verilere dayalı bir matematiksel model oluşturur (Zhang, 2020). Makine öğrenmesinin genel amacı, görünmeyen sorunların nasıl ele alındığını bildiren verideki örüntüleri tanımaktır. Örneğin, otonom bir araba gibi oldukça karmaşık bir sistemde, sensörlerden gelen büyük miktardaki veri, “tehlike” örüntüsünü tanımayı “öğrenmiş” bir bilgisayar tarafından arabanın nasıl kontrol edileceğine dair kararlara dönüştürülmelidir (Carleo vd., 2019). Makine öğrenmesinin önemli bir avantajı, makine öğrenmesi yöntemlerinin aksi durumda gözden kaçırılması mümkün olan örüntüleri bulmak için çok miktarda veriyi inceleyebilmesidir (Camacho vd., 2018).

Son yıllarda, daha uygun maliyetli ve daha hızlı bilgi işlem gücüyle birlikte toplanan ve depolanan verinin bolluğu, verideki eğilimleri veya örüntüleri bulmak için algoritmaların hızla gelişmesine neden olmuştur. Bu da, makine öğrenmesi alanını ortaya çıkarmıştır. Bilgisayar biliminin bir alt alanı olarak doğmuş olan makine öğrenmesi, hafif modelleme varsayımlarını kullanmakta ve faydalı karar kurallarını keşfetmek amacıyla büyük veri kümeleriyle uğraşabilen ölçeklenebilir algoritmalar oluşturmak için veri, istatistik ve hesaplama teorisine dayanmaktadır (Bastani vd., 2020). Birçok gerçek dünya olayı, doğrudan kapalı bir formda girdi-çıktı ilişkisi olarak modellenemeyecek kadar karmaşıktır. Makine öğrenmesi, mevcut verileri işleyerek ve probleme bağlı bir performans kriterini en üst düzeye çıkararak bu karmaşık ilişkilerin hesaplama modelini otomatik olarak oluşturabilen teknikler sağlar.

Model oluşturmanın otomatik sürecine “eğitim”, eğitim amacıyla kullanılan verilere ise “eğitim verisi” adı verilir. Eğitilen model, girdi değişkenlerinin çıktı ile nasıl eşlendiğine dair yeni bilgiler sağlayabilir ve eğitim verisi dışındaki yeni girdi değerleri için tahminler yapmak amacıyla kullanılabilir (Baştanlar ve Özuysal, 2014).

Makine öğrenmesine dair genel bir diyagram Şekil 1.1’de verilmiştir. Eğitim aşamasında, eğitim verisi kullanılarak ve hiper parametreler ayarlanarak bir makine öğrenmesi modeli optimize edilir. Ardından, sisteme sağlanan yeni veriler için tahminler oluşturmak üzere eğitilen model kullanılır.

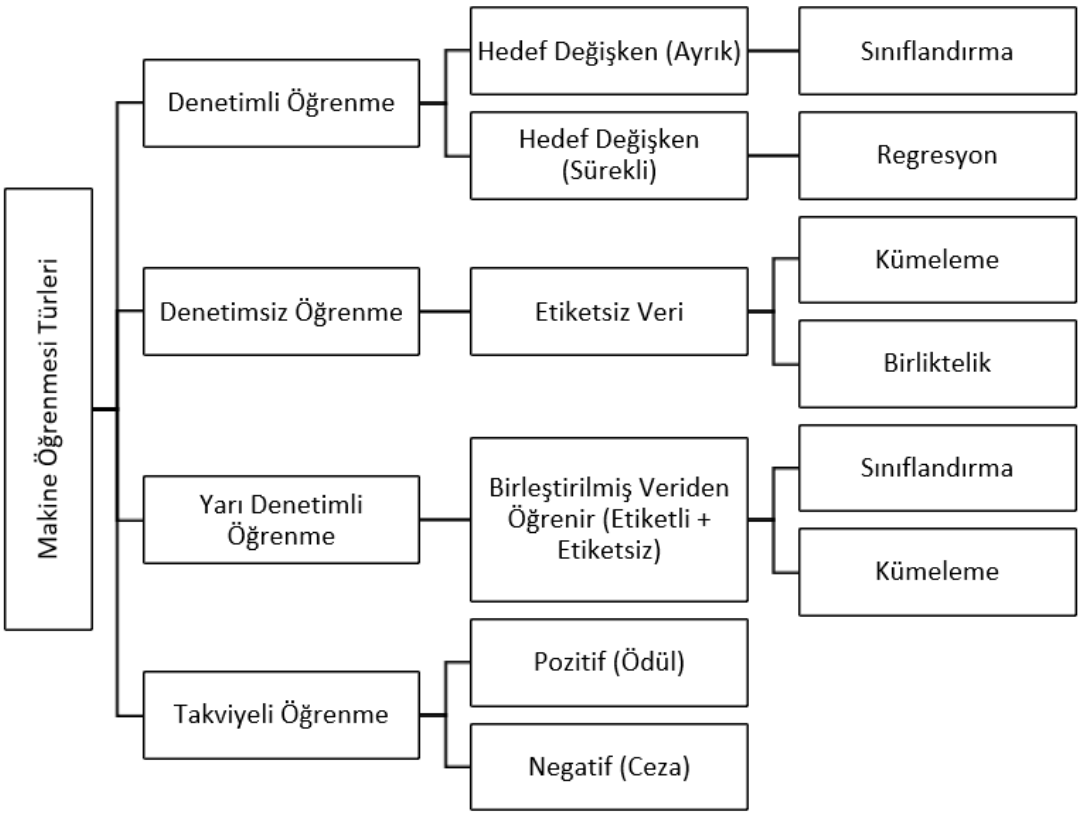


Şekil 1.1. Makine öğrenmesi genel diyagram (Verbraeken vd., 2020)

Makine öğrenmesi, öğrenme sistemlerinin resmi çalışmasına ayrılmış bir araştırma alanıdır. İstatistik, bilgisayar bilimi, mühendislik, bilişsel bilim, optimizasyon teorisi ve diğer birçok bilim ve matematik disiplininden fikirler ödünç alan ve bunlara dayanan oldukça disiplinler arası bir alandır (Ghahramani, 2003). Makine öğrenmesi; metin ve doküman sınıflandırma, doğal dil işleme, konuşma işleme uygulamaları, bilgisayarlı görü uygulamaları, hesaplamalı biyoloji uygulamaları gibi çok geniş bir dizi pratik uygulamaya izin vermektedir. Ayrıca kredi kartı, telefon ve sigorta şirketleri için sahtekârlık tespiti, ağ saldırısı tespiti, satranç, tavla veya Go gibi oyunları oynamayı öğrenme, robotlar ya da arabalar gibi araçların yardımsız kullanımı, medikal

teşhis, öneri sistemlerinin tasarımı, arama motorları ya da bilgi çıkarma sistemleri gibi problemler makine öğrenmesi teknikleri kullanılarak ele alınmaktadır (Mohri vd., 2018).

Makine öğrenmesi, veri ve bilgi arasındaki temel ilişkileri sentezlemek için sistematik olarak algoritmalar uygulamaktadır (Awad ve Khanna, 2015). Makine öğrenmesi algoritmaları öğrenme türlerine göre temel olarak denetimsiz öğrenme (unsupervised learning), denetimli öğrenme (supervised learning), yarı denetimli öğrenme (semi supervised learning) ve takviyeli öğrenme (reinforcement learning) olmak üzere 4 gruba ayrılabilir (Xu, 2019). Makine öğrenmesi türleri Şekil 1.2'de detaylı olarak verilmiştir.



Şekil 1.2. Makine öğrenmesi türleri (Sarker, 2021)

1.2. Makine Öğrenmesi Türleri

1.2.1. Denetimsiz Öğrenme

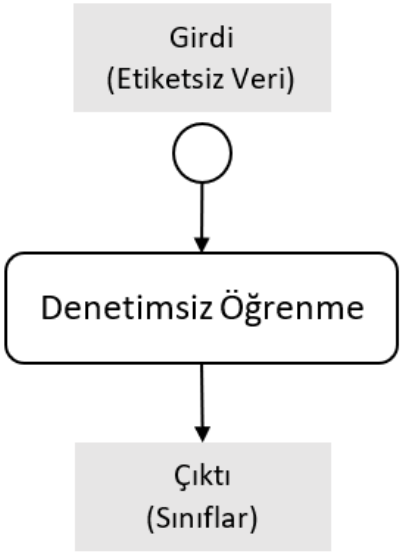
Denetimsiz öğrenme, tanımlayıcı veya yönlendirilmemiş sınıflandırma olarak bilinmektedir. İstenen çıktıya (etikete) sahip olmayan veri bulunması durumunda,

"denetimsiz" olarak adlandırılmaktadır (Kwon vd., 2019). Denetimsiz öğrenme, verilerdeki gizli kalıpları veya gruplamaları keşfetmek için kullanışlıdır (Paper, 2020). Tüm $i \in [n] \coloneqq \{1,2,\dots,n\}$ için $x_i \in \mathcal{X}$ olmak üzere $X = (x_1,\dots,x_n)$, n örnekten (ya da noktadan) oluşan bir küme olsun. Satırlar olarak veri noktalarını içeren $(n \times d)-$matris $\mathbf{X} = (x_i^{\mathrm{T}})_{i\in[n]}^{\mathrm{T}}$ tanımlamak genellikle uygun olmaktadır. Bu durumda, denetimsiz öğrenmenin amacı X verisindeki ilginç yapıları bulmaktır (Olivier vd., 2006).

Denetimsiz öğrenmede, makine öğrenmesi algoritması, önceden etiketlenmemiş bir veri kümesindeki veri örüntülerini aramaya çalışır. Örneğin, reklam kampanyalarını yönetmeye yardımcı olan çevrimiçi bir reklamcılık platformunun günlük operasyonları düşünülebilir. Platform, web sitelerini ziyaret eden ve reklamlara tıklayan çok sayıda kullanıcının veri kayıtlarını toplar. Bu örnekte denetimsiz öğrenme, platformun belirli bir kullanıcının belirli bir reklama tıklayıp tıklamayacağını tahmin etmesine yardımcı olur. Bu durumda, odak kullanıcının geçmiş davranışları, demografik özellikleri ve reklamların karakteristiği "özellikler" ve odak kullanıcının tıklama veya geri dönüş gibi bir reklama yönelik eylemi "etiket" olmaktadır. Aynı platforma, önceki tüketicilerin davranışlarının örüntülerini belirlemek ve tüketicileri farklı kategorilere göre ayırmak ve buna göre reklam gösterimini ayarlamak için de denetimsiz öğrenme uygulanabilir. Bu durum denetimsiz öğrenmedir çünkü müşteri kategorileri etiketlerden değil özelliklerden öğrenilmektedir (Bastani vd., 2020).

Denetimsiz öğrenmede yanıt değişkenleri yani etiketler mevcut değildir ve denetimsiz öğrenmenin amacı gözlemlerin altında yatan karakteristikleri anlamaktır. Dolayısıyla denetimsiz öğrenme, verinin dağılımından verideki ayırt edici özellikleri ve ilişkileri öğrenmeye çalışır. Bu nedenle denetimsiz öğrenmenin ana kullanım durumu, içgörüleri (insight) çıkarmak için örnekleri bölümlemek ve kümelemek amaçlı keşifsel veri analizidir (Gambella vd., 2021).

Denetimsiz öğrenme mantığı Şekil 1.3'te verilmiştir.



Şekil 1.3. Denetimsiz öğrenme mantığı (Wickham, 2018)

Denetimsiz öğrenmenin yaygın uygulamaları şunlardır (Bonaccorso, 2017):

- Nesne segmentasyonu (Örneğin: kullanıcılar, ürünler, filmler, şarkılar vb.)
- Benzerlik tespiti
- Otomatik etiketleme

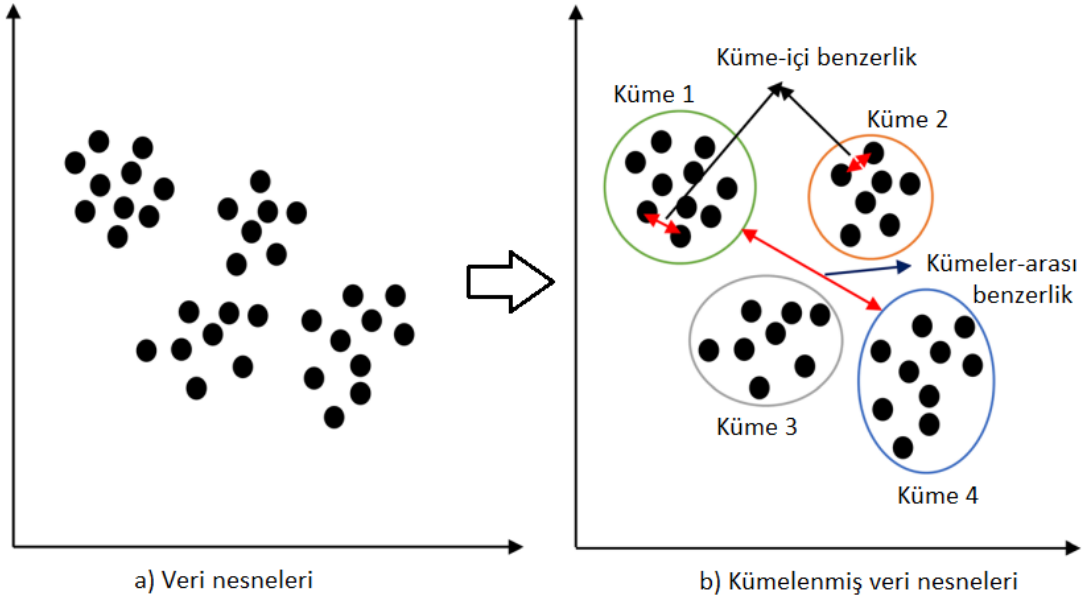
Denetimsiz öğrenme, kümeleme ve boyut indirgeme (dimensionality reduction) olmak üzere iki ana alt tipe sahiptir (Mak vd., 2019).

❖ **Kümeleme**

Kümeleme; kayıtların, gözlemlerin veya vakaların benzer nesnelerin sınıfları halinde gruplandırılmasını ifade eder. Küme, birbirine benzeyen ve diğer kümelerdeki kayıtlardan farklı olan kayıtlar topluluğudur. Kümeleme, kümeleme işlemi için hedef değişken olmaması bakımından sınıflandırmadan farklıdır. Kümeleme görevi, bir hedef değişkenin değerini sınıflandırmaya, kestirmeye veya tahmin etmeye çalışmaz. Bunun yerine kümeleme algoritmaları, tüm veri kümesini, küme içindeki kayıtların benzerliğinin en üst düzeye çıkarıldığı ve bu kümenin dışındaki kayıtlara olan benzerliğin en aza indirildiği, nispeten homojen alt gruplara veya kümelere ayırmaya çalışır (Larose, 2015). $i \neq j$ için $S = \bigcup_{i=1}^{k} C_i$ ve $C_i \cap C_j \neq \emptyset$ olmak üzere kümeleme yapısı S'nin $C = C_1, \dots, C_k$ alt kümelerinin bir dizisi olarak temsil edilir. Sonuç olarak, S'deki herhangi bir örnek tam olarak bir ve yalnızca bir alt kümeye aittir. (Rokach ve Maimon, 2005).

Kümelemede örüntülere eklenmiş/iliştirilmiş herhangi bir etiket olmadığı için kümeleme, denetimli sınıflandırmadan daha zordur. Denetimli sınıflandırma durumunda verilen etiket, veri nesnelerinin bir bütün olarak gruplandırılması için ipucu olmaktadır. Kümeleme durumunda ise etiket olmadığı için bir örüntünün hangi gruba ait olacağına karar vermek zorlaşır (Saxena vd., 2017).

Küme analizi, "benzer" (bazı benzerlik ölçütlerine dayalı olarak) nesneleri birlikte gruplayarak "doğal" gruplamalar bulma sürecidir (Dy ve Brodley, 2004). Veri noktalarının, ait oldukları kümelere göre düzenlenmesi Şekil 1.4'te gösterildiği gibi küme içi yüksek benzerlik ve kümeler arası düşük benzerlikle sonuçlanmalıdır.



Şekil 1.4. Küme içi ve kümeler arası gösterimlerle kümeleme örneği (Ezugwu vd., 2020)

Veri kümesindeki kümeleme bilgisini kullanarak en az insan etkileşimi ile veri içinde gezinen ve keşfeden yöntemler tasarlanabilir. Bu nedenle kümeleme otomatik bilgi işlemenin temel bir yönüdür (Gaertler, 2005). Benzer örneklerin/nesnelerin gruplandırıldığı kümeleme için iki nesnenin benzer olup olmadığını belirleyebilecek bir tür ölçü gereklidir. Bu ilişkiyi tahmin etmek için uzaklık ölçüleri ve benzerlik ölçüleri olmak üzere iki ana ölçü türü vardır. Birçok kümeleme yöntemi, herhangi bir nesne çifti arasındaki benzerliği veya farklılığı belirlemek için uzaklık ölçülerini kullanır (Rokach ve Maimon, 2005).

❖ **Boyut İndirgeme**

Boyut indirgeme, sahip olunan girdilerin sayısını azaltma görevidir. Boyut indirgeme teknikleri, diğer makine öğrenmesi görevlerini daha doğru hale getirmek için verileri daha kolay kullanmaya ve genellikle gürültüyü ortadan kaldırmaya olanak tanır. Genellikle, verileri başka bir algoritmaya uygulamadan önce temizlemek için yapılabilecek bir ön işleme adımıdır (Harrington, 2012). Boyutluluğun indirgenmesi, girdi uzayındaki benzer noktaların manifold üzerindeki komşu noktalara eşlenmesi için, yüksek boyutlu girdilerin daha az boyutluluğa eşlenmesini gerektirir (Reddy vd., 2020). Boyut/özellik sayısını azaltma adımı olan boyut indirgeme, çok boyutlu verilerin analizindeki en önemli görevlerden biridir. Boyut indirgeme işlemini gerçekleştirmek için temel motivasyonlar şunlardır (Baştanlar ve Özuysal, 2014):

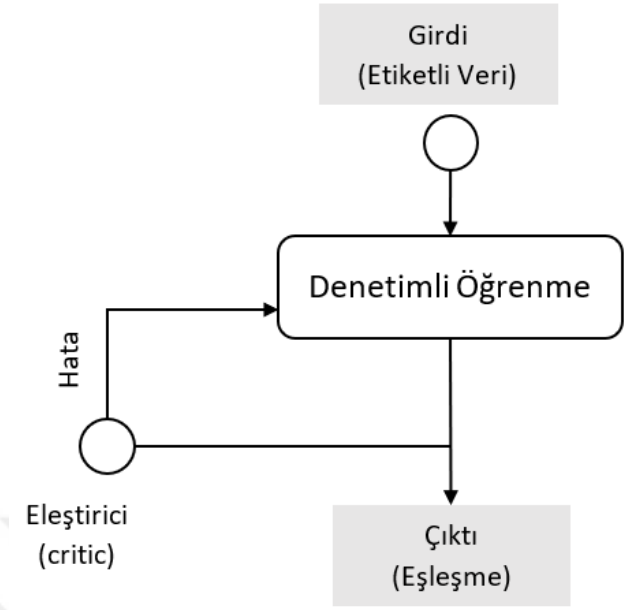
- Daha az özellik ile hesaplama daha hızlıdır.
- Ayırt edici olmayan özellikler tespit edilerek kaldırılırsa zamandan ve iş gücünden tasarruf sağlanabilir ve tahmin doğruluğu artırılabilir.
- İki veya üç boyutlu iz düşümler; içgörü sağlamak için verinin görsel olarak temsil edilmesine, bariz aykırı değerler için veriyi taramaya ve denetimsiz öğrenmeyi kullanırken küme eğilimlerini gözlemlemeye yardımcı olur.

1.2.2. Denetimli Öğrenme

Denetimli öğrenme, bir dizi girdi değişkeni ile bir çıktı değişkeni arasında bir eşlemenin öğrenilmesini ve bu eşlemenin görülmeyen (çıktıları bilinmeyen) veriler için çıktıları tahmin etmek üzere uygulanmasını gerektirir (Cunningham, 2008). Geleneksel denetimli öğrenmede, hipotezler çok sayıda eğitim örneğinden öğrenilir. Her eğitim örneğinin, örnek tarafından açıklanan olayın istenen çıktısını gösteren bir etiketi vardır. Sınıflandırma probleminde etiket, ilgili örneğin dâhil olduğu kategoriyi belirtirken regresyonda; sıcaklık, yükseklik veya fiyat gibi gerçek değerli bir çıktıyı ifade eder (Zhou ve Li, 2010).

Denetimli öğrenmenin amacı, (x_i, y_i) çiftlerinden oluşan bir eğitim kümesi verildiğinde, x'ten y'ye olan bir eşleşmeyi öğrenmektir. Burada $y_i \in Y$, x_i örneklerinin etiketleri veya hedefleri olarak adlandırılır. Etiketler gerçek sayı (regresyon) veya tamsayı (sınıflandırma) ise $\mathbf{y} = (y_i)_{i\in[n]}^{\mathrm{T}}$ etiketlerin sütun vektörünü ifade eder (Olivier vd., 2006).

Denetimli öğrenme mantığı Şekil 1.5'te verilmiştir.



Şekil 1.5. Denetimli öğrenme mantığı (Jones, 2017)

Denetimli öğrenmede, bilinen yanıtın/etiketin tahminini değerlendirerek yapılan net bir doğruluk ölçüsü bulunurken, denetimsiz öğrenmede çıkarılan yapının geçerliliğini değerlendirmek zordur (Gambella vd., 2021).

Yaygın denetimli öğrenme uygulamaları şunlardır (Bonaccorso, 2017):

- Regresyon ve kategorik sınıflandırmaya dayalı tahmin analizi
- İstenmeyen e-posta tespiti
- Örüntü tanıma
- Doğal dil işleme
- Duygu analizi
- Otomatik görüntü sınıflandırma
- Otomatik sıra işleme (Örneğin, müzik veya konuşma)

Denetimli öğrenmenin, sınıflandırma ve regresyon olmak üzere iki ana alt tipi vardır (Mak vd., 2019).

❖ **Sınıflandırma**

Sınıflandırma, insan faaliyetinin en sık karşılaşılan karar verme görevlerinden birisidir. Bir takım gözlenen özelliklere dayalı olarak bir dizi grup veya sınıfın kümelenmesinden sonra, bir nesnenin, o nesneyle ilgili önceden tanımlanmış bir gruba veya sınıflara atanması gerektiğinde bir sınıflandırma sorunu ortaya çıkar. Başka bir deyişle, belirli bir test örneğinin bir dizi sınıf arasından hangi sınıfa ait olduğuna karar verilmesi gerekir (Zhang, 2020). Sınıflandırma, makine öğrenmesinin en yaygın görevlerinden biridir ve önceden sunulan kategorilerden birinde (sınıflarda) bilinmeyen örneği sınıflandırma sorunudur. Sınıflandırmadaki önemli gözlem, hedef fonksiyonların kesikli (discrete) olmasıdır. Genel olarak, sınıf etiketine sayısal veya diğer bazı değerler anlamlı bir şekilde atanamaz. Bu, değeri belirlenmesi gereken sınıf özelliğinin kategorik özellik olduğu anlamına gelir (Novaković vd., 2017).

Veriyi sınıflandırma görevi, her x_i'nin bilinen bir y_i sınıf etiketine sahip olduğu eğitim veri kümesine (X, y) dayalı olarak etiketlenmemiş bir x veri öğesinin sınıf üyeliğine karar verilmesidir (Gambella vd., 2021). Örneğin, bir hedef pazarlama uygulamasında, her kayıt, müşterinin belirli bir ürüne olan ilgisini (veya ilgisinin eksik olmasını) temsil eden belirli bir etiketle etiketlenebilir. Müşterilerle ilişkilendirilen etiketler, müşterinin önceki satın alma davranışından alınmış olabilir. Ek olarak, müşteri demografisine karşılık gelen bir dizi özellik de mevcut olabilir. Amaç, satın alma davranışı bilinmeyen bir müşterinin demografik özelliklerini sınıf etiketiyle ilişkilendirerek belirli bir ürünle ilgilenip ilgilenmeyeceğini tahmin etmektir. Bu nedenle, daha sonra sınıf etiketlerini tahmin etmek için kullanılan bir eğitim modeli oluşturulur (Aggarwal, 2015).

❖ **Regresyon**

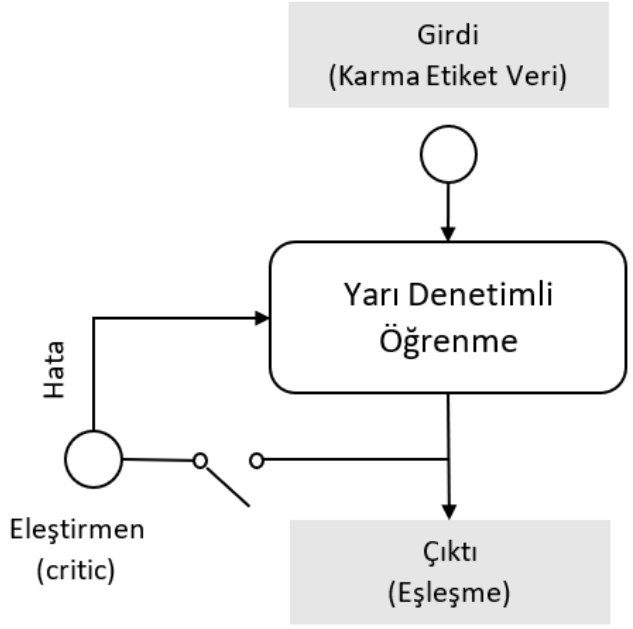
Regresyon, her bir öğe için gerçek bir değer tahmin etme problemidir. Regresyon örnekleri arasında, stok değerlerinin veya ekonomik değişkenlerin varyasyonlarının tahmini yer alır. Regresyonda, yanlış bir tahminin cezası, tipik olarak çeşitli kategoriler arasında yakınlık kavramının olmadığı sınıflandırma probleminin aksine, gerçek ve tahmin edilen değerler arasındaki farkın büyüklüğüne bağlıdır (Mohri vd., 2018).

1.2.3. Yarı Denetimli Öğrenme

Yarı denetimli öğrenme, denetimli ve denetimsiz öğrenmenin tam ortasındadır. Etiketlenmemiş verilere ek olarak, algoritmaya bazı denetim bilgileri sağlanır, ancak bu tüm örnekler için zorunlu değildir. Çoğu zaman, bu bilgi, bazı örneklerle ilişkili hedefler/etiketler olmaktadır. Bu durumda, $X = (x_i)_{i\in[n]}$ veri kümesi, $Y_l \coloneqq (y_1, \dots, y_l)$ etiketlerinin sağlandığı $X_l \coloneqq (x_1, \dots, x_l)$ noktaları ve etiketi bilinmeyen $X_u \coloneqq (x_{l+1}, \dots, x_{l+u})$ noktaları olmak üzere iki bölüme ayrılabilir (Olivier vd., 2006).

Yarı denetimli öğrenmede küme varsayımı ve manifold varsayımı olmak üzere iki temel varsayım vardır. Küme varsayımı, benzer girdilere sahip verilerin benzer sınıf etiketlerine sahip olması gerektiğini varsaymaktadır. Manifold varsayımı ise benzer girdilere sahip verilerin benzer çıktılara sahip olması gerektiğini varsayar. Küme varsayımı sınıflandırma ile ilgiliyken, manifold varsayımı sınıflandırma dışındaki görevlere de uygulanabilir. Bir anlamda, manifold varsayımı, küme varsayımının bir genellemesidir (Zhou ve Li, 2010).

Şekil 1.6'da yarı denetimli öğrenme süreç diyagramı verilmiştir. Girdi ve çıktılar, eleştirmen (critic) öncesine yerleştirilen bir anahtar dışında denetimli öğrenme tarzıyla aynıdır. Anahtar, bir veri örneği etiketsiz olduğunda eleştirmen fonksiyonun devre dışı bırakılmasını sağlamaktadır (Wickham, 2018).



Şekil 1.6. Yarı denetimli öğrenme mantığı (Wickham, 2018)

Yarı denetimli öğrenme, verilerin nasıl etiketlendiğine bağlı olarak aşağıdaki kategorilere ayrılabilir (Zhang, 2020):

- **Kendi kendine eğitim (self-training)**, kendi kendine öğretmek için kendi tahminlerini kullanan yarı denetimli bir öğrenmedir.
- **Ortak eğitim (co-training)**, ortak eğitim ayarını kullanan çoklu görünüm verileri (multi-view data) için zayıf yarı denetimli bir öğrenmedir ve kendilerine öğretmek için kendi tahminlerini kullanırlar.
- **Aktif öğrenme (active learning)**, öğrenenin, bir uzman veya öğretmen tarafından hangi veri noktalarının etiketlenmesini isteyeceğini belirlemede aktif veya katılımcı bir role sahip olduğu yarı denetimli bir öğrenmedir.

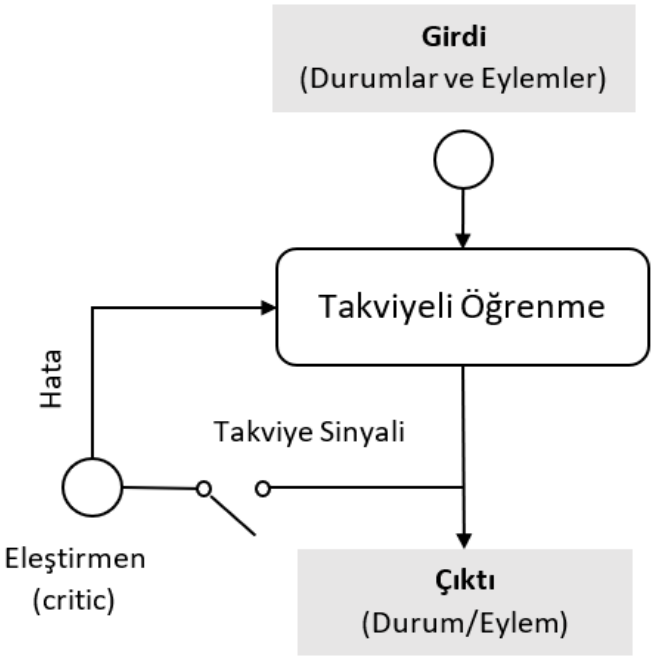
1.2.4. Takviyeli Öğrenme

Eğitim verileri, ödüller ve cezalar şeklinde dinamik bir çevrede bir yapay zekâ ajanına yalnızca geri bildirim olarak verilir. Öğrenme sistemi ve etkileşim deneyimi arasındaki bu geri bildirim, öğrenilmekte olan görevdeki performansı artırmak için yararlıdır. Veri geri bildirimine dayalı bu makine öğrenmesi türüne takviyeli öğrenme denmektedir. Popüler modelsiz takviyeli öğrenme algoritmalarından bir tanesi *Qlearning* algoritmasıdır ve bu algoritma bir ödül veya ceza fonksiyonu öğrenmektedir (Zhang, 2020). Takviyeli öğrenme; robotik, otonom sürüş görevleri, üretim ve tedarik zinciri lojistiği gibi karmaşık sistemlerin otomasyonunu artırmaya veya operasyonel verimliliğini optimize etmeye yardımcı olabilecek yapay zekâ modellerini eğitmek için güçlü bir araçtır, ancak temel veya doğrudan sorunları çözmek için kullanılması tercih edilmez (Sarker, 2021).

Takviyeli öğrenme, istenen davranış örneklerinin bulunmadığı ancak bazı performans kriterlerine göre davranış örneklerinin puanlanmasının mümkün olduğu durumlarda devreye girer. Örneğin, cep telefonu kullanıcıları kapsamanın yetersiz olduğu bir yerde iyi sinyal almak için uğraşır. Bu basit takviyeli öğrenme problemini, bilinmeyen bir R ödül fonksiyonunun optimize edilmesi olarak ele alabiliriz. Dünyada bir x konumu verildiğinde, $R(x)$, o konumda elde edilebilecek ödüldür (örneğin, telefon sinyal gücü). Takviyeli öğrenmenin amacı, maksimum ödül $R(x^*)$ veren x^* konumunu belirlemektir. Bir takviyeli öğrenme sistemine R veya herhangi bir eğitim örneği verilmez. Bunun yerine, x'in değerlerini seçme ve sonuçta ortaya çıkan $R(x)$ ödülünü gözlemleme yeteneğine sahiptir (Barto ve Dietterich, 2004).

Takviyeli öğrenmenin doğrudan (modelsiz) ve dolaylı yöntemler (model tabanlı) olmak üzere iki ana alt türü bulunmaktadır (Mak vd., 2019).

Takviyeli öğrenmenin temel mantığı Şekil 1.7'de verilmiştir.



Şekil 1.7. Takviyeli öğrenme mantığı (Jones, 2017)

1.3. Tezin Amacı ve Katkılar

Makine öğrenmesinde eğitim verilerinin miktarına ek olarak girdi verilerinin kalitesi, tüm makine öğrenmesi sürecinin anahtarıdır. Herhangi bir makine öğrenmesi algoritmasının performansı, modeli eğitmek için kullanılan verilere bağlıdır (Camacho vd., 2018). Bu çalışmada, makine öğrenmesinde önemli bir rol oynayan veriden ve öğrenme sonuçları ve etkilerini belirleyen veri örüntülerinden daha etkin yararlanmak amaçlanmıştır. Literatür çalışmalarından motivasyon alınarak veri kümesindeki doğal yapılanmaların/yapısal bilginin elde edilmesi ve bu bilginin denetimli öğrenme sürecinde kullanılması hedeflenmiştir. Bu doğrultuda, mevcut çalışmalardan farklı olarak, benzerlik tabanlı yeni bir kümeleme yönteminin araştırılması ve bu yöntem ile yapısal bilgilerin tespit edilmesi hedeflenmiştir. Bununla birlikte, ikinci bir yaklaşım olarak eğitim kümesinden yapısal bilginin elde edilmesi amacıyla başarılı bir kümeleme yöntemi olan ve literatürde benzer bir amaçla kullanımı bulunmayan CURE kümeleme yönteminin uygulanması araştırılmıştır.

Bu doğrultuda önemli denetimli öğrenme problemlerinden biri olan sınıflandırma ele alınmıştır. Belirtilen şekilde benzerlik tabanlı kümeleme ve CURE kümeleme yöntemleri ile veri kümesinden doğal yapılanmaların elde edilmesi ve elde edilen yapılanmaların mevcut sınıflandırma yöntemlerine adapte edilmesini sağlayacak yaklaşımların geliştirilmesi üzerine durulmuştur.

Bu çalışmanın katkıları şunlardır:

- Veri kümesindeki doğal yapılanmaların tespitinde kullanılabilecek benzerlik tabanlı yeni bir kümeleme yaklaşımı araştırılmış ve geliştirilmiştir.
- Önerilen benzerlik tabanlı kümeleme yönteminin sınıflandırma sürecine adapte edilmesi için farklı yaklaşımlar araştırılmıştır. Sonuç olarak, doğal yapılanmaları farklı şekillerde kullanarak sınıflandırma sürecini gerçekleştiren 3 faklı sınıflandırma yöntemi önerilmiştir.
- Diğer bir yaklaşımda, mevcut kümeleme yöntemleri incelenmiş ve literatürde benzer amaçlarla kullanımına rastlanmayan CURE kümeleme yöntemi temel alınmıştır. Veri kümesinden, sınıflandırma sürecinde kullanılabilecek yapısal bilginin CURE kümeleme ile tespiti sağlanmıştır.
- CURE yönteminden elde edilen yapısal bilgiyi iyi bilinen sınıflandırma yöntemlerinden birisi olan SVM yöntemine adapte eden yeni bir sınıflandırma yaklaşımı önerilmiştir.
- Önerilen yöntemler, deneysel olarak analiz edilerek sonuçlar sunulmuştur. Araştırma bulgularına göre önerilen yöntemler ile dikkate değer sonuçlar elde edilmiştir. Bu şekilde, incelenen alanda araştırma ve geliştirmeye açık yöntemler ortaya koyulmuştur. Ayrıca önerilen yöntemler, veriden etkin yararlanmanın faydalı olabileceği büyük veri gibi çalışma alanları için araştırma zemini oluşturmaktadır.

1.4. Tezin Organizasyonu

Bu çalışma, aşağıda belirtildiği şekilde organize edilmiştir.

- Bölüm 1: Bu bölümde çalışma için genel bir giriş yapılmış ve makine öğrenmesi hakkında genel bilgilere yer verilmiştir. Bu kapsamda, makine öğrenmesi türleri, ele alınan problemler ve kullanım alanlarına değinilmiştir.

- Bölüm 2: Bu çalışma kapsamında ele alınan araştırma konusuyla benzerlik gösteren literatür çalışmaları ve bu çalışmalardan elde edilen bulgular detaylı olarak incelenmiştir. Bu doğrultuda, benzer çalışmalar ve çalışmalardan elde edilen sonuçlar hakkında bilgi bu bölümde sunulmuştur.
- Bölüm 3: Bu bölümde, çalışma sürecinde kullanılan materyal ve yöntemler hakkında bilgi verilmektedir. Çalışma kapsamında ele alınan makine öğrenmesi algoritmaları hakkında detaylı bilgi sunulmuştur. Bununla birlikte, bu çalışma kapsamında önerilen yöntemler hakkında detaylı bilgi verilmiştir. Bu yöntemlerin temel adımlarına değinilmiş ve yöntemler uygun diyagramlar ile ifade edilmiştir.
- Bölüm 4: Çalışmanın bu bölümünde, önerilen yöntemlerin deneysel olarak analiz edilmesi sürecine yer verilmiştir. Bu kapsamda gerekli parametrelerin belirlenmesi ve değerlendirme yöntemleri hakkında detaylı bilgi verilmektedir. Belirlenen koşullar altında yürütülen deneysel analiz çalışmasına ait bulgular da bu bölümde ele alınmıştır. Elde edilen sonuçlar uygun grafikler kullanılarak karşılaştırmalı olarak sunulmuştur. Ayrıca, deneysel analizden elde edilen bulgular karşılaştırmalı olarak incelenmiş ve sonuçlar tartışılmıştır.
- Bölüm 5: Bu bölümde, önerilen yöntemler ve elde edilen sonuçlar genel olarak değerlendirilmiştir. Ayrıca önerilen yöntemlerin uygulanabilir olduğu alanlara değinilmiştir.

2. BENZER ÇALIŞMALAR

Literatür çalışmaları incelendiğinde, son dönemlerde, mevcut makine öğrenmesi yöntemlerinin performansını artırmak için veri kümesinden daha etkin yararlanmayı amaçlayan çalışmaların arttığı görülmektedir. Bu kapsamda, veri kümesinin çizge tabanlı yöntemlerle temsil edilmesi, veri kümesini alt parçalara ayırarak yerel çözümlerden yararlanılması ve veriden yapısal bilgi elde edilmesini sağlayan kümeleme gibi denetimsiz öğrenme yöntemlerinin uygulanması yönündeki araştırmalar dikkat çekmektedir.

Literatürde, veri kümesini alt parçalara ayırarak yerel çözümler üzerinden daha etkin yöntemler geliştirmeyi amaçlayan çeşitli çalışmalar bulunmaktadır.

Gu ve Han (2013), veriyi parçala ve yönet mantığına göre ele alan CSVM yöntemini önermişlerdir. Bu yöntemde, K-ortalamalar kümeleme algoritması ile veri birkaç küme halinde gruplamakta ve ardından veriyi yerel olarak ayırmak için her bir kümede bir lineer destek vektör makinesi eğitilmektedir. Her bir yerel SVM'nin aşırı öğrenmesini (over-fitting) engellemek için bir global düzenleme (global regularization) eklenmiştir. Çeşitli veri kümeleri üzerinde yapılan testler sonucunda, önerilen yöntemin doğrusal SVM yönteminden ve diğer bazı ilgili yerel lineer sınıflandırıcılardan daha iyi performans elde ettiği tespit edilmiştir. Ayrıca, önerilen yöntemin çekirdek SVM'den (kernel SVM) daha etkin iken tahmin performansı açısından ince ayarlanmış bir çekirdek SVM (fine-tuned kernel SVM) ile karşılaştırılabilir düzeyde olduğu belirtilmiştir. Harris (2015) ise Gu ve Han (2013) tarafından önerilmiş olan CSVM yönteminin kredi skorlama için kullanımını araştırmışlardır. Araştırmacılar, CSVM üzerinde bazı yeni ayarlamalar yapmış ve Gu ve Han (2013) tarafından yerleştirilen lineer çekirdeğe ek olarak yarıçap temel fonksiyon (radius basis function) çekirdeklerin kullanımını araştırmışlardır. Çalışmada, CSVM yöntemi diğer doğrusal olmayan SVM tabanlı tekniklerle karşılaştırılmış ve CSVM yönteminin hesaplama açısından nispeten ucuz kalırken sınıflandırma performansı açısından kıyaslanabilir seviyelere ulaşabildiği tespit edilmiştir.

Son yıllarda veriyi çizge tabanlı yaklaşımlarla modelleyerek denetimli öğrenmenin etkinliğini artıran çalışmalar da dikkat çekmektedir. Çizge tabanlı tekniklerin sağlaması gereken gereksinimler şunlardır (Bertini vd., 2011):

i. Girdi verilerinin topolojik yapısını yakalaması
ii. Hiyerarşik veri sunumuna izin vermesi (yani bir çizge alt çizgelere bölünebilir, her alt çizge daha küçük alt çizgelere bölünebilir vb.)
iii. Rastgele şekillerdeki küme veya sınıfların tespit edilmesi
iv. Verinin yerel yapısı ile global istatistiklerini birleştirmeye izin vermesi

Lopes vd. (2009), veriyi modellemek için çizge tabanlı yeni bir sınıflandırıcı önermişlerdir. Önerilen sınıflandırıcı, optimal *K*-ilişkili ağ olarak adlandırılan özel bir ağ kullanmaktadır. *K*-ilişkili ağ, veri örnekleri ve veri sınıfları arasındaki benzerlik/benzeşmezlik ilişkilerini temsil edebilmektedir. Araştırmacılar yaptıkları deneysel değerlendirme sonucunda, optimal *K*-ilişkili ağa dayanan modelin, eğitim veri kümesinin topolojik yapısını elde ettiğini ve bu şekilde özellikle gürültülü veride sınıflandırma işleminde iyi sonuçlar elde edebildiğini tespit etmişlerdir. Ayrıca çalışma, karmaşık ağların (complex networks) sadece kümeleme problemlerinde değil aynı zamanda sınıflandırma işlemlerinde de kullanılabileceğini göstermektedir.

Bertini vd. (2011) yaptıkları çalışmada, girdi veri kümesinden oluşturulan çizgenin yerel yapısını ve global istatistiksel özelliklerini birleştiren çizge tabanlı parametrik olmayan çok sınıflı bir sınıflandırma algoritması sunmuşlardır. *K*-ilişkili çizge (*K*-associated graph) olarak adlandırılan özel bir çizge oluşturulan bu yöntemde saflık (purity) olarak adlandırılan yeni bir ölçü de tanıtılmıştır. Çalışmada önerilen algoritma parametrik olmayan bir algoritma olduğu için parametre ayarlaması olmadan sınıflandırma işlemini gerçekleştirmekte ve model seçimine ihtiyaç duymamaktadır. Bu durum, pratik uygulamalarda yüksek verimlilik anlamına gelmektedir. Önerilen yöntemin, eğitim kümesinin topolojik yapısını yakaladığı ve özellikle gürültülü verinin sınıflandırılması işinde iyi sonuçlar ortaya koyduğu tespit edilmiştir. Çalışmanın deneysel sonuçlarına göre önerilen algoritmanın, özellikle gürültülü veriler için sınıflandırma doğruluğu açısından diğer iyi bilinen sınıflandırma teknikleri kadar iyi olduğu belirtilmiştir.

Bertini vd. (2013), sınıflandırıcının ömrü boyunca veri dağılımındaki değişikliklerle ilgili olan durağan olmayan sınıflandırma problemlerine (non-stationary classification

problems) değinmiş ve durağan olmayan alanlar üzerinde sınıflandırma ile başa çıkmak için *K*-ilişkili optimal çizge öğrenme algoritmasının bir uzantısını sunmuşlardır. Önerilen yöntem, birçok bağlantısız bileşenden (alt çizgeden) oluşan bir çizge yapısına dayanmaktadır. Ayrıca, zaman içinde yeni veriler sunulduğunda bileşenler dizisini güncelleyerek ve yeni bileşenler ortaya çıktıkça eski bileşenleri kaldırarak çizgenin dinamik evriminden yararlanmaktadır. Önerilen algoritma, yüksek sınıflandırma doğruluğu ile birlikte çizge boyutu ve bellek kullanımı açısından istikrarlı bir performans göstermiştir. Araştırmanın yapay ve gerçek alanlar üzerindeki deneysel sonuçları ve ileri istatistiksel analizler ile önerilen algoritmanın durağan olmayan sınıflandırma problemlerine etkili bir çözüm olduğu tespit edilmiştir.

Mohammadi vd. (2015), *K*-ilişkili optimal çizge algoritmasının eğitim aşamasını değiştirerek ve test aşamasında yeni bir etiketleme yöntemi uygulayarak, farklı seviyelerde gürültü bulunması durumunda sağlam olan yeni bir yöntem sunmuşlardır. Önerilen sınıflandırma yönteminde, veri kümesindeki her bir sınıf, bir alt-çizgeler dizisi olarak ifade edilmiştir. Araştırmacıların eğitim aşamasında yaptıkları iyileştirme sayesinde, gürültülü ve gürültüsüz alt çizgeler ayırt edilebilmektedir. Önerilen yöntem, bir çizge tabanlı sınıflandırıcı ve iyi bilinen sınıflandırıcılardan olan Karar Ağacı ve Çok-Sınıflı Destek Vektör Makinesi ile karşılaştırılmıştır. Önerilen algoritmanın, %5 veya daha yüksek gürültü seviyesinde çizge tabanlı sınıflandırma algoritmasından ortalama %7 daha iyi performans gösterdiği tespit edilmiştir. %20 gürültü seviyesinde ise önerilen algoritmanın, Karar Ağacı ve Çok-Sınıflı Destek Vektör Makinesi yöntemlerinden ortalama %5 daha iyi performans gösterdiği belirtilmiştir.

Bertini vd. (2017), veriden çizgeler oluşturmak için alternatif yollar bulmak amacıyla yeni bir çizge türü oluşturmak üzere Özellik Tabanlı Karar Çizgesi (Attribute-based Decision Graph, AbDG) adı verilen bir algoritma önermişlerdir. Önerilen yöntemde, vektör tabanlı bir veri kümesi verildiğinde, verinin her özellik aralığı ayrık bölüntülere (disjoint intervals) ayrılarak ve her bölüntü bir düğüm noktası olarak temsil edilerek bir AbDG oluşturulur. Sonrasında, farklı özelliklerden gelen düğümler arasında önceden tanımlanmış bir örüntüye göre kenarlar oluşturulur. Bu yöntemde sınıflandırma, yeni örneğin özellik değerleri ile AbDG arasında bir eşleştirme işlemi kullanılarak gerçekleştirilir. Ayrıca, AbDG eksik özellik değerlerini ele almak için bir iç mekanizma sağlayarak uygulanabilirliğini artırmaktadır. Sınıflandırma sonuçlarına göre AbDG'nin iyi bilinen çok sınıflı algoritmalara kıyasla rekabetçi bir yaklaşım

olduğu tespit edilmiştir. Araştırmacılar, bu çalışmada önerilen çerçevenin temel katkısını, sağlam model eşleştirmeli veri sınıflandırması gerçekleştirmek için özellik ve çizge tabanlı tekniklerin avantajlarının birleştirilmesi olarak vurgulamışlardır.

Veri kümesinden daha etkin yararlanmak için uygulanan bir diğer yaklaşım ise bir denetimsiz öğrenme yöntemi olan kümeleme ile elde edilen doğal yapılanmaların/yapısal bilginin denetimli öğrenme sürecinde kullanımıdır. Bu yaklaşımı uygulamak için farklı kümeleme ve sınıflandırma algoritmalarının birleştirildiği çeşitli çalışmalar bulunmaktadır.

Kayaalp ve Arslan (2014), nümerik özellikler için bağımsızlık varsayımı olmadan yeni bir Bulanık Bayes Sınıflandırıcı (Fuzzy Bayes Classifier) geliştirmişlerdir. Sınıflandırmada yüksek doğruluk elde etmek için üyelik fonksiyonları, FCM kümeleme yöntemi ile oluşturulmuştur. Araştırmacılar, bir uzmana danışmak yerine üyelik fonksiyonlarını doğrudan veri kümesi üzerinden elde etmek için FCM kümeleme yöntemini kullanmışlardır. Önerilen yöntem, literatürde iyi bilinen ve sadece sayısal özellikler içeren iki veri kümesi üzerinde gösterilmiştir. Araştırma sonuçları, önerilen Bulanık Bayes Sınıflandırıcının en az diğer yöntemlerle kıyaslanabilir düzeyde olduğunu göstermiştir.

Shamsollahi vd. (2018), koroner arter hastalıkları olan hastaları etkili bir şekilde tahmin etmede sağlık sistemi uzmanlarına yardımcı olmayı amaçlayan ve veri madenciliğinin tanımlayıcı ve tahmin edici tekniklerini birleştiren bir model sunmuşlardır. Bu modelde, bazı kümeleme ve sınıflandırma teknikleri kullanılmıştır. Önerilen yöntemde, tanımlayıcı yöntem olarak K-ortalamalar kümeleme ve tahmin edici yöntemler olarak Karar Ağaçları ve Yapay Sinir Ağı (Artificial Neural Network) sınıflandırma yöntemleri kullanılmıştır. Yöntemde ilk olarak kümeleme indisleri kullanılarak küme sayısı belirlenmekte ve sonrasında koroner arter hastalarının tahmini için her bir kümeye bazı karar ağacı metotları ve yapay sinir ağı uygulanmaktadır. Yapılan çalışmada bir kalp kliniği veri tabanından toplanan gerçek veri kullanılmıştır. Araştırma sonuçlarına göre C&RT Karar Ağacı yönteminin 0,0074 hata ile kullanılan tüm veride en iyi sonucu elde ettiği tespit edilmiştir.

SVM yöntemini daha etkin hale getirmek veya bazı kısıtlarını gidermek için iyi bilinen başarılı kümeleme yöntemlerinden biri olan K-ortalamalar kümeleme yönteminin SVM yöntemi ile adapte edildiği çalışmalar bulunmaktadır.

Wang vd. (2005), gerçek zamanlı iş zekâsı (real time business intelligence) sistemleri için K-ortalamalar kümeleme algoritması ile SVM sınıflandırma yöntemini birleştirerek K-ortalamalar Destek Vektör Makinesi (K-means SVM, KMSVM) yöntemini önermişlerdir. Araştırmacılar, gerçek veri tabanları üzerinde yaptıkları testlerde, KMSVM algoritmasının SVM'ye kıyasla destek vektörlerini azaltarak ve benzer test doğruluğunu sürdürerek yanıt süresini (response time) hızlandırdığını tespit etmişlerdir. Chen ve Pan (2010) ise büyük boyutlardaki veri kümeleri üzerinde Destek Vektör Makinelerinin eğitiminin yavaşlığına değinmiş ve bu problemi çözmek için Kümeleme Tabanlı Geometrik Destek Vektör Makineleri (Clustering-Based Geometric Support Vector Machines, CBGSVM) yöntemini önermişlerdir. Araştırmacılar, CBGSVM yönteminin neredeyse aynı sınıflandırma hassasiyeti altında standart Destek Vektör Makinelerine kıyasla eğitim sürecini hızlandırdığını belirtmişlerdir.

Yao vd. (2013), son eğitim kümesi olarak kullanılmak üzere orijinal eğitim kümesinden sadece küçük bir alt kümenin seçildiği, kümeleme algoritmasına dayalı yeni bir yaklaşım sunmuşlardır. Bu yaklaşımda, K-ortalamalar kümeleme algoritması kullanılarak en bilgi sağlayıcı örnekler seçilmeye çalışılmakta ve seçilen bu örnekler üzerinde eğitilerek SVM sınıflandırıcı oluşturulmaktadır. Araştırmacılar yaptıkları testlerde, önerilen yaklaşımın eğitim kümesinin ölçeğini oldukça azalttığını ve dolayısıyla SVM'nin eğitim ve tahmin süresini büyük ölçüde kısalttığını tespit etmişlerdir. Aynı zamanda, önerilen yöntemin genelleme performansını garanti ettiği belirtilmiştir.

Gan vd. (2017), Destek Vektör Makinesi için K-ortalamalar tabanlı Aktif Öğrenme (K-means based on Active Learning for Support Vector Machine, KA-SVM) yaklaşımını önermişlerdir. Bu yaklaşımda, eğitim kümesi olarak önemli örneklerin bir alt kümesini elde eden bir ön seçim şeması oluşturmak için K-ortalamalar kümeleme yöntemi kullanılmakta ve sonrasında SVM tüm veri kümesi yerine böyle bir alt küme ile eğitilmektedir. Araştırmada bazı veri kümeleri üzerinde benzer yaklaşımlarla yapılan karşılaştırmalar sonucunda, önerilen yaklaşımın hem sınıflandırma doğruluğu hem de hesaplama verimliliği bakımından üstün performansa sahip olduğu tespit edilmiştir.

Bang ve Jhun (2014) farklı küme merkezlerinin sınıflandırmaya katkısının farklı olabileceğine ancak KMSVM algoritmasında yanlış sınıflandırma cezasının her bir

küme merkezi için eşit uygulandığına dikkat çekmiştir. Bu doğrultuda araştırmacılar, sınıflandırma doğruluğunu artırmak için her bir kümedeki veri noktalarının sayısını ağırlık olarak kullanarak küme merkezlerinin yanlış sınıflandırılmasına farklı cezalar uygulayan Ağırlıklandırılmış KMSVM (Weighted KMSVM, WKM-SVM) algoritmasını önermişlerdir. KMSVM yönteminin değiştirilmesi ile elde edilen WKM-SVM algoritması için yapılan testlerde, önerilen WKM-SVM algoritmasının, KM-SVM algoritmasının performansını artırabildiği tespit edilmiştir.

Bazı çalışmalarda ise diğer bir başarılı kümeleme yöntemi olan BIRCH kümeleme yönteminin, SVM sınıflandırma yöntemine adapte edilerek daha etkin sınıflandırma yaklaşımlarının araştırıldığı görülmektedir.

Yu vd. (2003) özellikle çok büyük veri kümelerinde kullanılmak üzere BIRCH kümeleme yöntemini temel alan Kümeleme Tabanlı SVM (Clustering-Based SVM, CB-SVM) yöntemini sunmuşlardır. CB-SVM yöntemi, SVM yöntemine verinin istatistiksel özetlerini taşıyan yüksek kaliteli örnekler sağlamak için tüm veri kümesini yalnızca bir kez tarayan hiyerarşik bir mikro kümeleme algoritması uygulamaktadır. CB-SVM, sınırlı miktarda kaynak verilen çok büyük veri kümeleri için en iyi SVM sınırını oluşturmaya çalışmaktadır. Araştırmacılar, sentetik ve gerçek veri kümeleri üzerinde yaptıkları testler sonucunda, CB-SVM yönteminin yüksek sınıflandırma doğruluğu elde ederken çok büyük veri kümeleri için oldukça ölçeklenebilir olduğunu tespit etmişlerdir.

Horng vd. (2011), hiyerarşik bir kümeleme algoritması, basit bir özellik seçim yöntemi ve SVM algoritmasını birleştiren SVM tabanlı bir saldırı tespit sistemi önermişlerdir. Sistemde kullanılan BIRCH hiyerarşik kümeleme algoritması, SVM yönteminin eğitimi için orijinal büyük veri kümesi yerine yüksek nitelikli, soyutlanmış ve azaltılmış veri kümesi sağlamaktadır. Bu şekilde yöntemin, eğitim süresinde önemli bir azalma sağladığı ve buna ek olarak ortaya çıkan SVM sınıflandırıcıların, orijinal veri kümesini kullanan SVM sınıflandırıcılardan daha iyi performans gösterdiği tespit edilmiştir. Ayrıca, önerilen sistem aynı veri kümesini kullanan diğer saldırı tespit sistemleri ile karşılaştırıldığında, önerilen sistemin Hizmet Reddi (Denial of Service) ve Bilgi Tarama (Probe) saldırılarının tespitinde daha iyi performans gösterdiği ve genel doğrulukta en iyi performansı elde edebildiği belirtilmiştir.

K-means ve BIRCH kümeleme yöntemleri dışında farklı kümeleme yöntemleri ile SVM yöntemini birlikle kullanan çalışmalar da bulunmaktadır.

Bang vd. (2010), gerçek dünya durumlarında veri kümesinin aykırı değerler tarafından kirletilmiş olmasının muhtemel olduğuna ve K-ortalamalar kümeleme yönteminin bu aykırı değerlere karşı hassas olduğuna dikkat çekmişlerdir. Bu doğrultuda aykırı değerlere karşı sağlam olan K-uzamsal medyanlar (K-spatial medians) kümeleme yöntemi ile SVM yöntemini birleştirerek K-uzamsal Medyanlar SVM (KS-SVM) algoritmasını önermişlerdir. Ayrıca, KS-SVM algoritmasını değiştirerek destek vektörlerine yakın veri noktalarını kurtaran ve böylece sınıflandırma doğruluğunu artıran KS-SVM Tabanlı Kurtarma Sürecini (Recovery Process Based on KS-SVM) önermişlerdir. Araştırmacılar yaptıkları testler sonucunda, önerdikleri KS-SVM yönteminin sınıflandırma doğruluğu ve destek vektörlerinin sayısı açısından KM-SVM yönteminin performansını iyileştirebileceğini tespit etmişlerdir. Ayrıca KS-SVM yönteminin birçok açıdan KM-SVM yönteminden daha kararlı olduğunu vurgulamışlardır.

Almasi ve Rouhani (2016), büyük ve gerçek dünya veri kümelerinde SVM eğitimi için hızlı ve gürültü gideren bir SVM eğitim yöntemi önermişlerdir. İlk aşamada, gürültü içeren veya gürültülü olduğundan şüphelenilen veriler belirlenmekte ve orijinal eğitim kümesinden çıkarılmaktadır. Dışbükey gövde verileri, QHull algoritması ile hesaplanmaktadır. Öte yandan, eğitim kümesinin boyutunu sıkıştırmak ve azaltmak için Bulanık Kümeleme Yöntemi uygulanmıştır. Son olarak, azaltılmış ve saflaştırılmış (purified) küme merkezleri, SVM yöntemini eğitmek için kullanmaktadır. Araştırma sonucunda, önerilen yöntemin bilgilendirici ve anlamlı veriler çıkararak büyük ve gerçek dünya veri kümelerinde SVM yönteminin eğitim süresini azaltabildiği tespit edilmiştir. Ayrıca önerilen yöntemin, gürültülü ve gürültü şüphesi olan verilerin kaldırılmasında etkili bir performansa sahip olduğu belirtilmiştir.

Chitrakar ve Chuanhe (2012) doğruluk, tespit oranı (detection rate) ve yanlış alarm oranı (false alarm rate) açısından daha iyi sınıflandırma performansı elde etmek için nispeten daha iyi bir kümeleme yöntemi olan k-Medoids kümeleme yöntemi ile Destek Vektör Makinesini birleştirmişlerdir. Çalışmada, Kyoto2006+ veri kümeleri kullanılarak sınıflandırmanın performansı, doğruluğu, tespit oranı ve yalancı pozitif (false positive) oranını değerlendirmek için simülasyonlar yapılmıştır. Yapılan

deneyler ve analizler sonucunda, önerilen hibrit yaklaşımın, Naive Bayes ile k-ortalamalar/k-Medoids kümelemeyi birleştiren yaklaşımdan daha iyi performans elde ettiği gösterilmiştir. Ayrıca, önerilen yeni yaklaşımın saldırı tespiti (intrusion detection) için daha etkili ve verimli olabileceği belirtilmiştir.

3. MATERYAL VE YÖNTEM

Bu çalışma kapsamında araştırılan yöntemlerin geliştirilmesi ve analiz edilmesi sürecinde çeşitli materyal ve yöntemlerden faydalanılmıştır. Bu bölümde, çalışma kapsamında kullanılan materyal ve yöntemlere detaylı olarak yer verilmiştir.

3.1. Veri Kümesi

Bu çalışmanın deneysel analiz kısmında çeşitli gerçek hayat veri kümeleri kullanılmıştır. Bu veri kümeleri, UCI Makine Öğrenmesi Deposu (UCI Machine Learning Repository) (Dua ve Graff, 2019) ve KEEL Veri Kümesi Deposundan (KEEL Dataset Repository) (Alcalá-Fdez vd., 2011) elde edilmiştir. Kullanılan veri kümeleri, özellikleri ve boyutları bakımından farklılıklara sahiptir. Bu veri kümeleri Çizelge 3.1'de özetlenmiştir.

Çizelge 3.1. Veri kümeleri

Veri Kümesi	Veri Türü	Boyut	Özellik Sayısı	Özellik Karakteristiği	Sınıf Sayısı
Iris	Çok değişkenli	150	4	Reel sayı	3
Wine	Çok değişkenli	178	13	Tam sayı, reel sayı	3
Sonar	Çok değişkenli	208	60	Reel sayı	2
Glass	Çok değişkenli	214	9	Reel sayı	7
Haberman	Çok değişkenli	306	3	Tam sayı	2
E.coli	Çok değişkenli	336	7	Reel sayı	8
Bupa	Çok değişkenli	345	6	Tam sayı, reel sayı	2
Ionosphere	Çok değişkenli	351	33	Tam sayı, reel sayı	2
Breast-cancer	Çok değişkenli	699	9	Tam sayı	2
Transfusion	Çok değişkenli	748	4	Reel sayı	2
Vehicle	Çok değişkenli	846	18	Tam sayı	4

Çizelge 3.1'de görüldüğü üzere kullanılan veri kümelerinin boyutu en az 150 ve en fazla 846'dır. Veri kümelerindeki özellik sayısı ise 3 ila 60 arasında değişmektedir. Kullanılan veri kümeleri, tam sayı ve/veya reel sayıları içeren nümerik türde veri kümeleridir. Veri kümelerindeki sınıf sayısı ise 2 ila 8 arasında değişmektedir. Deneysel analiz aşamasında, farklı sınıf sayısına sahip veri kümelerinin kullanılması önerilen yöntemlerin hem iki sınıflı ve hem de çok sınıflı sınıflandırma problemi üzerinde test edilmesini sağlamaktadır.

Kullanılan veri kümelerinden Breast Cancer Wisconsin (Original) veri kümesinin bir özelliğinde %2,29 oranında eksik veri bulunmaktadır. Diğer veri kümelerinde eksik veri bulunmamaktadır. Breast-cancer veri kümesi kullanılmadan önce bir ön işlemden geçirilerek eksik veri tamamlanmıştır. İlgili veri kümesi, nümerik türde olduğu için eksik verinin tamamlanması işleminde ortalama atama (mean imputation) yöntemi uygulanmıştır.

3.2. Makine Öğrenmesi Algoritmaları

Makine öğrenmesi kapsamında ele alınan sınıflandırma ve kümeleme problemleri için literatürde çeşitli algoritmalar bulunmaktadır. Bu kısımda, bu algoritmalardan tez kapsamında yararlanılan ve/veya önerilen yöntemlerin karşılaştırmalı analizinde kullanılan bazı kümeleme ve sınıflandırma algoritmaları hakkında detaylı bilgi verilmiştir.

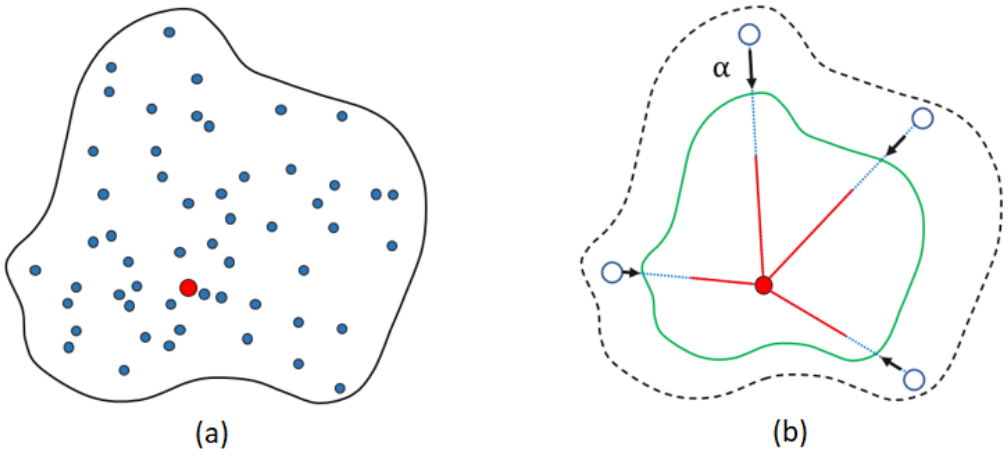
3.2.1. CURE Kümeleme Algoritması

Temsilcileri Kullanarak Kümeleme (Clustering Using Representatives, CURE) algoritması, Guha vd. (1998) tarafından önerilmiş hiyerarşik bir kümeleme yöntemidir. Guha vd. (1998) çalışmalarında, geleneksel kümeleme yöntemlerinin benzer boyutlarda ve küresel şekillerdeki kümeleri tercih ettiğine veya aykırı değerlere karşı oldukça hassas olduğuna dikkat çekmiştir. Bu doğrultuda, aykırı değerlere karşı daha dayanıklı ve küresel olmayan şekiller ile boyut olarak geniş varyanslara sahip kümeleri tanımlayan CURE kümeleme algoritmasını geliştirmişlerdir. Ayrıca bu algoritmada, büyük veri tabanları için rastgele örnekleme (random sampling) ve parçalama (partitioning) işlemleri uygulanmaktadır.

CURE algoritmasının genel diyagramı Şekil 3.1'de verilmiştir.

Şekil 3.1. CURE algoritması genel diyagramı (Guha vd., 2001)

Bu yöntemde ilk olarak bir kümedeki iyi saçılmış noktalar için sabit bir c sayısı belirlenmektedir. Saçılmış noktalar, kümenin şeklini ve kapsamını yakalamaktadır. Sonrasında seçilen saçılmış noktalar, kümenin merkezine doğru bir α katsayısı (fraction) ile daraltılmaktadır (Şekil 3.2). Daraltmadan sonraki bu saçılmış noktalar, temsili noktalar olarak kullanılmaktadır. CURE hiyerarşik kümeleme algoritmasının her bir adımında, temsili noktaları en yakın olan küme çiftleri birleştirilmektedir.



Şekil 3.2. Temsili noktaların daraltılması: (a) başlangıç örnekleri, (b) saçılmış noktaların merkeze doğru α katsayısı ile daraltılması

$S \subset \mathbb{R}^n$ girdi veri kümesini göstersin ve bu kümenin örnek sayısının $m = |S|$ olduğunu varsayalım. Bu durumda, CURE algoritmasının başlangıcında Denklem 3.1'de belirtilen şekilde m adet küme olacaktır.

$$\mathcal{C}^{(0)} = \left\{C_1^{(0)}, C_2^{(0)}, \dots, C_m^{(0)}\right\} = \left\{\{x_1\}, \{x_2\}, \dots, \{x_m\}\right\} \tag{3.1}$$

Burada $\mathcal{C}^{(0)}$ başlangıçtaki kümeler topluluğunu göstermektedir.

Bu ilk kümeleme, önceden belirlenmiş k sayıda küme elde edilene kadar en yakın iki kümenin birleştirilmesi ve her kümenin temsili noktalarının belirlenmesi şeklinde ilerleyecektir.

Algoritmanın önemli bir adımı, mevcut küme topluluğundaki her bir küme çifti arasındaki mesafenin hesaplanmasıdır. Mevcut aşamada, $\mathcal{C} = \{C_1, C_2, \dots, C_t\}$ şeklinde t adet küme olduğunu varsayarsak kümeler arasındaki uzaklık Denklem 3.2'de belirtildiği şekilde olacaktır.

$$d_{ij} = dist(C_i, C_j), 1 \leq i, j \leq t \tag{3.2}$$

Bu uzaklıklar, mevcut kümelemedeki en yakın küme çiftini bulmak için kullanılır.

$1 \leq i \leq t$ olmak üzere her C_i kümesi için en yakın küme Denklem 3.3 ile ifade edilsin.

$$d_i^* = \min_{j:j \neq i}\{d_{ij}\} \tag{3.3}$$

O halde en yakın küme çifti Denklem 3.4 tarafından verilir.

$$d^* = min\{d_1^*, d_2^*, \dots, d_t^*\} \tag{3.4}$$

Yani en yakın küme çifti, Denklem 3.5 ve 3.6 bulunarak tanımlanacaktır.

$$l = arg\min_j\{d_1^*, d_2^*, \dots, d_t^*\} \tag{3.5}$$

$$u = arg\min_{j:j \neq l}\{d_{lj}\} \tag{3.6}$$

O halde, mevcut kümelemedeki en yakın küme çifti C_u (u ile gösterilir) ve C_v ($u.closest$ ile gösterilir) olarak tanımlanır.

Sonrasında, bu iki küme birleştirilecek ve yeni temsili noktalar belirlenecektir. $C_{uv} = C_u \cup C_v$ olsun ve birleştirilmiş kümenin C_{uv} (w ile gösterilsin) merkezi $\bar{x}_{uv}$ ile gösterilsin. c adet temsili noktanın belirleneceğini varsayarsak; C_{uv} kümesinden C_{uv} kümesinin merkezine ($\bar{x}_{uv}$) en uzak nokta, ilk saçılmış nokta (scattered point) olarak seçilecektir. Bu nokta s_1 ile gösterilirse, o zaman Denklem 3.7 ile s_1 belirlenir.

$$s_1 = arg \max_{x \in C_{uv}} \{dist(x, \bar{x}_{uv})\} \tag{3.7}$$

Bundan sonra, C_{uv} kümesinden, daha önce belirlenen saçılmış noktalara en uzak olan bir nokta seçilir. Eğer seçilmiş olan h adet saçılmış nokta varsa, bu önceden seçilmiş noktaların kümesini $D = \{s_1, \dots, s_h\}, (h < c)$ ile gösterelim. O halde, bir sonraki saçılmış nokta Denklem 3.8'in bulunmasıyla seçilir.

$$s_{h+1} = arg \max_{s \in C_{uv}, s \notin D} \{dist(s, D)\} \tag{3.8}$$

Bu sürece, c tane saçılmış nokta seçilene kadar devam edilir.

CURE algoritmasının temel adımları şunlardır:

Cluster (S,k)

Girdi: n örnek içeren girdi veri kümesi, S

Çıktı: k adet küme

Her bir u kümesi için $u.rep$, $u.mean$ ve $u.closest$ tutulmaktadır:

$u.rep$: kümenin c adet temsili noktasının dizisi

$u.mean$: kümedeki noktaların ortalaması

$u.closest$: u' ya en yakın olan küme

Adım1. Başlangıçta, her u kümesi için $u.rep$ temsili noktalar dizisi yalnızca kümedeki noktayı içerir. Bu nedenle, bu adımda, tüm girdi veri noktaları k-d ağacı (k-d tree) olan T'ye eklenir.

Adım2. Her bir girdi noktasını ayrı bir küme olarak ele al, her u kümesi için $u.closest$ hesapla ve ardından her bir kümeyi Q yığınına (heap) ekle. (Yığında, kümeler u ve $u.closest$ arasındaki uzaklığın artan sırasına göre düzenlenir.)

Adım3. While $size(Q) > k$ (sadece k adet küme kalana kadar)

Adım4. Q yığınının en üstündeki u kümesi, u ve $u.closest$ kümelerinin en yakın küme çifti olduğu kümedir. Q'nun en tepesindeki u elemanını çıkar ve u'yu Q'dan sil.

Adım5. $v = u.closest$ olarak ayarla. v'yi Q'dan sil.

Adım6. En yakın küme çifti olan u ve v kümelerini birleştir ve bu birleşimden oluşan yeni küme (w) için yeni temsili noktaları hesapla.

Adım7. u ve v'yi T'den sil. w'yi T'ye ekle.

Adım8. Birleştirilmiş w kümesi için, temsili noktaların kümesi değişebileceğinden, bu kümenin diğer tüm kümelere olan mesafesini hesapla ve w kümesine en yakın olan kümeyi $w.closest$ olarak ayarla.

Adım9. Q'da yer alan farklı bir x kümesi için, $x.closest$ değişebilir ve x'in Q'da yeniden konumlandırılması gerekebilir. Q'daki her bir x kümesi için, $x.closest$'ı güncelle ve x'i Q'da yeniden konumlandır.

Adım10. w'yi Q'ya ekle.

Adım11. Verilen koşul sağlanana kadar Adım 4-10'u tekrarla.

CURE, her bir kümeyi çoklu temsili noktalar ile temsil ettiği için rastgele (arbitrary) şekil ve boyutlardaki kümeleri bulabilmektedir. Bununla birlikte, temsili noktaların merkeze doğru daraltılması yöntemin gürültü ve aykırı değerler ile alakalı problemlerden kaçınmasını sağlamaktadır (Karypis vd., 1999).

CURE algoritmasının kümeleme aşamasında uygulanan prosedür ise aşağıda verilmiştir (Guha vd., 2001).

```
procedure cluster (S, k)
begin
1.    T := build_kd_tree(S)
2.    Q := build_heap(S)
3.    while size(Q) > k do {
4.        u := extract_min(Q)
5.        v := u.closest
6.        delete(Q, v)
7.        w := merge(u, v)
8.        delete_rep(T, u); delete_rep(T, v); insert_rep(T, w)
9.        w.closest := x     /* x, Q'da rastgele bir kümedir. */
10.       for each x ∈ Q do {
11.           if (dist(w, x) < dist(w, w.closest))
12.               w.closest := x
13.           if x.closest is either u or v {
14.               if dist(x, x.closest) < dist(x, w)
15.                   x.closest := closest_cluster(T, x, dist(x, w))
16.               else
17.                   x.closest := w
18.               relocate(Q, x)
19.           }
20.           else if dist(x, x.closest) > dist(x, w) {
21.               x.closest := w
22.               relocate(Q, w)
23.           }
24.       }
25.       insert(Q, w)
26.   }
end
```

Bu prosedürde dist(p, q), bir çift nokta (p, q) arasındaki uzaklığı belirtmektedir. Bu durumda iki küme (u ve v) arasındaki uzaklık Denklem 3.9'da verildiği gibi ifade edilir.

$$dist(u, v) = \min_{p,q} \{dist(p, q) \mid p \in u.rep, q \in v.rep\} \tag{3.9}$$

Guha vd. (1998), uzaklık için Manhattan (L_1) veya Öklid (L_2) gibi L_p metriklerinden herhangi birinin kullanılabileceğini belirtmişler ve CURE algoritmasının testlerinde Öklid uzaklığını kullanmışlardır. Uygun uzaklık fonksiyonun belirlenmesi çoğu makine öğrenmesi algoritması için önemlidir (Xiang vd., 2008). Literatürde, özellik uzayında iki nokta arasındaki uzaklığı ölçmek için çeşitli uzaklık fonksiyonları

bulunmaktadır. Bu uzaklık fonksiyonlarından en yaygın kullanılan Öklid uzaklığıdır (Hu vd., 2016). Bu nedenle, önerilen RP-SVM yönteminin kümeleme aşamasında noktalar arası uzaklığın hesaplanmasında Öklid uzaklığı kullanılmıştır.

3.2.2. K-ortalamalar SVM (K-means SVM, KMSVM)

K-ortalamalar, iyi bilinen kümeleme problemini çözen en popüler denetimsiz öğrenme algoritmalarından biridir (Sinaga ve Yang, 2020). $O = \{O_1, O_2, \dots, O_n\}$, K adet kümeden oluşan bir dizi, $C = \{C_i,\ i = 1, \dots, k\}$, halinde kümelenecek olan n adet veri örneğinden oluşan bir küme olsun. K-ortalamalar kümelemenin amacı, Denklem 3.10'da tanımlandığı gibi k kümenin tamamı üzerindeki karesel hata (squared error) toplamını en aza indirmektir (Xie vd., 2019).

$$J(C) = \sum_{i=1}^{k} \sum_{O_l \in C_i} (O_l - Z_i)^2 \tag{3.10}$$

Denklem 3.10'da C_i, Z_i, O_l ve k sırasıyla i. kümeyi, i. kümenin ağırlık merkezini, i. kümeye ait veri örneklerini ve toplam küme sayısını temsil eder.

K-ortalamalar prosedürü aşağıdaki gibi özetlenebilir (Han vd., 2011).

Algoritma: K-ortalamalar kümeleme algoritması (Her bir küme merkezinin, kümedeki nesnelerin ortalama değeri ile temsil edildiği bölümleme için k-ortalamalar algoritması)

Girdi:

k: kümelerin sayısı

D: n adet nesne içeren veri kümesi

Çıktı: k adet küme içeren dizi

Yöntem:

1. İlk küme merkezleri olarak D veri kümesinden rastgele k adet nesne seç.
2. Tekrar et.
3. Kümedeki nesnelerin ortalama değerine dayalı olarak her bir nesneyi en benzer olduğu kümeye yeniden ata.

4. Küme ortalamalarını güncelle, yani her bir küme için nesnelerin ortalama değerini hesapla.
5. Değişiklik olmayana kadar Adım 2-6 tekrar et.

K-ortalamalar SVM, Wang vd. (2005) tarafından K-ortalamalar kümeleme yöntemi ile SVM yönteminin birleştirilmesiyle oluşturulmuş sınıflandırma algoritmasıdır. Bu algoritmada, k-ortalamalar kümeleme yöntemi tüm veri kümesine uygulanmaktadır. Bu işlem belirli bir sıkıştırma oranına (compression rate, CR) göre yapılmaktadır. Sıkıştırma oranı Denklem 3.11'de verilen şekilde belirlenmektedir.

$$Sıkıştırma\ Oranı\ (CR) = Orijinal\ veri\ sayısı/Küme\ sayısı \qquad (3.11)$$

K-ortalamalar kümeleme yöntemi ile tespit edilen k adet küme merkezine yeni sınıf etiketlerinin atanması çoğunluk oylaması (majority voting) yöntemiyle yapılmaktadır.

KMSVM algoritmasının, SVM sınıflandırıcılarından çok daha az destek vektörü ve daha yüksek yanıt hızı ile sınıflandırıcılar oluşturması mümkündür.

KMSVM algoritmasının adımları şunlardır:

Adım1. 3 girdi parametresi seçilir: çekirdek parametresi γ, ceza parametresi C ve sıkıştırma oranı CR.

Adım2. Orijinal veri üzerinde K-ortalamalar kümeleme algoritması çalıştırılır ve tüm küme merkezleri sınıflandırıcılar oluşturmak için sıkıştırılmış veri olarak kabul edilir.

Adım3. Sıkıştırılmış veri üzerinden SVM sınıflandırıcılar oluşturulur.

Adım4. Girdi parametreleri belirlenir (Parametrelerin belirlenmesinde ilgili çalışmada önerilen, test doğruluğu ve yanıt zamanı arasındaki dengeye dayalı sezgisel arama stratejisi kullanılmaktadır).

Adım5. Girdi parametrelerinin yeni kombinasyonlarını test etmek için 1. adıma geri dönülür ve eğer test doğruluğu ve yanıt zamanına göre kombinasyon kabul edilebilir ise durulur.

Adım6. KMSVM sınıflandırıcılar, Denklem 3.12'de verildiği gibi temsil edilir.

$$f(x) = sgn\left(\sum_{i=1}^{n} a_i K(x, x_i) + b\right) \tag{3.12}$$

Lee vd. (2007) ise kümeleme aşamasında sınıf bilgisinden yararlanarak KMSVM yönteminin modifiye edilmiş bir versiyonunu geliştirmişlerdir. Bu şekilde kümeleme aşamasında sınıf bilgisinden yararlanılarak baskın olan yani daha çok örneğe sahip olan sınıfın etkisi azaltılmıştır. Araştırmacılar, bunun dengesiz sınıf problemine sahip veri kümelerinde faydalı olduğunu göstermişlerdir. Bu yöntemde, her bir sınıfa atanan kümelerin sayısı, orijinal eğitim kümesinde ilgili sınıfın boyutu ile orantılıdır.

3.2.3. *K*-En Yakın Komşuluk (*K*-Nearest Neighbor, *K*NN)

Denetimli öğrenme yöntemlerden birisi Tembel Öğrenme (Lazy Learning) olarak da adlandırılan Örnek Tabanlı Öğrenmedir (Instance Based Learning) (Soofi ve Awan, 2017). Örnek tabanlı öğrenme yönteminin klasik örneği *K*-En Yakın Komşuluk sınıflandırma algoritmasıdır (Aggarwal, 2014). En Yakın Komşuluk (Nearest Neighbor, NN) sınıflandırıcı, verilen bir test örneğine uzaklık fonksiyonuna göre eğitim kümesindeki en yakın komşusunun sınıfını atar. K bir tamsayı ve $K \geq 1$ olmak üzere *K*-En Yakın Komşuluk sınıflandırıcı, en yakın komşuluk sınıflandırıcının bir genellemesidir (Viswanath ve Sarma, 2011). *K*NN algoritması, her yeni örneği *K*-en yakın komşusu arasından çoğunluk etiketine göre sınıflandırarak çalışır. Makul bir mesafe (tipik olarak Öklid mesafesi) varsa ve eğitim kümesindeki veri noktalarının sayısı çok büyük değilse, K-En Yakın Komşuluk yönteminin iyi çalıştığı bilinmektedir (Lemm vd., 2011).

En yakın komşuluk kuralının naif uygulamasında, önceden sınıflandırılmış tüm veri noktalarının saklanması ve sonrasında her bir örnek noktasını sınıflandırmak için her bir saklanan nokta ile karşılaştırılması gerekmektedir (Angiulli ve Narvaez, 2018).

$U \subset R^d$ bir girdi veri kümesi ve $x_i, x_j \in U$ bu veri kümesinde yer alan iki örnek olmak üzere x_i ve x_j örnekleri arasındaki Öklid uzaklığı Denklem 3.13'te belirtilen şekilde ifade edilebilir.

$$d_E(x_i, x_j) = \sqrt{\sum_{a \in B} w_a (f(x_i, a) - f(x_j, a))^2} \tag{3.13}$$

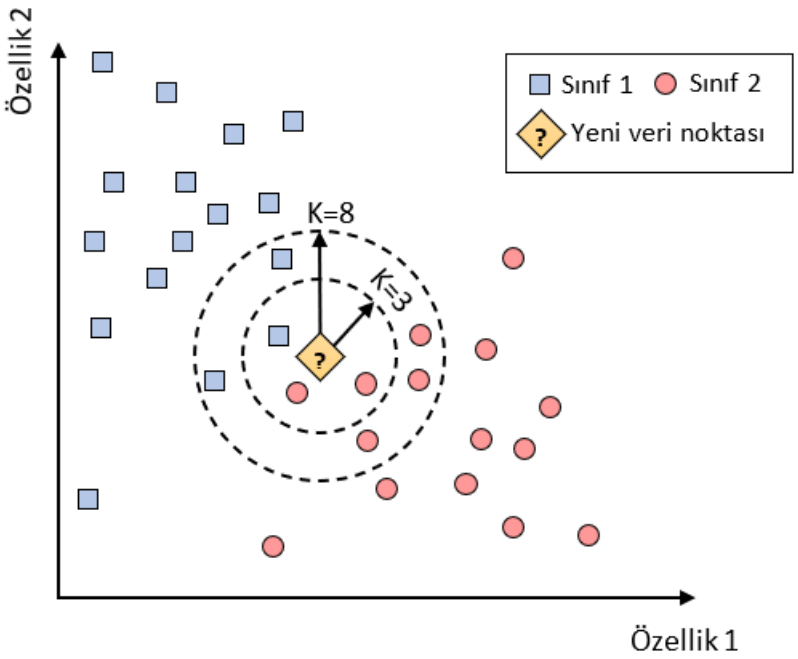
Denklem 3.13'te w_a, $a \in B$ özelliğinin ağırlığını ifade etmektedir. Verilen bir x_j örneği için karar fonksiyonu ise Denklem 3.14'te verildiği şekilde tanımlanır.

$$y(x_j, c_i) = \begin{cases} 1, & eğer\ x_j, \text{örneği}\ c_i\ sınıfında\ ise \\ 0, & diğer\ durumda \end{cases} \tag{3.14}$$

Bu durumda, $N_k(x)$, x'in K-en yakın komşularının kümesi olmak üzere yeni bir x örneğinin sınıf etiketi Denklem 3.15 ile belirlenir.

$$y(x) = arg\max_i \sum_{x_j \in N_k(x)} y(x_j, c_i) \tag{3.15}$$

Şekil 3.3'te iki boyutlu bir düzlemde basit bir *K*NN örneği gösterilmiştir. Örnekte, iki sınıflı bir veri kümesinde yeni bir veri noktasının *K*=3 ve *K*=8 için en yakın komşuları belirtilmiştir. Çoğunluk oylaması yöntemi uygulandığında her iki durum için de yeni örneğin Sınıf 2'ye dahil olduğu görülmektedir.



Şekil 3.3. K-en yakın komşuluk algoritması örnek gösterimi

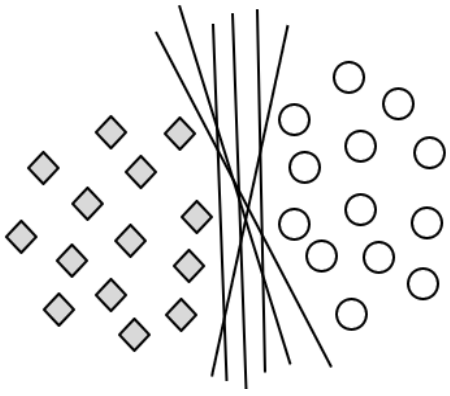
3.2.4. Destek Vektör Makineleri (Support Vector Machines, SVMs)

Destek Vektör Makineleri, istatistiksel öğrenme teorisine dayanan denetimli öğrenme algoritmalarıdır (Kavzoglu ve Colkesen, 2009). Sınıflandırma ve regresyon uygulamalarında yaygın olarak kullanılmaktadır. Sınıflandırma işleminde hata en aza

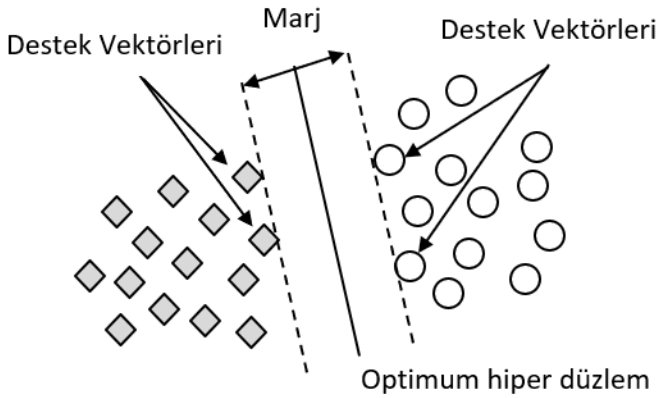
indirilmeye ve eş zamanlı olarak iki sınıf arasındaki uzaklık/marj en yüksek hale getirilmeye çalışılır. Bu şekilde iyi bir genelleme sağlayarak aşırı öğrenmenin oluşmasını önlenmek istenmektedir.

Çoğu öğrenme algoritmasının en büyük kısıtlarından birisi, çekirdek fonksiyonunun $k(x_n, x_m)$, eğitim noktalarındaki tüm olası x_n ve x_m çiftlerini değerlendirmesinin gerekmesidir. Bu işlem, eğitim esnasında hesaplama açısından uygun değildir ve yeni veri noktaları için tahminler yapıldığında yoğun hesaplama zamanına neden olabilmektedir (Bishop, 2006). Ancak seyrek (sparse) çözümler sağlayan çekirdek tabanlı algoritmalarda yeni girdiler için tahminler, sadece eğitim veri noktalarının bir alt kümesiyle değerlendirilen çekirdek fonksiyonuna bağlıdır. Bu alt kümeler, seyrek çekirdek tabanlı makineler olan Destek Vektör Makinelerinde, *destek vektörleri* ve İlgililik Vektör Makinelerinde (Relevance Vector Machines, RVMs), *ilgililik vektörleri* olarak adlandırılan veri noktalarını içermektedir.

Sınıflandırma problemini ele alalım. SVM, sınıflandırma probleminde iki sınıfı optimum şekilde ayıran hiper düzlemi bulmayı hedeflemektedir. Doğrusal olarak ayrılabilen veri kümesinde, iki sınıfı ayırabilen birden fazla hiper düzlem mevcuttur (Şekil 3.4). Ancak, marj olarak adlandırılan iki sınıf arasındaki uzaklık değerini en büyük yapan tek bir düzlem vardır ve bu düzlem optimum hiper düzlemdir (Şekil 3.5). Şekil 3.5'te görüldüğü üzere, marjı sınırlandıran veri noktaları destek vektörleri olarak adlandırılmaktadır. Destek vektörleri, eğitim örneklerinin bir alt kümesidir ve SVM yönteminde karar fonksiyonunu belirlemektedir.



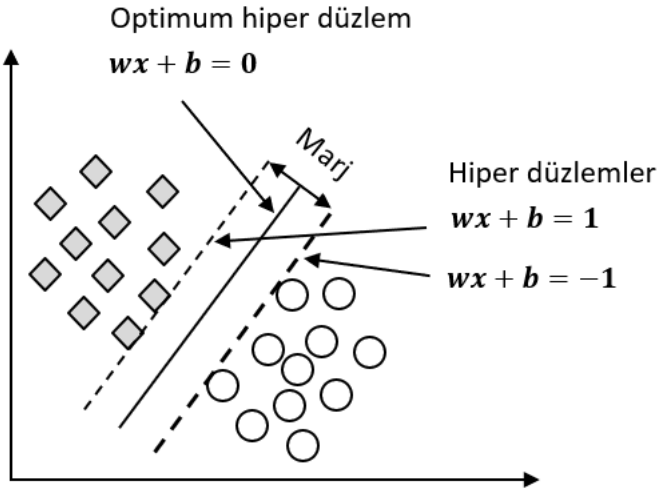
Şekil 3.4. Lineer olarak ayrılabilen veride hiper düzlemler



Şekil 3.5. Optimum hiper düzlem ve destek vektörleri

Bir veri kümesi içerisinden, n tane örnek içeren eğitim noktaları $(\boldsymbol{x}_i, y_i)$, $i = 1, 2, 3, \ldots n$ ele alınsın. Her bir girdi $\boldsymbol{x}_i$, D adet özelliğe sahiptir ve $\boldsymbol{x} \in R^D$, D-boyutlu uzayda tanımlıdır. İki sınıf için $y_i \in \{-1, +1\}$ sınıf etiketlerini ifade eder.

Şekil 3.6'da görüldüğü üzere optimum hiper düzlem $w\boldsymbol{x}_i + b = 0$ ile tanımlanabilir. Bu denklemde; b hiper düzlemin orijinden uzaklığını, w hiper düzlemin uzaydaki oryantasyonunu ve x hiper düzlem üzerinde yer alan bir noktayı ifade etmektedir.



Şekil 3.6. Lineer ayrılabilen ikili sınıflandırma için optimum hiper düzlem

İki sınıf için hiper düzlem tanımlamaları ise Denklem 3.16 ve Denklem 3.17'de verilen eşitsizlikler ile yapılmaktadır.

$$\mathbf{w} \cdot \mathbf{x}_i + b \geq +1 \qquad (3.16)$$

(tüm $y = +1$ değerleri için)

$$\mathbf{w} \cdot \mathbf{x}_i + b \leq -1 \tag{3.17}$$

(tüm $y = -1$ değerleri için)

Denklem 3.16 ve Denklem 3.17'de verilen eşitsizlikler, tek bir eşitsizlik olarak Denklem 3.18 şeklinde tanımlanabilir.

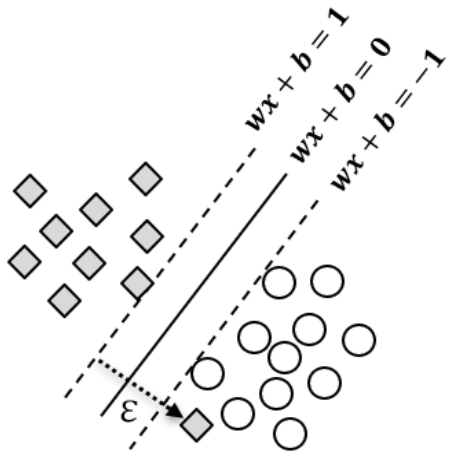
$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \geq 0 \tag{3.18}$$

Sınıfları optimum şekilde ayıran bir düzlem, iki sınıfı birbirinden ayırmakta ve sınıfların birbirine en yakın noktaları arasındaki mesafeyi yani marjı maksimize etmektedir. Marj, $2/\|w\|$ değerine eşittir ve optimum hiper düzlem, $\|w\|^2$ değeri minimize edilerek bulunmaktadır. Dolayısıyla, optimum hiper düzlemin bulunması, Denklem 3.19'da verilen optimizasyon probleminin belirtilen kısıtlar dikkate alınarak çözülmesini gerektirmektedir.

$$\underset{\mathbf{w},b}{\text{minimize}} \frac{1}{2}\|\mathbf{w}\|^2 \tag{3.19}$$

Kısıtlar: $y_i\big((\mathbf{w} \cdot \mathbf{x}_i) + b\big) \geq 1$ ve $y_i \in \{+1, -1\}$

Sınıflandırma işleminde, veri kümesi her zaman lineer olarak ayrılamayabilir (Şekil 3.7). Bu durumda, verinin lineer bir fonksiyon ile girdi uzayında sınıflandırılması mümkün olmamaktadır. Tamamen lineer olarak ayrılamayan veriler için SVM modelleri, lineer olmayan yüzeylerde çalışacak şekilde genelleştirilmiştir. Yumuşak marjlı SVM (Soft Margin SVM) olarak adlandırılan bu yaklaşımda genel formülasyona, C ceza parametresi ve ε_i (slack variable) değişkeni eklenmiştir.



Şekil 3.7. Lineer olarak ayrılamayan veri kümesi

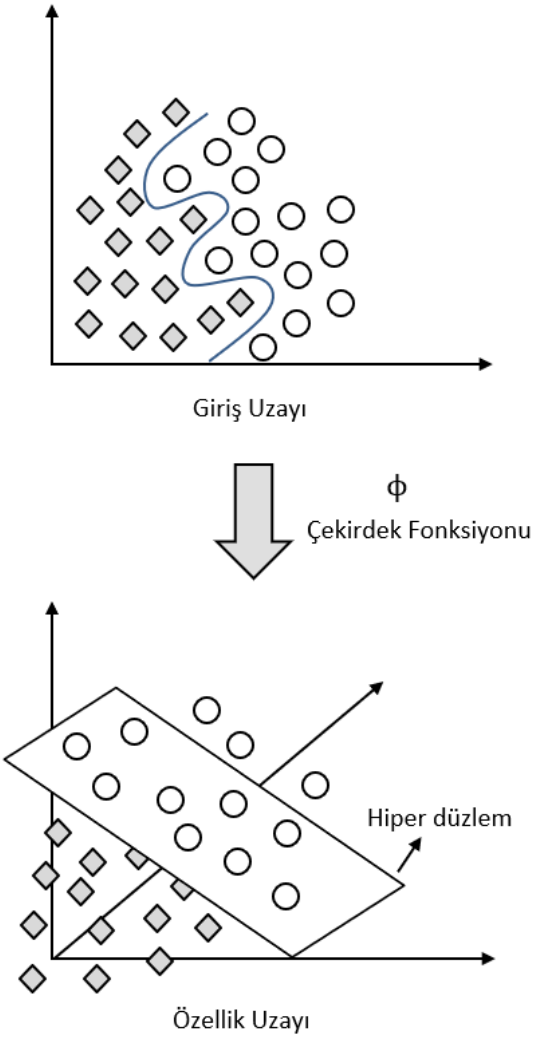
Genelleştirilen durumda optimizasyon problemi Denklem 3.20'de verilen hale gelmektedir.

$$minimize\left[\frac{\|w\|^2}{2} + C\sum_{i=1}^{r} \varepsilon_i\right] \tag{3.20}$$

Kısıtlar: $y_i\left((\mathbf{wx}_\mathrm{i}) + b\right) \geq 1 - \varepsilon_i$, $\varepsilon_i \geq 0$ ve $i = 1,2,\ldots,n$

C: ceza parametresi (marjın maksimize edilmesi ve hatanın minimize edilmesi arasında denge sağlamaktadır) ve ε_i: Yanlış sınıflandırılan noktaların optimum hiper düzleme uzaklığı.

Bazı durumlarda ise lineer eşitliklerle hiper düzleme karar vermek mümkün olmayabilir. Böyle durumlarda veri, bazı lineer olmayan haritalama (çekirdek) fonksiyonları (ϕ) yardımıyla yüksek boyutlu bir uzayda haritalanabilir (Şekil 3.8).



Şekil 3.8. Lineer olarak ayrılamayan veri kümesinin ayrılması

Bir x giriş verisi yüksek boyutlu uzayda ϕ(x) ile ifade edilebilir. Ancak ϕ(x).ϕ(x) işleminin hesaplama yoğunluğu fazladır. Bu yoğunluğu azaltmak için çekirdek fonksiyonları kullanılmaktadır. Bu durumda sınıflandırma karar fonksiyonu, Denklem 3.21'da verildiği gibi olmaktadır.

$$f(x) = sign\left(\sum_{i}^{n} \alpha_i y_i K(x, x_i) + b\right) \tag{3.21}$$

$K(x, x_i)$: Çekirdek fonksiyonu

Destek Vektör Makinelerinde yaygın olarak kullanılan çekirdek fonksiyonları genel olarak lineer, polinomiyal (polynomial), radyal tabanlı fonksiyon (Radial Based Function, RBF) ve sigmoid çekirdekler olmak üzere 4 gruba ayrılır (Kavzoglu ve Colkesen, 2009). Çizelge 3.2'de yaygın kullanılan bu çekirdek fonksiyonları formülasyonları ile birlikte verilmiştir.

Çizelge 3.2. Çekirdek fonksiyonları (Dimitriadou vd., 2009)

#	**Çekirdek Fonksiyonu**	**Formül**
1	Lineer	$K(x, x_i) = x' x_i$
2	Radyal	$K(x, x_i) = exp(-\gamma \lvert x - v \rvert^2)$
3	Sigmoid	$K(x, x_i) = tanh(\gamma x_i' v + r)$
4	Polinomiyal	$K(x, x_i) = (\gamma x' x_i + r)^d$
γ:gamma r:coefficient d:derece		

Destek Vektör Makineleri, temelde iki sınıflı sınıflandırma için geliştirilmiştir. Ancak, çok sınıflı sınıflandırma problemlerinde SVM kullanımını sağlayan çeşitli algoritmalar da önerilmiştir. Bu algoritmalardan bazıları Bire Karşı Hepsi (One-Against-Rest), Bire Karşı Bir (One-Against-One), Yönlü Çevrimsiz Çizge (Directed Acyclic Graph), Adaptif Çevrimsiz Çizge (Adaptive Acyclic Graph) ve Hata Düzelten Çıktı Kodlaması (Error Correcting Output Code) olarak belirtilmiştir (Aburomman ve Reaz, 2017).

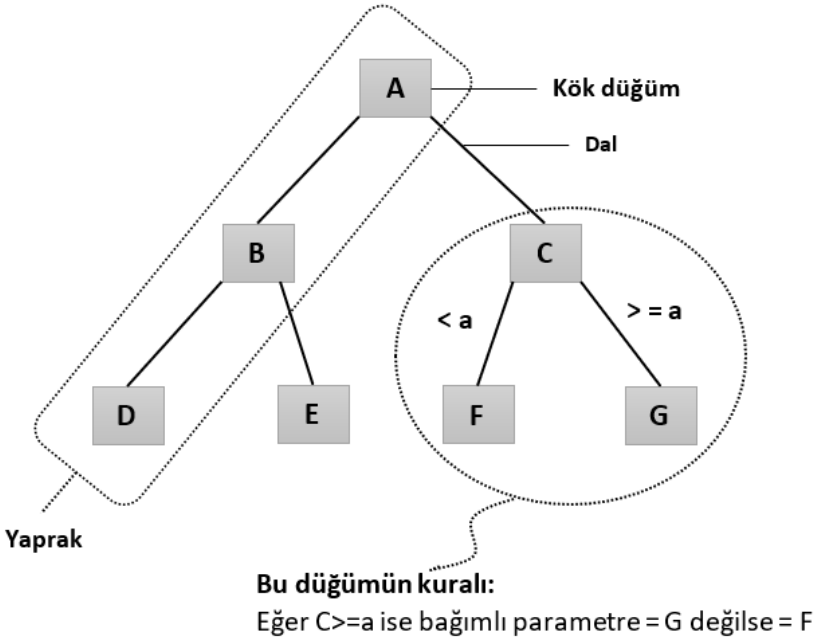
Başarılı sınıflandırma yöntemlerinden birisi olan Destek Vektör Makinelerinin birtakım dezavantajları da bulunmaktadır. Bu dezavantajlardan bazıları şunlardır (Tipping, 2000; Tipping 2001):

- Tahminler/çıktılar olasılıksal değildir. Sınıflandırma işleminde tahminlerdeki belirsizliği saptayabilmek için koşullu dağılımın $p(t|\mathbf{w})$ tahmin edilmesi istenmektedir.
- Nispeten seyrek olmasına rağmen, gerekli destek vektörlerin sayısı tipik olarak eğitim kümesinin boyutu ile doğrusal olarak büyüdüğü için SVM'ler temel fonksiyonları gereksiz yere liberal olarak kullanır.
- Ödünleşim (trade-off) parametresinin (C) tahmin edilmesi gerekmektedir. Bu genellikle hem veri hem de hesaplama açısından israflı olan çapraz doğrulama işlemine neden olmaktadır.
- Kullanılan çekirdek fonksiyonunun Mercer şartını karşılaması gerekmektedir.

SVM için belirtilen bu dezavantajları ortadan kaldırmak için Tipping (2000) tarafından İlgililik Vektör Makineleri önerilmiştir. RVM belirtilen dezavantajları ortadan kaldırmasının yanında daha az çekirdek fonksiyonu kullanarak çalışma verimini artırmaktadır.

3.2.5. Sınıflandırma ve Regresyon Ağaçları (Classification and Regression Trees, CART)

Veri madenciliğinde yaygın olarak kullanılan parametrik olmayan yöntemlerden biri de karar ağaçları analizleridir. Karar ağaçlarının temel amacı, bağımlı değişken(ler)i tahmin etmek için bağımsız değişken(ler) kullanarak tahmine dayalı bir model oluşturmaktır. Karar ağaçları, doğal ağaçları taklit eden kök, yaprak, dal ve düğümlerden oluşur (Khandelwal, 2017). Bir karar ağacının yapısı Şekil 3.9'da verilmiştir. Karar ağacı; sınıflandırma, tanıma, karar verme ve tahmin amaçları için kullanılabilecek basit ve anlaşılır bir yapıya sahiptir. Karmaşık bir yapıya sahip olan ve kara kutu olarak bilinen Yapay Sinir Ağları gibi makine öğrenmesi algoritmalarına kıyasla, bir tahmin şeması için genellikle basitliği, açıklanabilirliği ve düşük hesaplama maliyetleri nedeniyle karar ağacı uygulanması tercih edilir. Karar ağacı çıkarımı için ana algoritmalardan birisi Sınıflandırma ve Regresyon Ağaçlarıdır (Pitombo vd., 2017).



Şekil 3.9. Bir karar ağacının yapısı (Salimi vd., 2018)

CART, model oluşturmak ve değerlendirmek için eğitim örneklerini kullanan bir denetimli öğrenme sınıflandırma algoritmasıdır. Tüm gözlemlerine önceden sınıf atanmış bir dizi geçmiş verinin öğrenme örneğini kullanır. CART, gruplama için tüm girdi özelliklerinden en iyi özelliğin bulunması ve özelliğin aralığında optimal bir ayırma eşiğinin belirlenmesi olmak üzere iki öğeden oluşmaktadır (Çelik ve Yılmaz, 2018). CART yaklaşımı, 1984 yılında Breiman vd. (1984) tarafından karar ağacı oluşturmak için sunulmuştur. CART tarafından üretilen karar ağaçları, her bir karar düğümü için tam olarak iki dal içerir yani tam olarak/tamamen ikilidir. CART, eğitim veri kümesindeki kayıtları özyinelemeli olarak hedef özellik için benzer değerlere sahip kayıtlar içeren alt kümelere böler (Larose, 2015).

CART, hem Sınıflandırma hem de Regresyon Karar Ağaçları oluşturmak için kullanılabilir. Bir veri kümesini iki sınıfa ayırmak için karar ağacı kullanıldığında, model bir sınıflandırma ağacıdır, ancak hedef değişken sayısal veya sürekli olduğunda tahmin görevi regresyon olmaktadır. Bir sınıflandırma ağacı kullanıldığında amaç, eldeki veri kümesini, verilerin homojenliğini kullanarak iki parçaya bölmektir. Hangi özelliğin bölüneceğine ve nereden bölüneceğine karar vermek için CART, entropi veya Gini indeksi gibi safsızlık (impurity) ölçülerine dayanır. Regresyon ağaçlarında çıktı özelliğinin (hedef değişkenin) sınıfları yoktur ve buradaki amaç bir kaydın ait olduğu sınıfı tahmin etmek değil, hedef değişkeninin değerini tahmin etmektir (Zacharis, 2018).

Klasik sınıflandırma ve regresyon modelleriyle karşılaştırıldığında, CART yönteminin avantajları şunlardır (Zhang vd., 2018):

- Daha az veri hazırlığı: veri normalizasyonu gerekmez.
- Sürekli ve ayrık verileri aynı anda ele alabilmektedir.
- Çoklu sınıflandırma problemlerini ele alabilir.
- Tahmin süreci, Sinir Ağı ve SVM gibi diğer öğrenme modellerinin aksine Boolean mantığı kullanılarak kolayca açıklanabilir.

CART yaklaşımının bir diğer avantajı da aykırı değerlerle başa çıkma kolaylığıdır. Aykırı değerler, Temel Bileşen Analizi (Principal Component Analysis) ve Lineer Regresyon gibi bazı istatistiksel modellerin sonuçları üzerinde olumsuz bir etkiye sahip olabilir. Ancak CART algoritması, gürültülü verileri ayrı bir düğüme ayırarak kolayca işleyecektir. Ayrıca bu sorunun üstesinden gelmek için CART algoritmasında aykırı değerleri ortadan kaldırmak veya ortalama, mod veya en yakın komşu yöntemlerini kullanmak için bir prosedür uygulanabilir (Khandelwal, 2017).

CART, açıklama gücü ve varyans açısından bir modelde veya ilişkide özellikle hangi faktörlerin önemli olduğunu istatistiksel olarak gösterebilir. Bu işlem, bazı tanıdık regresyon teknikleriyle matematiksel olarak aynıdır, ancak veri istatistiksel analizde iyi olmayanlar tarafından kolayca yorumlanacak şekilde sunulur. Bu şekilde, CART, verilerdeki değişkenlerin ilişkisinin gelişmiş bir anlık görüntüsünü sunar ve bilgilendirici bir model inşa edilmesinde veya önemli ilişkilerin nihai görselleştirilmesinde ilk adım olarak kullanılabilir (Morgan, 2014).

CART modeli aşağıdaki üç ana adımdan oluşur (Zhang vd., 2018):

Adım1. CART başlatma: Eğitim veri kümesine dayalı bir karar ağacı oluşturulur.

Adım2. CART Budama ve Optimizasyon: Regresyon ağacı; ağacın maksimum derinliği, yaprak düğümünün minimum örnek sayısı ve düğümün minimum safsızlığı gibi bazı kısıtlamalara göre budanır. Model, farklı parametrelerin (*max_depth*: ağacın maksimum derinliği, *min_samples_leaf*: yaprak düğümlerinin minimum örnek sayısı, *min_impurity_split*: düğümlerin minimum safsızlığı) birleşimi ile en iyi genellemeye sahip olur. Farklı parametrelerin her bir kombinasyonu için farklı CART modelleri oluşmaktadır.

Adım3. CART Tahmini: Eğitilen modele test kümesi verilir ve tahmin yapılır.

3.3. Makine Öğrenmesi Yöntemlerinin Değerlendirilmesi

Makine öğrenmesi modelleri, verinin anlaşılmasını artırmaya/geliştirmeye yönelik modeller ve tahmin için modeller olmak üzere farklı roller için oluşturulabilir. Rollerinden bağımsız olarak, model oluştururken en kritik adımlardan biri modellerin değerlendirilmesidir. Değerlendirme, modelleme faaliyetinin belirtilen amacına ulaşılıp ulaşılmadığını belirler; farklı modelleme yaklaşımlarını karşılaştırmaya ve gelecekteki araştırmaları yönlendirmeye olanak tanır (Reich ve Barai, 1999). Sınıflandırma modellerini değerlendirme görevi, örneğin gerçek sınıflandırmasına karşılık gelen modeli kullanarak önerilen sınıflandırmanın derecesini ölçmektir. Gözlem yöntemine bağlı olarak, modelin performansını değerlendirmek için farklı ölçütler bulunmaktadır. Sorunun özelliklerine ve uygulama yollarına bağlı olarak en uygun ölçütler seçilir (Novaković vd., 2017).

Değerlendirme yöntemi, sınıflandırma performansını değerlendirmede ve sınıflandırıcı modellemesine rehberlik etmede önemli bir faktördür. Sınıflandırma sürecinin eğitim aşaması, doğrulama aşaması ve test aşaması olmak üzere üç ana aşaması vardır. Model, girdi kalıpları kullanılarak eğitilir ve bu aşamaya eğitim aşaması denir. Bu girdi modellerine, modeli eğitmek için kullanılan eğitim verileri denir. Bu aşamada, sınıflandırma modelinin parametreleri ayarlanır. Eğitim hatası, eğitilen modelin eğitim verilerine ne kadar iyi uyduğunu ölçer. Ancak, eğitilen model, eğitim aşamasında kullanılan aynı verilere uyduğundan, eğitim hatası her zaman test hatasından ve doğrulama hatasından daha küçüktür. Bir öğrenme algoritmasının amacı, daha önce görülmeyen verilerin sınıf etiketlerini tahmin etmek için eğitim verilerinden öğrenmektir; bu test aşamasıdır (Tharwat, 2020).

Bir karışıklık matrisi (confusion matrix), gerçek değerlerin bilindiği bir dizi test verisi üzerinde bir sınıflandırma modelinin (veya sınıflandırıcının) performansını tanımlar (Paper, 2020). İki sınıflı sınıflandırma problemi için oluşturulan bir karışıklık matrisi Çizelge 3.3'te verilmiştir.

Çizelge 3.3. İki sınıflı sınıflandırma problemi için karışıklık matrisi (Markoulidakis vd., 2021)

		Tahmin Edilen Sınıf	
		Pozitif	Negatif
Gerçek Sınıf	Pozitif	TP	FN
	Negatif	FP	TN

Tabloda belirtilen ifadeler şunlardır:

- TP (True Positive): Gerçekte pozitif olan ve pozitif olarak tahmin edilen noktaların sayısıdır.
- TN (True Negative): Gerçekte negatif olan ve negatif olarak tahmin edilen noktaların sayısıdır.
- FP (False Positive): Gerçekte negatif olan ve pozitif olarak tahmin edilen noktaların sayısıdır.
- FN (False Negative): Gerçekte pozitif olan ve negatif olarak tahmin edilen noktaların sayısıdır.

Çok sınıflı sınıflandırma problemi için karışıklık matrisi ise Çizelge 3.4'te verilmiştir.

Çizelge 3.4. Çok sınıflı sınıflandırma problemi karışıklık matrisi (Markoulidakis vd., 2021)

		Tahmin Edilen Sınıf			
		C_1	C_2	…	C_N
Gerçek Sınıf	C_1	$C_{1.1}$	FP	…	$C_{1.N}$
	C_2	FN	TP	…	FN
	…	…	…	…	…
	C_N	$C_{N.1}$	TP	…	$C_{N.N}$

Doğruluk (Accuracy, Acc): Bir sınıflandırıcıyı değerlendirirken kullanılan en yaygın ve en basit ölçüdür. Sadece bir modelin doğru tahminlerinin derecesi (veya tersine, yanlış sınıflandırma hatalarının yüzdesi) olarak tanımlanır (Ferri vd., 2009).

Doğruluk değeri Denklem 3.22'de verilen formülasyon kullanılarak hesaplanmaktadır (Hossin ve Sulaiman, 2015).

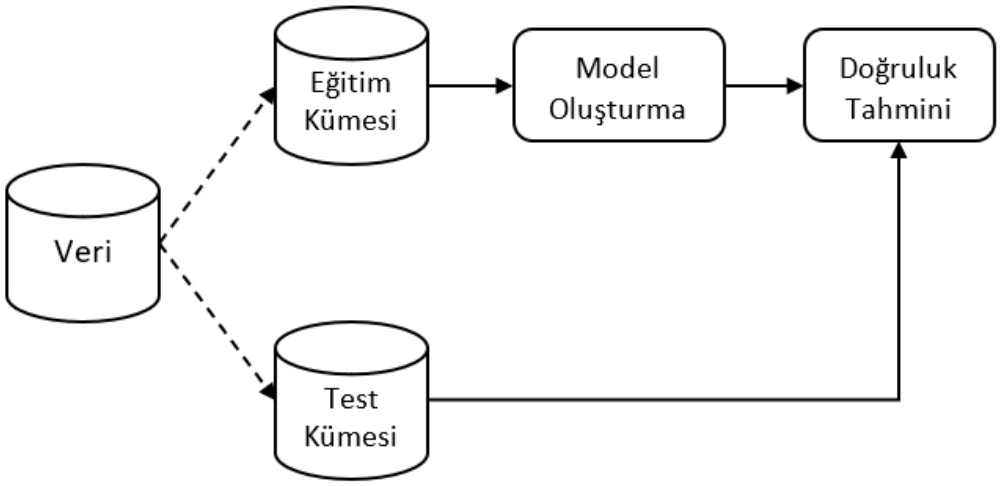
$$Doğruluk(Acc) = \frac{TP + TN}{TP + FP + TN + FN} \tag{3.22}$$

Bu çalışma kapsamında, önerilen yöntemlerin değerlendirilmesi aşamasında doğruluk ölçütü kullanılmıştır. Bunun yanı sıra, yöntemlerin eğitim aşamasında kullandıkları eğitim kümesi boyutu açısından değerlendirme yapılmıştır. Ayrıca, standart SVM ve SVM temelli olarak önerilen yaklaşımlarda model oluşturmada kullanılan destek vektörlerinin sayıları dikkate alınmıştır.

3.3.1. Dışarıda Tutma (Holdout) Yöntemi

Dışarıda tutma yönteminde, etiketlenmiş veriler, eğitim ve test verilerine karşılık gelen rastgele iki ayrık kümeye bölünür. Tipik olarak çoğunluk (örneğin üçte ikisi veya dörtte üçü) eğitim verisi olarak kullanılır ve geriye kalan test verisi olarak kullanılır. Yaklaşım, nihai bir tahmin sağlamak için birden fazla örnekle birkaç kez tekrarlanabilir (Aggarwal, 2015). Rastgele örnekleme (random sampling) ise dışarıda tutma yönteminin k kez tekrarlanan bir varyasyonudur. Genel doğruluk tahmini, her yinelemeden elde edilen doğrulukların ortalaması olarak alınır (Han vd., 2011). Verilerin 2/3'ünün eğitim seti ve kalan 1/3'ünün test seti olarak atanması yaygındır (Kohavi, 1995).

Dışarıda tutma yöntemi ile doğruluk tahminine ait diyagram Şekil 3.10'da verilmiştir.



Şekil 3.10. Holdout yöntemiyle doğruluk tahmini (Han vd., 2011)

3.3.2. Çapraz Doğrulama (Cross Validation)

Çapraz doğrulama (cross validation, CV), modellerin gerçek tahmin hatasını değerlendirmek ve model parametrelerini ayarlamak için en yaygın kullanılan veri yeniden örnekleme (data resampling) yöntemlerinden biridir. Pratikte genellikle 10-kat katmanlı çapraz doğrulama uygulanır (Berrar, 2019).

Çapraz doğrulamanın aşağıda belirtilen iki muhtemel amacı bulunmaktadır (Refaeilzadeh vd., 2009):

- Tek bir algoritma kullanarak öğrenilen modelin performansını mevcut verilerden tahmin etmek. Başka bir deyişle, bir algoritmanın genellenebilirliğini ölçmek.
- İki veya daha fazla farklı algoritmanın performansını karşılaştırmak ve mevcut veriler için en iyi algoritmayı bulmak veya alternatif olarak parametreli bir modelin iki veya daha fazla varyantının performansını karşılaştırmak.

Çeşitli CV şemaları, örnek verileri bölme biçimlerine göre farklılık göstermektedir. En yaygın kullanılan yöntem ise k-katlı çapraz doğrulamadır (Lemm vd., 2011).

- **k-Katlı Çapraz Doğrulama (k-Fold Cross Validation)**

Model seçimi için k-katlı çapraz doğrulamanın sözde kodu aşağıda verilmiştir (Shalev-Shwartz ve Ben-David, 2014). Prosedür girdi olarak bir S eğitim kümesini, Θ muhtemel parametre değerleri dizisini, katların sayısını temsil eden bir k tamsayısını ve girdi olarak bir eğitim kümesi ve bir $\theta \in \Theta$ parametresi alan bir A öğrenme algoritmasını almaktadır. Prosedür, en iyi parametre ve bu parametre ile tüm veri kümesi üzerinde eğitilen hipotez çıktısını verir.

Model Seçimi için k-Fold Cross Validation

girdi:

eğitim kümesi $S = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$

parametre değerleri dizisi Θ

öğrenme algoritması A

tamsayı k

S kümesini $S_1, S_2, \ldots, S_k$ şeklinde **bölümle**

foreach $\theta \in \Theta$

for $i = 1 \ldots k$

$$h_{i,\theta} = A(S \backslash S_i; \theta)$$

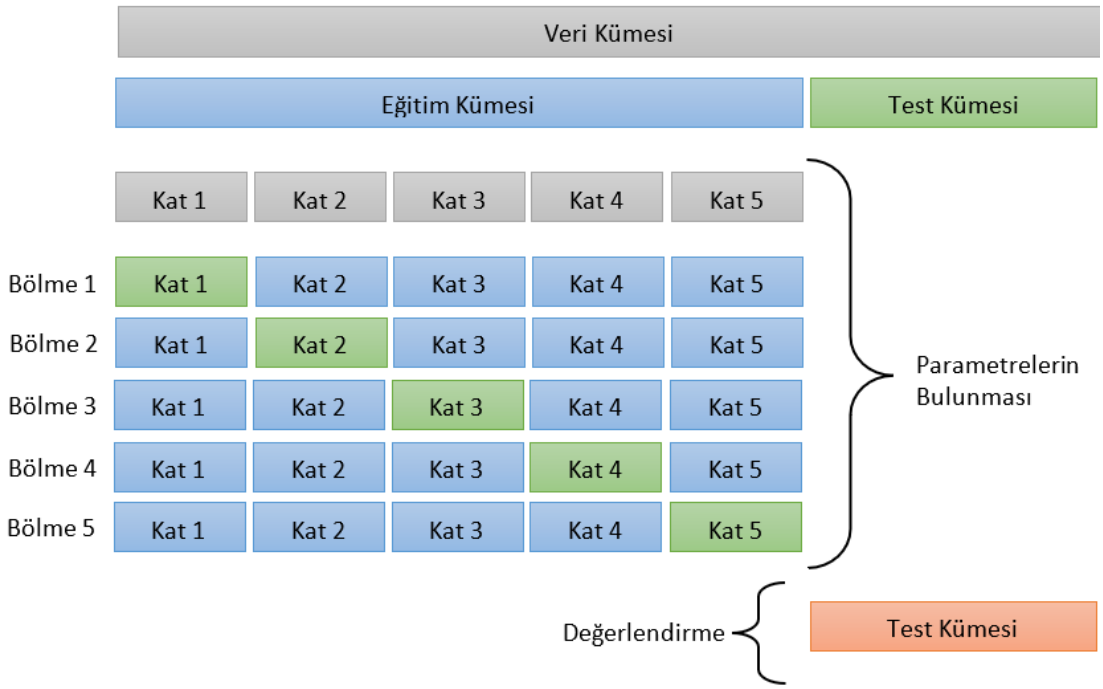
$$error(\theta) = \frac{1}{k} \sum_{i=1}^{k} L_{S_i}(h_{i,\theta})$$

çıktı

$$\theta^* = argmin_\theta [hata(\theta)]$$

$$h_{\theta^*} = A(S; \theta^*)$$

$k = 5$ için 5-katlı çapraz doğrulama işlemi Şekil 3.11'de verilmiştir.

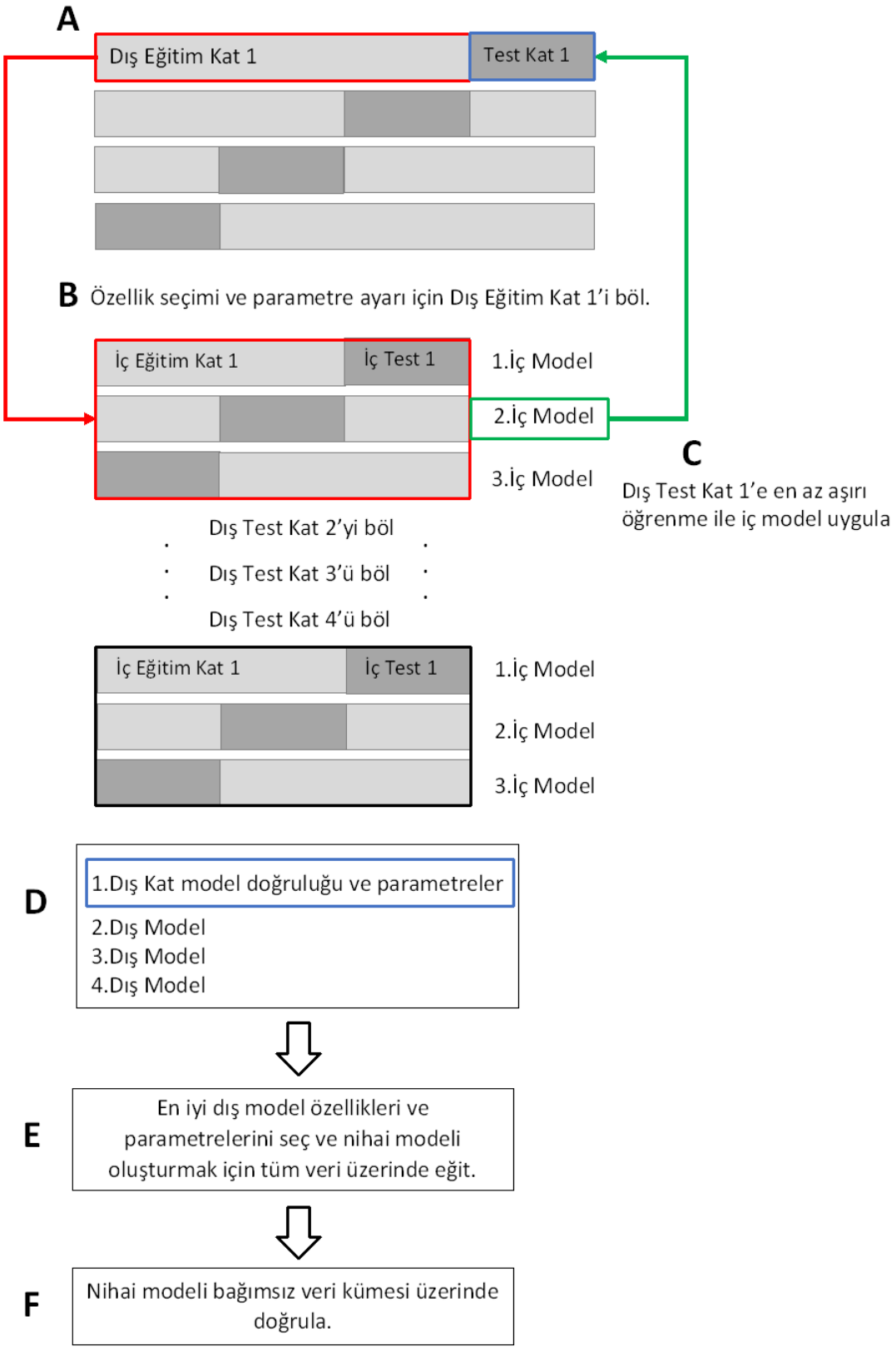


Şekil 3.11. 5-katlı çapraz doğrulama (Learn, 2017)

- **İç İçe Çapraz Doğrulama (Nested Cross Validation, nCV)**

İç içe çapraz doğrulama, optimal modeli seçmek için kullanılan performans tahminini sağlamak için bir dış çapraz doğrulama prosedürü (outer cross validation procedure) gerçekleştirilir. Dış çapraz doğrulamanın her katında, modelin hiper parametreleri, genelleme performansının bir iç çapraz doğrulama tahminini en aza indirmek için bağımsız olarak ayarlanır. Dış çapraz doğrulama, daha sonra, çapraz doğrulamaya dayalı hiper parametre ayarlama dahil olmak üzere, bir model uydurmak için bir yöntemin performansını esasen tahmin etmektedir. Bu, düz çapraz doğrulama prosedürünün getirdiği yanlılığı (bias) ortadan kaldırır çünkü dış çapraz doğrulamanın her yinelemesindeki test verileri, modelin performansını hiçbir şekilde optimize etmek için kullanılmamıştır ve bu nedenle, en iyi modeli seçmek için daha güvenilir bir kriter sağlayabilir. Bununla birlikte, iç içe çapraz doğrulamanın hesaplama maliyeti (computational expense) önemli ölçüde daha yüksektir (Wainer ve Cawley, 2021).

Standart iç içe çapraz doğrulama işlemi Şekil 3.12'de verilmiştir.



Şekil 3.12. Standart nCV (Parvandeh vd., 2020)

Şekil 3.12'de verilen adımlar aşağıda açıklanmıştır.

A. Veri, eğitim ve test verisi çiftleri olarak dış katlara ayrılır (bu gösterimde dört dış kat bulunuyor). Ardından, kırmızı ile gösterilen Dış Eğitim Kat 1 ile başlayarak her bir dış eğitim katı için aşağıdaki adımlar uygulanır.

B. Özellik seçimi ve grid arama ile olası hiper parametre ayarı için dış eğitim katı iç katlara ayrılır.

C. Dış test katını (Test Kat 1) test etmek için iç katlarda en az aşırı öğrenmeye dayalı özellikler ve parametreler içeren en iyi eğitim modeli kullanılır. Şekilde en iyi model, yeşil çerçeve ile işaretlenmiş olan 2. iç modeldir.

D. Özellikler ve test doğrulukları dahil olmak üzere bu dış kat için en iyi model kaydedilir. Kalan dış katlar için B–D aralığındaki adımlar tekrarlanır.

E. Minimum aşırı öğrenmeye dayalı olarak özellikleri ile birlikte en iyi dış model seçilir. Nihai modeli oluşturmak için tüm veriler üzerinde eğitilir.

F. Nihai model, bağımsız veriler üzerinde doğrulanır.

3.4. Tanımlamalar ve Temel Notasyon

Kümeleme ve sınıflandırma problemlerinde, örnekler özellikler ile ifade edildiği için bilgi sistemi olarak da adlandırılan bir *veri tablosu* kullanılarak tanımlama yapılabilir. Veri tablosu, 4'lü tanımlama grubunu $< U, A, V, f >$ içermektedir. U, nesnelerin sonlu bir kümesi ve $A = \{a_1, a_2, \dots, a_m\}$ özelliklerin sonlu bir kümesidir. Burada bir $\alpha \in A$ özelliğinin alanı V_a ile ifade edilir ve $V = \bigcup_{a \in A} V_a$ olmaktadır. f fonksiyonu, bir toplam fonksiyonudur (total function) öyle ki her $\alpha \in A$ ve $x \in U$ için $f(x, \alpha) \in V_a$'dır. Bu fonksiyon bilgi fonksiyonu (information function) olarak adlandırılır. Eğer özellikler kümesi *şart* (condition) ($C \neq \phi$) özellikleri ve *karar* (decision) ($D \neq \phi$) özellikleri olarak bölünürse veri tablosu karar tablosu (*decision table*) olarak adlandırılır. Veri tablosu, bilgi sistemi ve ilgili notasyonlar için detaylı bilgi Greco vd. (2001)'de yer almaktadır.

Belirtilen tanımlamalar ile bir veri tablosunun ifadesi Çizelge 3.5'te verilmiştir.

Çizelge 3.5. Veri tablosu

U/A	**Nümerik Özellikler**				**Kategorik Özellikler**			**Sınıf**
	a_1	a_2	...	a_k	a_{k+1}	...	a_{m-1}	a_m
x_1	$f(x_1, a_1)$	$f(x_1, a_2)$	...	$f(x_1, a_k)$	$f(x_1, a_{k+1})$	...	$f(x_1, a_{m-1})$	$f(x_1, a_m)$
x_2	$f(x_2, a_1)$	$f(x_2, a_2)$	...	$f(x_2, a_k)$	$f(x_2, a_{k+1})$	...	$f(x_2, a_{m-1})$	$f(x_2, a_m)$
x_3	$f(x_3, a_1)$	$f(x_3, a_2)$	...	$f(x_3, a_k)$	$f(x_3, a_{k+1})$	...	$f(x_3, a_{m-1})$	$f(x_3, a_m)$
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
x_N	$f(x_N, a_1)$	$f(x_N, a_2)$	...	$f(x_N, a_k)$	$f(x_N, a_{k+1})$	...	$f(x_N, a_{m-1})$	$f(x_N, a_m)$

Önerilen kümeleme algoritmasındaki önemli bir kavram, veri kümesinin herhangi iki örneği arasındaki benzerliğin tanımıdır. Aşağıdaki tanımlamalar, iki örnek arasındaki benzerlikleri tanımlamak için kullanılır.

Tanımlama 1. Bir $a \in A$ özelliği açısından iki örnek $x, y \in U$ arasındaki benzerlik (similarity), a nümerik bir özellik ise Denklem 3.23'te belirtilen şeklinde tanımlanır.

$$sim_a(x, y) = 1 - \frac{|f(x, a) - f(y, a)|}{max(a) - min(a)} \tag{3.23}$$

Eğer a kategorik bir özellik ise iki örnek arasındaki benzerlik Denklem 3.24'te verilen şeklinde tanımlanır.

$$sim_a(x, y) = f(x) = \begin{cases} \frac{1}{|V_a|}, & eğer\ f(x, a) \neq f(y, a)\ ise \\ 1, & diğer\ durumlarda \end{cases} \tag{3.24}$$

Tanımlama 2. $a \in B$ özelliğinin ağırlık değeri w_a olmak üzere, bir özellik kümesi $B \subseteq A$ açısından iki örnek $x, y \in U$ arasındaki benzerlik değeri Denklem 3.25 şeklinde tanımlanır.

$$sim_B(x, y) = \sum_{a \in B} w_a sim_a(x, y) \tag{3.25}$$

Bu tanımlamada belirtilen w_a ağırlığı farklı şekillerde tanımlanabilir. Örneğin, Xiang vd. (2008) kümeleme algoritmasında ağırlıkları belirlemek için entropi (entropy) kullanmıştır. Bu çalışmada ise nümerik özelliklerin ağırlıklarının hesabı için Kayaalp ve Aslan (2014) tarafından tanıtılan yöntem kullanılmaktadır.

$D = \{d\}$ ve $|V_d| = t$ olsun ve her bir $x \in U$ örneğinin $V_d = \{c_1, c_2, \dots, c_t\}$'deki sınıflardan sade ve sadece bir tanesine ait olduğunu varsayalım. Verilen bir $a \in C$ özelliği için $C_i(a)$ ifadesi a özelliğinde i sınıfına $(1 \leq i \leq t)$ ait değerlerin kümesi olmak üzere Denklem 3.26 şeklinde ifade edelim.

$$A_i(a) = \left\{x \in U : \min\left(C_i(a)\right) \leq f(x,a) \leq \max\left(C_i(a)\right)\right\} \tag{3.26}$$

Bu durumda nümerik özelliğin $a \in C$ için ağırlığı w_a, Denklem 3.27 ve Denklem 3.28'deki ifadeler ile tanımlanmaktadır.

$$B_j(a) = A_j(a) - \bigcup_{\substack{k=1 \\ (k \neq j)}}^{t} A_k(a) \tag{3.27}$$

$$w_a = \frac{\bigcup_{i=1}^{t} |B_i(a)|}{|U|} \tag{3.28}$$

Önerilen kümeleme algoritmasında, aynı kümede olması gereken (must-link) ve aynı kümede olmaması gereken (cannot-link) kısıtlaması yoktur. Ancak bunun yerine veri kümesinin örnekleri arasındaki benzerlikler için bir eşik değeri (threshold) uygulanmaktadır. Bu şekilde eşik değeri kullanarak, veri kümesini, bir başlangıç çizge gösterimi ile temsil etmeye uygun olan benzerliklere sahip örneklerden oluşan bir alt küme elde edilmektedir. Bu temsil, bu alt kümeye bir kümeleme algoritması uygulanarak daha da geliştirilir. Bu nedenle, kaba küme teorisinde (rough set theory) temel bir kavram olan ayırt edilemezlik (indiscernibility) tanımına da ihtiyaç bulunmaktadır.

Tanımlama 3. Bir $B \subseteq A$ özellik kümesine göre t güven seviyesi (confidence level) ile ayırt edilemezlik ilişkisi (indiscernibility relation) IR_B, Denklem 3.29'da belirtilen şeklinde tanımlanmaktadır.

$$IR_B(t) = \{(x,y) \in U \times U : sim_B(x,y) \geq t\} \tag{3.29}$$

Burada t, benzerlik ilişkisi için bir eşik değeridir ve uygun t eşik değeri belirlenerek mutlaka aynı kümede olması gerekenler (must-link sets) tanımlanabilir.

Yukarıda tanımlandığı gibi ayırt edilemezlik ilişkisi, U üzerinde dönüşlü (reflexive) ve simetrik (symmetric) olan ancak geçişli olmayan (not transitive) ikili bir ilişki olduğu belirtilmelidir.

3.5. Önerilen Yöntemler

Bu bölümde tez çalışması kapsamında önerilen kümeleme ve sınıflandırma yöntemleri hakkında bilgi verilmektedir. Bu doğrultuda, önerilen yöntemler, benzerlik tabanlı yöntemler ve temsili noktalara dayalı yöntem olmak üzere iki ayrı başlıkta ele alınmıştır. Benzerliğe dayalı olarak geliştirilen yöntemler ve elde edilen bulgular Arslan vd. (2021) tarafından sunulan çalışmada yayınlanmıştır. Temsili noktalara dayalı yöntem ve bu yönteme ait bulgular ise Karabulut vd. (2021) tarafından sunulan çalışmada yer almaktadır.

3.5.1. Benzerliğe Dayalı Yöntemler

Kayaalp ve Arslan (2014) yaptıkları çalışmalarda veri kümesindeki doğal yapılanmaların önemine dikkat çekmiş ve ön çalışma niteliğinde geliştirmeye açık bir yaklaşım sunmuşlardır. Bu çalışmalardan temel alınarak, veri kümesindeki doğal yapılanmaların benzerlik temelli olarak tespit edilmesi ve bu yapılanmaların sınıflandırma sürecine katkısı araştırılmıştır. Bu doğrultuda benzerlik temelli bir kümeleme ve üç farklı sınıflandırma yöntemi önerilmiştir.

Benzerlik Tabanlı Doğal Kümeleme Yöntemi

Veri kümesindeki doğal yapılanmaların tespit edilmesi için nümerik özellikler arasındaki benzerliklere dayalı bir kümeleme yöntemi önerilmiştir. Bu yöntemde, kümeleri oluşturmak için, temel bileşenler (basic components) olarak adlandırılan kümenin bazı temel elemanları kullanılır. Bu temel bileşenler, çekirdek elemanlar (kernel elements) olarak adlandırdığımız, veri kümesinin en yakın iki örneğini içermektedir. Temel bileşenler, çekirdek elemanlara ek olarak iki ek elemana sahip olabilirler. Bu iki eleman, belirli bir eşik değerinin üzerinde benzerlik derecesi ile her iki çekirdek elemanına benzer olan bir dizi örnek arasından seçilmektedir. Bunlardan ilki, eğer varsa, çekirdekteki her iki elemana da en yakın olan örnektir. İkinci eleman

ise, eğer varsa, en yüksek sayıda benzerliklere sahip olan örnektir. Bu bileşenler, veri kümesinin diğer örnekleri veya diğer temel bileşenleri ile büyüyerek veri kümesinin yapısal bilgisini temsil eden ilk kümelemeleri oluşturacaktır. Bu nedenle temel bileşenler olarak adlandırılmışlarıdır.

Veri setindeki temel bileşenleri elde etme adımları şu şekilde özetlenebilir:

1. Verilen bir eşik değeri t için (örneğin $t = 0{,}90$), B özellik kümesine göre tüm benzer çiftleri bul; $IR_B(t) = \{(x, y) \in U \times U : sim_B(x, y) \geq t\}$.
2. $IR_B(t)$'deki çiftler arasından en benzer çifti bul; $\{x_0, y_0\}$. Küme çekirdeğini (kernel) oluştur; $Ker(C) = \{x_0, y_0\}$.
3. Küme çekirdeğindeki her bir elemana benzeyen örnekleri bul;
   a) x_0'a benzeyen örneklerin kümesini bul, bu küme R_{x_0} ile gösterilsin.
   b) y_0'a benzeyen örneklerin kümesini bul, bu küme R_{y_0} ile gösterilsin.
   c) Her iki elemana benzeyen örneklerin kümesini bul; $R_{x_0,y_0} = R_{x_0} \cap R_{y_0}$
4. Eğer varsa, R_{x_0,y_0} kümesinden çekirdeğe en yakın olan elemanı (x_{min}) bul ve bu elemanı çekirdeğe ekle: $C = \{x_0, y_0, x_{min}\}$.
5. B'ye göre $R_{x_0,y_0} \backslash \{x_{min}\}$'de yer alan tüm elemanların benzerliklerinin sayısını bul,
6. Eğer varsa, en yüksek sayıda benzerlikleri olan elemanı (x_{sim}) kümeye ekle:
   $C = \{x_0, y_0, x_{min}, x_{sim}\}$.
7. $\{x_0, y_0\}$ çifti ve geçerli bileşene atanan diğer örnekleri $IR_B(t)$'den kaldır.
8. Veri kümesindeki temel bileşenler elde edilene kadar adım 2-8'i tekrarla.

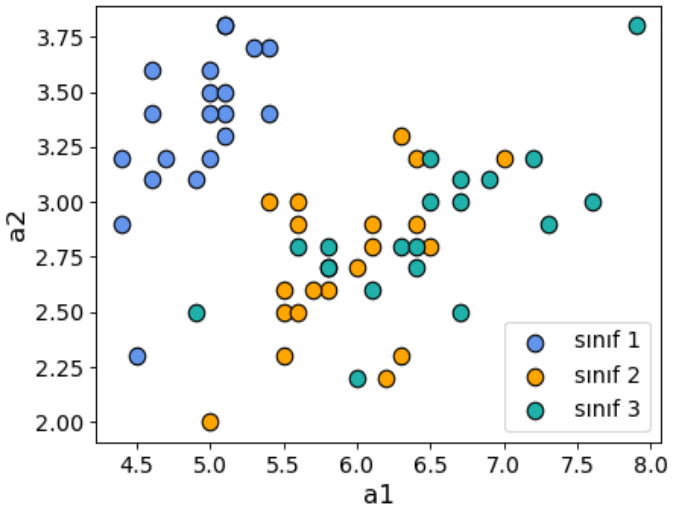
Bu algoritma belirtilen adımları uygulayarak, h temel bileşenlerin sayısı olmak üzere $\mathcal{C}_B = \{C_1, C_2, \ldots, C_h\}$ ile gösterilen sonlu sayıda temel bileşen elde ederek duracaktır.

Burada $\mathcal{C}_B$'deki her temel bileşen en az iki ve en fazla dört elemana sahip olacaktır. Dolayısıyla bir başlangıç kümelemesi oluşturmak için bu temel bileşenler, $IR_B(t)'$de kalan elemanlardan en benzer olan elemanları ekleyerek veya iki bileşeni en benzer elemanları ile birleştirerek büyüyecektir. Bu işlem ile veri kümesinin en uygun kümelenmesini bulmak için her kümeleme için entropi değerleri kullanılmaktadır. Bu şekilde, veri kümesinin karşılık gelen bölümü için sınıflara göre en yüksek bilgi

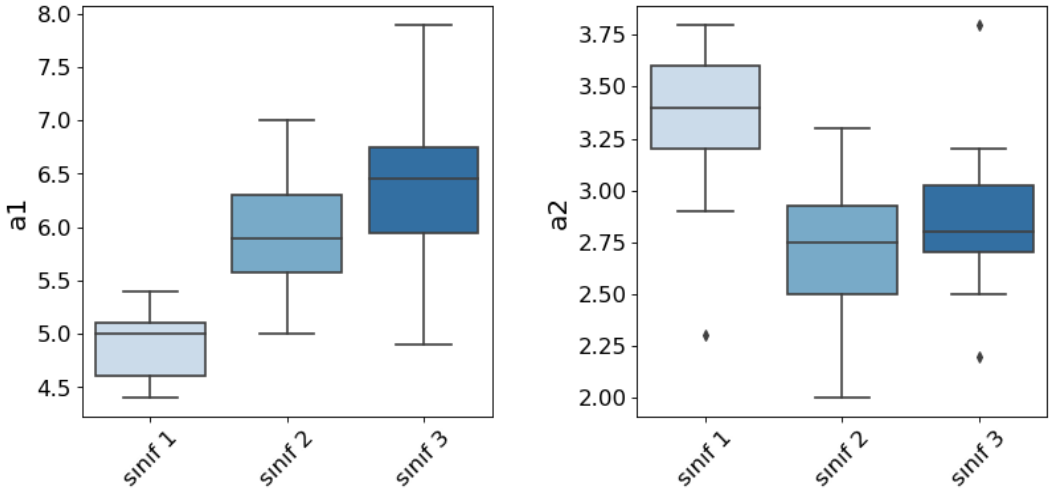
kazancına (information gain) sahip kümeleme elde edilecektir. Bu kümeleme algoritması, Benzerlik Tabanlı Doğal Kümeler (SNC) algoritması olarak adlandırılmıştır.

SNC Algoritması Örnek Uygulama:

SNC kümeleme algoritmasının adımlarını göstermek için iris veri kümesinden bir alt küme seçilmiştir. Sonuçları grafiksel olarak ifade edebilmek için bu alt kümede iris veri kümesinde yer alan iki özellik (a1 ve a2) kullanılmıştır. Bu alt kümeye ait saçılım grafiği Şekil 3.13'te ve kutu grafikleri Şekil 3.14'te verilmiştir.



Şekil 3.13. Alt kümeye ait saçılım grafiği



Şekil 3.14. Alt kümeye ait kutu grafikleri

Bu örnek için, özelliklerin standardize edilmiş ağırlıkları (standardized weights of the attributes) $w = (0{,}7059;\ 0{,}2941)$ olarak hesaplanmıştır. Şekil 3.14'teki kutu

grafiklerinden görüleceği üzere ilk özellik (a1), ikinci özelliğe (a2) kıyasla sınıf kategorilerini ayırmak için daha fazla bilgi içermektedir. Bu ağırlıkları kullanarak, her bir örnek çifti için benzerlikler hesaplanabilir. En benzer örneklerden bazıları Çizelge 3.6'da verilmiştir.

Çizelge 3.6. Benzer örnek çiftlerinden bazıları

#	i	j	$sim(x_i, x_j)$
1	55	57	1,0000000
2	9	20	1,0000000
3	54	60	0,9836601
4	53	58	0,9836601
5	49	57	0,9836601
6	49	55	0,9836601
7	36	58	0,9836601
8	35	57	0,9836601
9	35	55	0,9836601
10	31	39	0,9836601

Önerilen SNC algoritmasının adımları uygulanarak, bu örnek için temel bileşenler Çizelge 3.7'de verildiği gibi elde edilmiştir.

Çizelge 3.7. Alt küme için temel bileşenler

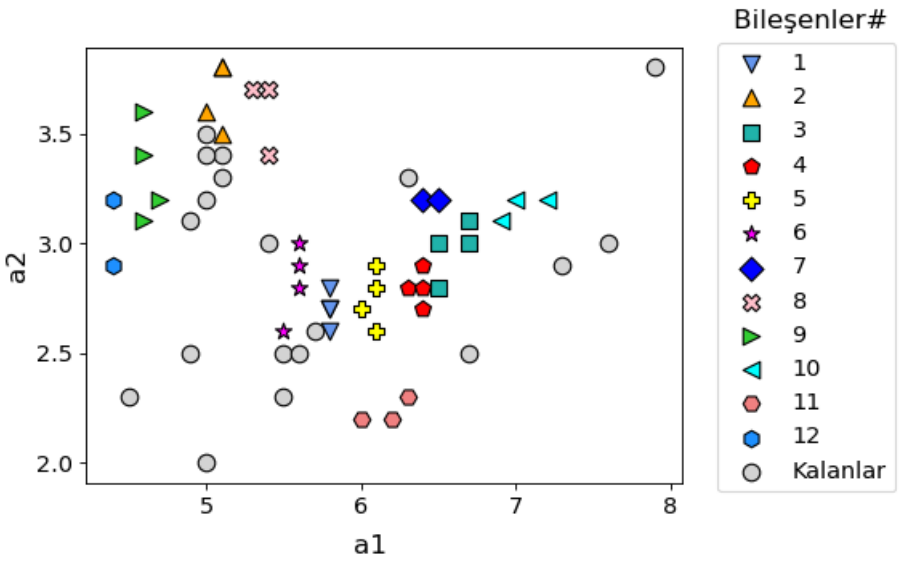
Bileşen#	***Elemanlar***			
1	55	57	49	35
2	9	20	17	14
3	54	60	43	25
4	53	58	36	41
5	31	39	47	30
6	29	48	37	24
7	38	56		
8	3	18	13	
9	5	15	10	2
10	28	44	42	
11	26	33	50	
12	4	11		

Temel bileşenler elde edildikten sonra Çizelge 3.8'de verildiği üzere nihai kümelenme elde edilir.

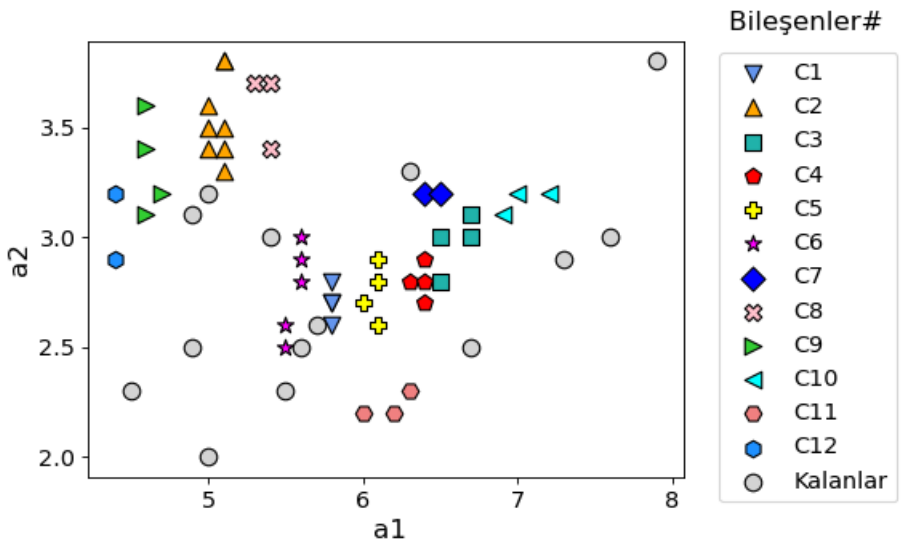
Çizelge 3.8. Alt küme için nihai doğal kümeler

Küme#	***Kümedeki elemanlar***							
1	55	57	49	35				
2	9	20	17	14	7	8	6	12
3	54	60	43	25				
4	53	58	36	41				
5	31	39	47	30				
6	29	48	37	24	21			
7	38	56						
8	3	18	13					
9	5	15	10	2				
10	28	44	42					
11	26	33	50					
12	4	11						

Alt küme için elde edilen temel bileşenler Şekil 3.15'te ve nihai kümeler Şekil 3.16'da saçılım grafiği ile gösterilmiştir.



Şekil 3.15. Alt küme için temel bileşenler



Şekil 3.16. Alt küme için nihai doğal kümeler

Benzerlik Tabanlı Sınıflandırma Yöntemleri

Kümelemenin sınıflandırmada bir ara adım olarak kullanımı, daha sağlam (robust) sınıflandırma algoritmaları geliştirmeyi sağlayabilir. Bu nedenle, girdi veri kümesinin küme temsilini elde etmek ve sonrasında bu temsili sınıflandırmada kullanmak için yöntemler araştırılmıştır. Önerilen SNC kümeleme algoritması kullanılarak belirtilen şekilde bir sınıflandırıcı geliştirilmesi üzerinde durulmuştur.

Önerilen sınıflandırıcının temel adımları şunlardır;

Adım1. Bir kümeleme algoritması yardımıyla girdi veri kümesinden doğal kümeler elde edilir.

Adım2. Veri kümesinin yapısını temsil etmek için doğal kümeler kullanılır.

Adım3. Sınıflandırma yöntemi uygulanır.

Bu adımlar doğrultusunda kümeleme tabanlı sınıflandırma için 3 temel yaklaşım geliştirilmiştir. Bu yaklaşımlar şunlardır:

- Doğal Kümeler Tabanlı En Benzer Örnekler (Natural Clusters-based Most Similar Instances, NC-MSI)
- Doğal Kümeler Tabanlı Destek Vektör Makinesi (Natural Clusters-based Support Vector Machine, NC-SVM)
- Doğal Kümeler Tabanlı Destek Vektör Makinesi - Sınırlar (Natural Clusters-based Support Vector Machine - Boundaries, NC-SVM-B)

NC-MSI Sınıflandırma Algoritması

Doğal Kümeler Tabanlı En Benzer Örnekler (Natural Clusters-based Most Similar Instances, NC-MSI) sınıflandırma yöntemi, daha önce tanıtılan SNC kümeleme algoritması üzerine en benzer örnek tabanlı bir sınıflandırma aşaması eklenerek oluşturulmuştur. Bu yaklaşımda, eğitim veri kümesine SNC kümeleme algoritması uygulandıktan sonra her bir test örneği, azaltılmış eğitim kümesindeki en benzer olduğu örneğe göre sınıflandırılmaktadır.

NC-MSI algoritmasında uygulanan genel adımalar şunlardır:

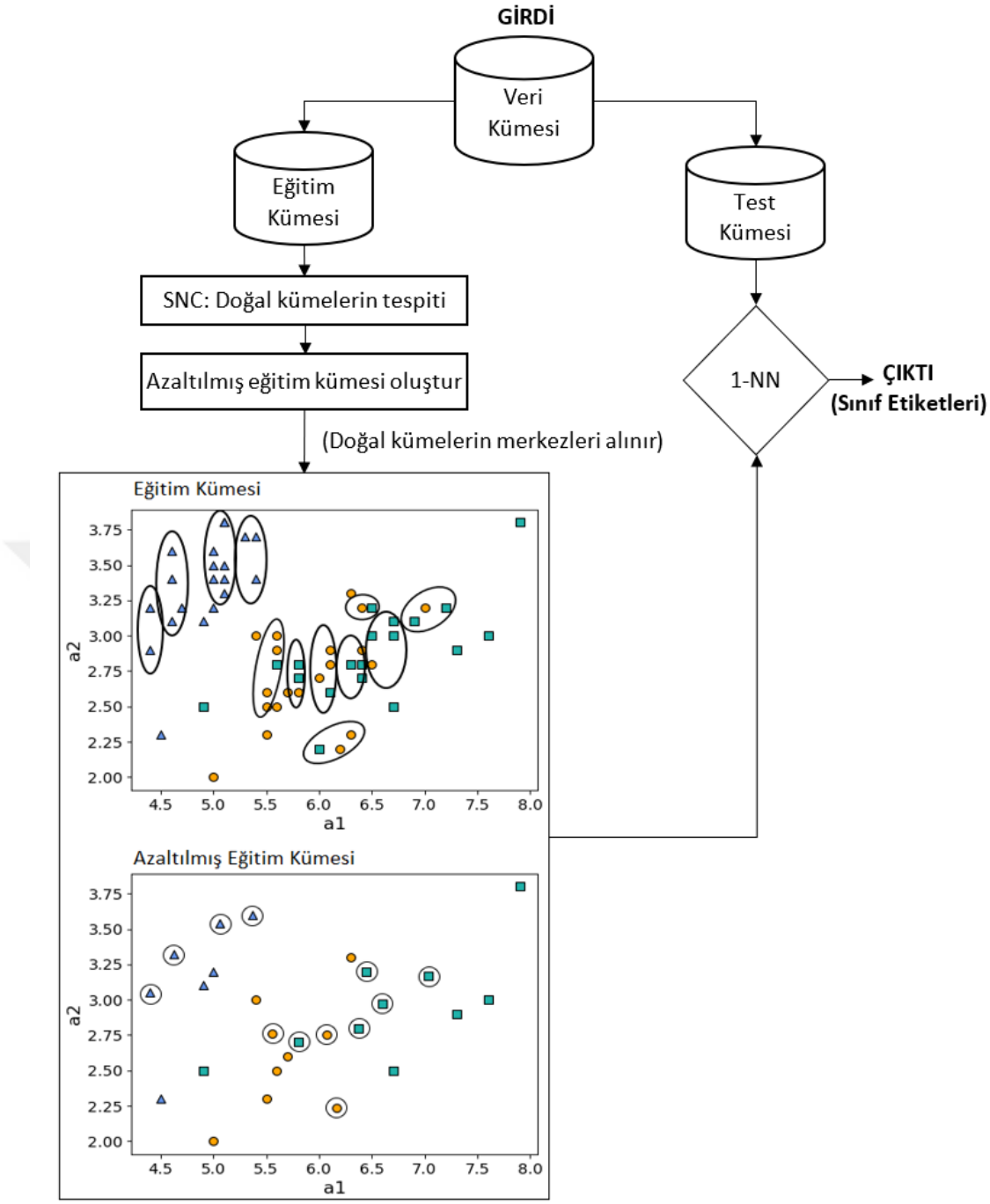
Adım1. SNC kümeleme algoritması uygulayarak doğal kümeleri belirle.

Adım2. Doğal kümelerin merkezleri ve kümelerin dışında kalan örnekler ile azaltılmış eğitim kümesini oluştur.

Adım3. Azaltılmış eğitim kümesinde test örneğine en benzer örneği bul.

Adım4. Eğer en benzer örnek, küme merkezlerinden biri ise o kümenin çoğunluk etiketini test örneğine ata, değilse en benzer örneğin etiketini ata.

NC-MSI algoritmasının akış diyagramı Şekil 3.17'de verilmiştir. Diyagramda azaltılmış eğitim kümesi gösteriminde doğal kümelerin merkezleri daire içene alınarak belirtilmiştir.



Şekil 3.17. NC-MSI algoritması akış diyagramı

SNC kümeleme algoritması uygulandığında, eğitim kümesi iki bölüme ayrılmaktadır: C ile ifade edilen doğal kümeler ve F ile ifade edilen kümelerin dışında kalan elemanlardır. SNC kümeleme sonucunda l adet küme oluştuğunu ve r adet örneğin kümelerin dışında kaldığını varsayalım, $x_j \in \boldsymbol{R}^d$ olmak üzere $C = \{C_1, C_2, \dots, C_l\}$ ve $F = \{x_1, x_2, \dots, x_r\}$ elde edilir.

NC-MSI yönteminde sınıflandırma öncesinde SNC kümeleme algoritmasının uygulanması, bazı avantajlar sağlamaktadır. İlk olarak, benzer bir yaklaşım olan *K*NN algoritmasında yer alan K parametresine ihtiyaç yoktur. SNC ile elde edilen doğal kümelerin merkezleri, her bir küme için kümedeki örneklerin çoğunluk oylaması ile belirlenen sınıf etiketi ile kullanılmakta ve bu şekilde $K = 1$ olarak uygulanması sağlanmaktadır. Ayrıca, $1 \leq j \leq l$ için C_j doğal kümesinin merkezi $c_j \in \boldsymbol{R}^d$ ile ifade edilmek üzere eğitim kümesi $T = \{c_1, c_2, \dots, c_l, x_1, x_2, \dots, x_r\}$ şeklinde azaltılmaktadır.

Önerilen algoritmanın kullanılması ile elde edilen azaltılmış eğitim kümesinin örnek sayısının ($n_T = l + r$), orijinal eğitim kümesi boyutundan (n) oldukça az olması beklenmektedir. Ayrıca, bazı test örnekleri için örneğin en benzer olduğu örneğin F içerisinde olması halinde $K = 1$ olacaktır. Diğer yandan, en benzer olduğu örnek C kümesindeki doğal kümelerden olan test örnekleri için K değeri algoritma tarafından dolaylı olarak belirlenmektedir. Dolayısıyla bu yaklaşım veri kümesinin farklı bölümleri için uyarlanabilir (adaptive) K değerleri elde etmek için doğal bir yol sağlamaktadır.

NC-SVM Sınıflandırma Algoritması

Doğal Kümeler Tabanlı Destek Vektör Makinesi (Natural Clusters-based Support Vector Machine, NC-SVM) sınıflandırma yöntemi, daha önce tanıtılan SNC kümeleme algoritması ile SVM sınıflandırma yöntemi birleştirilerek oluşturulmuştur. SNC kümeleme algoritması ile elde edilen kümeler üzerinde bir veri azaltma uygulanmakta ve bu şekilde azaltılmış eğitim kümesi üzerinde SVM modeli oluşturularak sınıflandırma yapılmaktadır. Doğal kümeler üzerinden yapılan azaltma işleminin sınıflar arasındaki en büyük marj (maximum margin) değerine dayalı bir yöntem olan SVM için yararlı olabileceği düşünülmektedir.

NC-SVM algoritmasında uygulanan genel adımlar şunlardır:

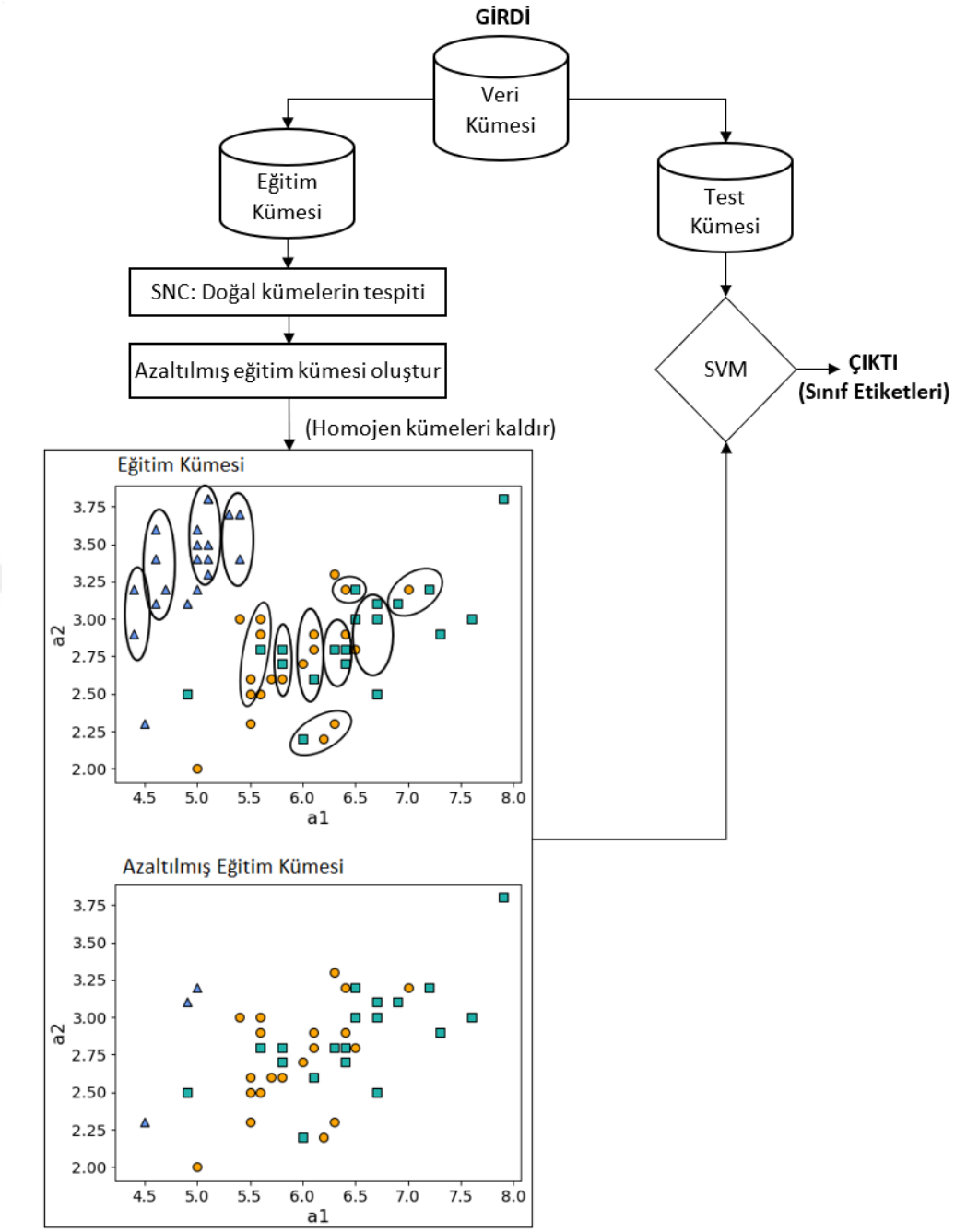
Adım1. SNC kümeleme algoritmasını uygulayarak doğal kümeleri tespit et.

Adım2. Sadece bir sınıfa ait örnekler içeren kümeleri (yani homojen/türdeş kümeleri) sil.

Adım3. Kalan örnekleri kullanarak azaltılmış eğitim kümesini oluştur.

Adım4. Azaltılmış eğitim kümesine SVM yöntemini uygula.

NC-SVM algoritmasının akış diyagramı Şekil 3.18'de verilmiştir.



Şekil 3.18. NC-SVM algoritması akış diyagramı

NC-SVM-B Algoritması

Doğal Kümeler Tabanlı Destek Vektör Makinesi-Sınırlar (Natural Clusters-based Support Vector Machine-Boundaries, NC-SVM-B) sınıflandırma yöntemi, daha önce tanıtılan NC-SVM yönteminin değiştirilmiş bir halidir. Bu algoritmada da, SNC kümeleme algoritması ile doğal kümeler tespit edildikten sonra SVM sınıflandırma yöntemi uygulanmaktadır. Ancak bu algoritmada tespit edilen doğal kümelenmelerin tamamı kaldırılmış ve sınırları oluşturduğu düşünülen kümeler dışında kalan örnekler azaltılmış eğitim kümesini oluşturmak üzere bırakılmıştır.

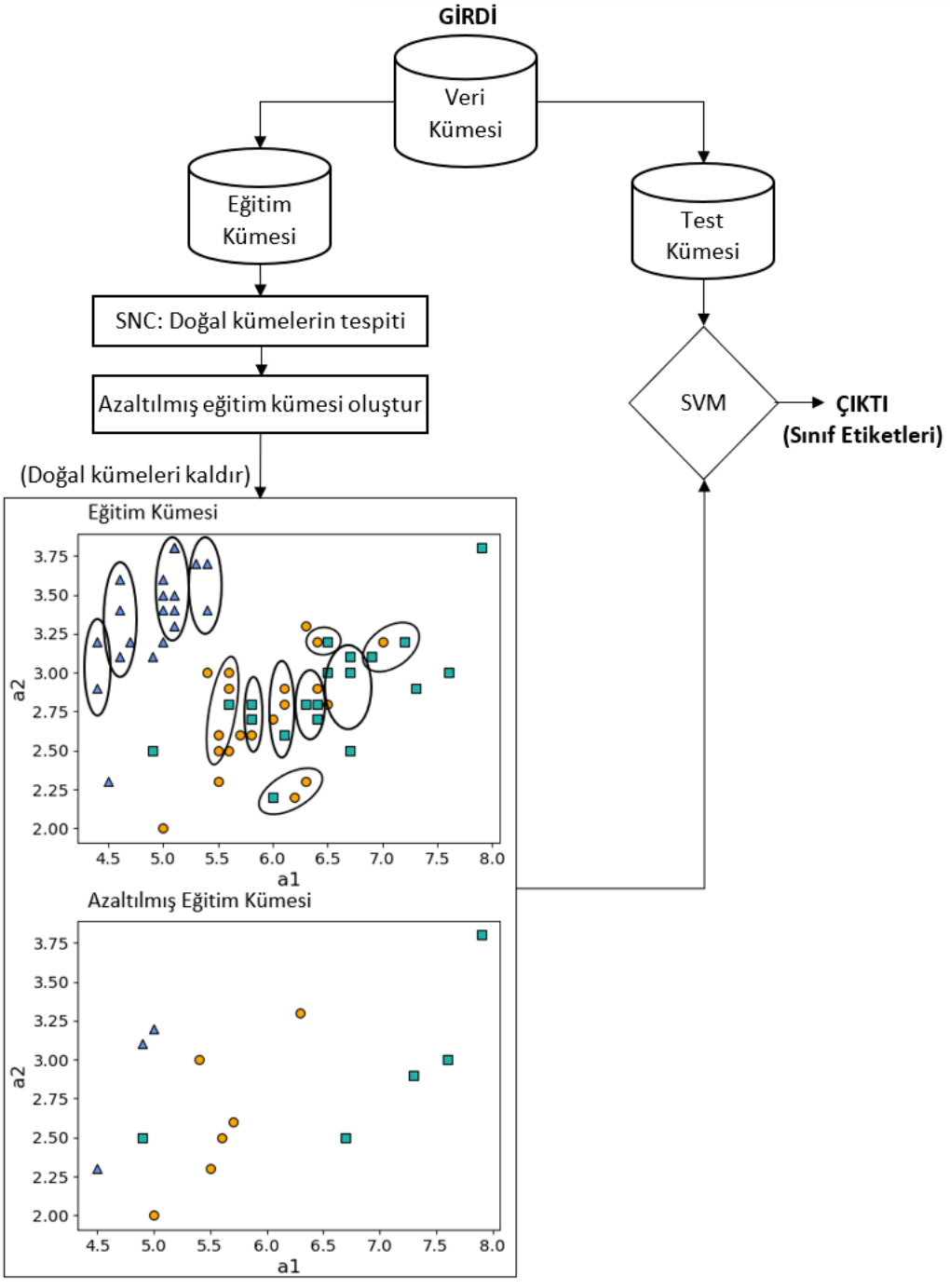
NC-SVM-B algoritmasında uygulanan genel adımlar şunlardır:

Adım1. SNC kümeleme algoritmasını uygulayarak doğal kümeleri tespit et.

Adım2. Tüm doğal kümeleri kaldır ve kalan örnekleri eğitim kümesi olarak kullan.

Adım3. Azaltılmış eğitim kümesine SVM yöntemini uygula.

NC-SVM-B algoritmasının temel adımlarını gösteren akış diyagramı Şekil 3.19'da verilmiştir.



Şekil 3.19. NC-SVM-B algoritması akış diyagramı

Bu yaklaşımda, veri azaltma, diğer iki yaklaşıma kıyasla daha da geliştirilmiştir. Bilindiği üzere, SVM farklı sınıflara ait örnekler arasındaki en büyük marj değerine sahip hiper düzlem fikrine dayanmaktadır. Bu, her sınıftan sınırlarda olan bazı örnekler kullanılarak sınıflar arasındaki en yüksek marjlı hiper düzleminin bulunmasıyla sağlanmaktadır. NC-SVM-B algoritmasında SVM, sınırlardaki örneklere uygulandığı için etkin bir yaklaşım sağlanabileceği düşünülmektedir.

3.5.2. Temsili Noktalar Tabanlı Destek Vektör Makinesi

Önerilen bu yöntem, veri kümesinin yapısal bilgisini oluşturan temsili noktaların tespit edilmesi ve bu noktaların tüm eğitim kümesi yerine denetimli öğrenme sürecinde kullanılması fikrine dayanmaktadır. Kümeleme yöntemleri, etiketsiz veri üzerinden ilgili veriye ait yapısal bilgiyi çıkarabilmektedir. Başarılı bir kümeleme yöntemi olan CURE algoritması, her aşamasında temsili noktaları kullanarak hiyerarşik olarak kümeleri tespit etmektedir. Bu algoritma, temsili noktalar yardımı ile küresel olmayan (non-spherical) kümeleri de tespit edebilmesi ile dikkat çekmektedir. Bu nedenle, önerilen yöntemde, veri kümesini daha az örnekle en iyi şekilde temsil edecek noktaların belirlenmesi ve bu şekilde veri kümesinin yapısal bilgisinin çıkarılması amacıyla CURE algoritması kullanılmıştır. CURE algoritması ile eğitim kümesinden elde edilen yapısal bilgi, bir denetimli öğrenme problemi olan sınıflandırma probleminde kullanılmıştır.

Literatürde, eğitim kümesindeki örnek sayısının artmasının SVM yönteminin performansını olumsuz etkilediğine değinilmektedir (Almasi ve Rouhani, 2016; Sayed ve Hassanien, 2017). Bu nedenle, veri kümesinden elde edilen yapısal bilgi, tüm eğitim kümesi yerine SVM yönteminin eğitiminde kullanılmıştır. Bu şekilde, CURE ve SVM yöntemlerinin birlikte kullanılması ile Temsili Noktalar Tabanlı Destek Vektör Makinesi (Representative Points-based Support Vector Machine, RP-SVM) olarak adlandırılan yeni bir yöntem önerilmiştir. Önerilen RP-SVM yaklaşıma ait temel adımlar aşağıda verilmiştir.

Adım1. Veri kümesini, eğitim ve test kümesi olarak ikiye ayır.

Adım2. Eğitim kümesini sınıf etiketlerine göre gruplara ayır ve her bir gruptan sınıf etiketlerini kaldır.

Adım3. Her bir gruba CURE kümeleme yöntemini uygula (CURE yöntemi için küme sayısı k, daraltma katsayısı α ve kümedeki temsili noktaların sayısı c parametrelerinin belirlenmesi gerekmektedir).

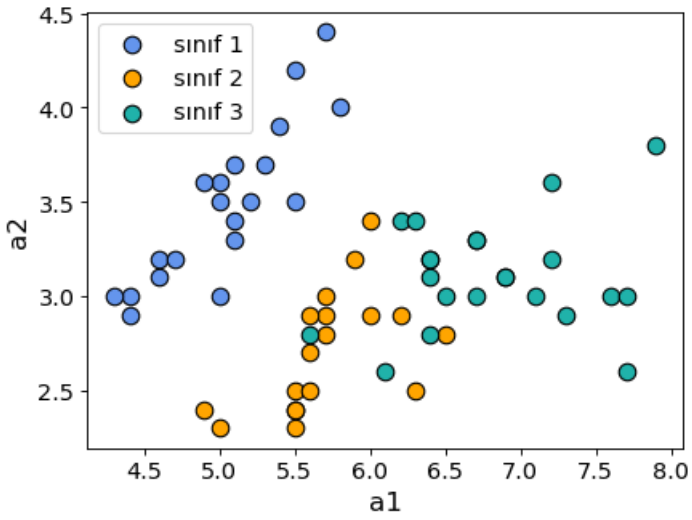
Adım4. Her bir grupta CURE yönteminden elde edilen temsili noktalara sınıf etiketlerini yeniden ekle ve temsili noktaları bir kümede birleştir.

Adım5. Tüm temsili noktaları eğitim kümesi olarak kullanarak SVM yöntemini uygula.

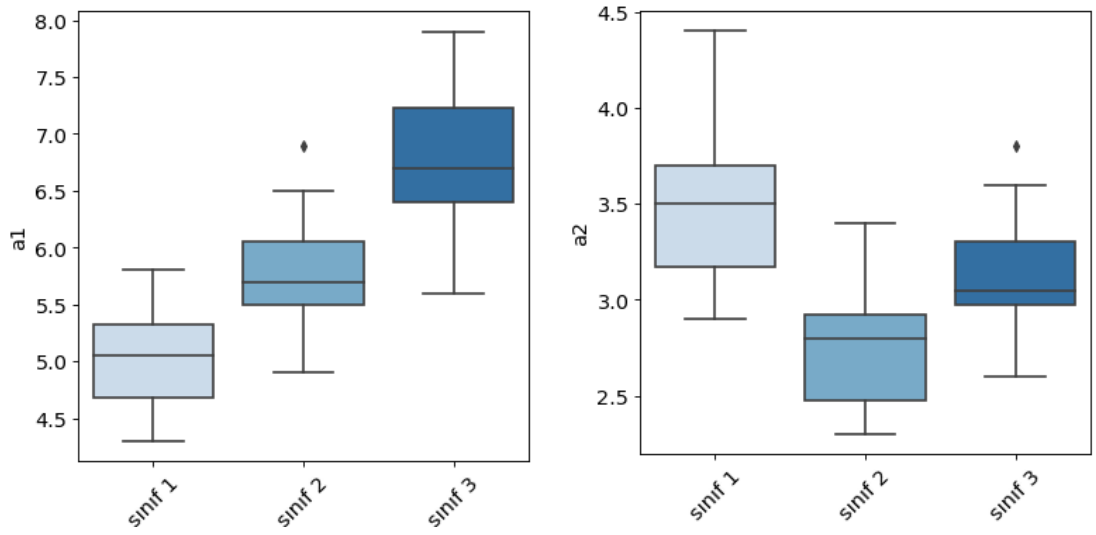
Adım6. SVM yönteminden elde edilen model yardımıyla test örneklerini sınıfla.

RP-SVM Yöntemi Örnek Uygulama

Önerilen RP-SVM yönteminin adımlarını uygulamalı olarak sunmak için iris veri kümesinden bir alt örneklem oluşturulmuştur. Bu alt örneklem, iris veri kümesinde bulunan 3 sınıfın her birinden rastgele 20 veri noktası olmak üzere toplam 60 veri noktası içermektedir. Ayrıca, veri kümesi ve elde sonuçları geometrik olarak ifade edebilmek için iris veri kümesinden sadece *Sepal Length* (a1) ve *Sepal Width* (a2) özellikleri kullanılmıştır. Bu alt örnekleme ait saçılım grafiği Şekil 3.20'de ve kutu-grafikleri Şekil 3.21'de verilmiştir.



Şekil 3.20. Alt örnekleme ait saçılım grafiği



Şekil 3.21. Alt örnekleme ait kutu grafikleri

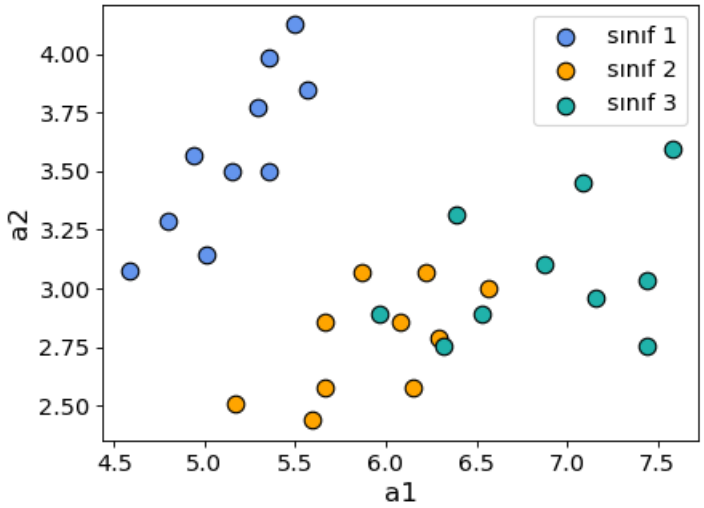
Adım1. Iris veri kümesinden alınan alt örneklem eğitim kümesi olarak kullanılmıştır.

Adım2. Alt örneklem, veri kümesinde bulunan 3 sınıfa göre 3 alt gruba ayrılmıştır. Her bir gruptan sınıf etiketleri kaldırılmıştır. Gruplarda yer alan veri noktaları aşağıda verilmiştir.

- Grup 1 (1. sınıfa ait veriler): 1, 2, 3, ... ,19 ve 20. örnek
- Grup 2 (2. sınıfa ait veriler): 21, 22, 23, ... ,39 ve 40. örnek
- Grup 3 (3. sınıfa ait veriler): 41, 42, 43, ... ,59 ve 60. örnek

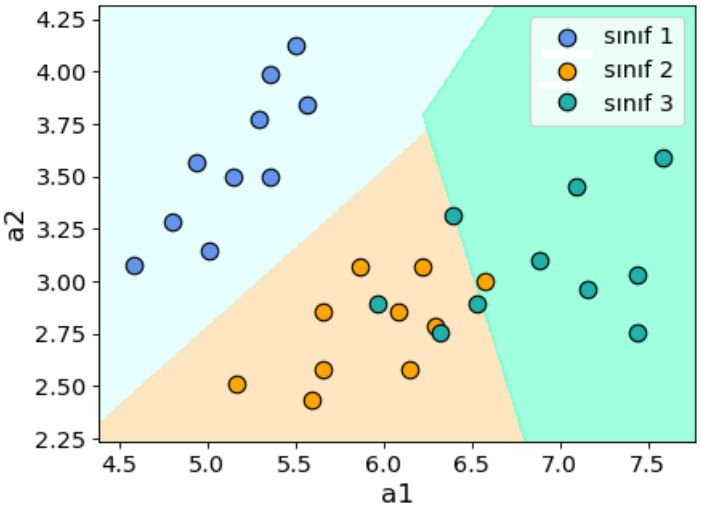
Adım3. Her bir gruba ayrı ayrı CURE kümeleme uygulanmıştır (CURE kümeleme yöntemi için gerekli olan küme sayısı, temsili noktaların sayısı ve daralma katsayısı parametreleri bu örnek için $k = 1$, $c = 10$ ve $\alpha = 0{,}3$ şeklinde belirlenmiştir). Bu şekilde her bir gruptan temsili noktalar elde edilir.

Adım4. Elde edilen temsili noktalara sınıf etiketleri eklenmiş ve tek bir kümede birleştirilmiştir. Bu noktalara ait saçılım grafiği Şekil 3.22'de verilmiştir.



Şekil 3.22. Birleştirilen temsili noktaların kümesi

Adım5. Bir önceki adımda elde edilen temsili noktalar SVM algoritmasında eğitim kümesi olarak kullanılmıştır. Bu şekilde SVM yönteminin oluşturduğu model Şekil 3.23'te verilmiştir.

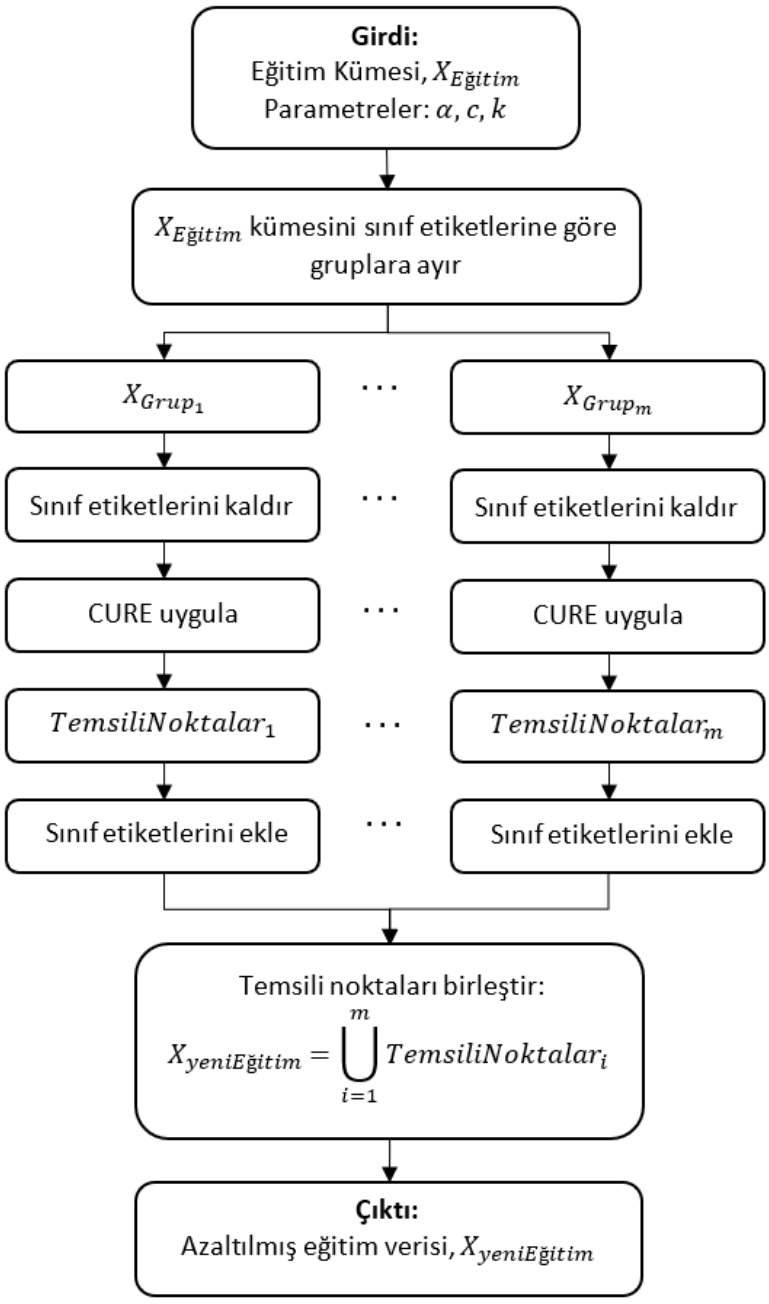


Şekil 3.23. Temsili noktaların kullanımı ile oluşan SVM modeli

Adım6. Elde edilen model kullanılarak etiketsiz olan test verileri sınıflandırılır.

Bu örnekte, eğitim kümesi olarak ele alınan alt örneklem 60 veri noktası içerirken sonuçta elde edilen temsili noktaların kümesi 30 veri noktası içermektedir. Bu şekilde, eğitim kümesi daha az sayıda örnek içeren temsili noktalarla ifade edilmiştir.

RP-SVM yönteminde veri kümesinden yapısal bilginin çıkarılması işlemine ait akış diyagramı Şekil 3.24'te verilmiştir.



Şekil 3.24. RP-SVM yönteminde eğitim kümesinden yapısal bilginin elde edilmesi aşaması

4. BULGULAR VE TARTIŞMA

Bu tez çalışması kapsamında önerilen sınıflandırma yaklaşımları literatürde yer alan benzer yaklaşımlar ve/veya iyi bilinen sınıflandırma yaklaşımları ile karşılaştırmalı olarak analiz edilmiştir. Bu bölümde, yapılan deneysel analiz aşamasında her bir yöntemin uygulanma koşullarına ve elde edilen sonuçlara değinilmiştir. Sonuçlar tablolar ve uygun grafiklerle ifade edilmiştir. Elde edilen sonuçlar üzerinden bulgulara değinilerek tartışmaya yer verilmiştir.

4.1. Deneysel Analiz

Daha önce bahsedildiği üzere bu çalışma kapsamında öncelikle bir kümeleme yaklaşımı (SNC kümeleme algoritması) önerilmiştir. Önerilen kümeleme yaklaşımı temel alınarak NC-MSI, NC-SVM ve NC-SVM-B olmak üzere üç faklı sınıflandırma yaklaşımı geliştirilmiştir. Ayrıca temsili noktaları temel alan CURE kümeleme yöntemine dayalı bir sınıflandırma yaklaşımı olan RP-SVM yöntemi önerilmiştir. Bu bölümde, önerilen sınıflandırma yaklaşımları, literatürde yer alan benzer ve/veya iyi bilinen sınıflandırma yöntemleri ile karşılaştırmalı olarak test edilmiştir. Test aşamasında daha önce tanıtılan *Iris*, *Wine*, *Sonar*, *Glass*, *Haberman*, *E.coli*, *Bupa*, *Ionosphere*, *Breast-cancer*, *Transfusion* ve *Vehicle* gerçek hayat veri kümeleri kullanılmıştır. Önerilen her bir yöntem belirlenen koşullar altında birbirinden bağımsız olarak test edilmiştir.

4.1.1. NC-MSI Sınıflandırma Algoritmasının Test Edilmesi

NC-MSI sınıflandırma algoritması, nümerik türde 10 farklı veri kümesi üzerinde test edilmiştir. Kullanılan veri kümeleri %70 eğitim ve %30 test verisi olacak şekilde rastgele bölünmüştür. NC-MSI yöntemi benzer yapısından dolayı iyi bilinen sınıflandırma yöntemlerinden biri olan *K*-En Benzer Komşuluk sınıflandırma algoritması ile karşılaştırılmıştır. NC-MSI yöntemi için belirlenmesi gereken parametre eşik değeri (t) parametresidir. Bu parametre için $t \in \{0{,}85;\ 0{,}90;\ 0{,}95\}$ değerleri uygulanmıştır. *K*NN sınıflandırma algoritması için ise sadece K

parametresinin belirlenmesi gerekmektedir. K parametresi için $K \in \{1,2,\dots,15\}$ değerleri uygulanmıştır. Her iki yöntem için en iyi sonuç veren parametre değerleri dikkate alınmıştır. NC-MSI ve *K*NN yöntemlerinden edilen sonuçlar ve en iyi parametre değerleri Çizelge 4.1'de verilmiştir. Çizelgede en iyi doğruluk değerleri koyu olarak işaretlenmiştir.

Çizelge 4.1. NC-MSI ve *K*NN için sonuçlar

#	Veri kümesi	*K*NN			NC-MSI		
		n_T	K	Doğruluk	n_T	Eşik değeri	Doğruluk
1	Iris	105	7	**0,98**	44	0,90	0,96
2	Wine	125	11	**0,81**	121	0,95	0,72
3	Sonar	146	4	**0,82**	128	0,95	0,79
4	Glass	148	1	**0,73**	79	0,85	0,67
5	Haberman	215	15	**0,74**	110	0,90	0,67
6	E.coli	235	3	**0,85**	139	0,95	0,83
7	Ionosphere	246	1	0,83	176	0,85	**0,84**
8	Breast-cancer	490	5	**0,97**	342	0,85	0,92
9	Transfusion	523	5	**0,81**	304	0,85	0,73
10	Vehicle	590	1	**0,66**	284	0,85	**0,66**

n_T – Eğitim kümesi örnek sayısı

4.1.2. NC-SVM Sınıflandırma Algoritmasının Test Edilmesi

NC-SVM sınıflandırma algoritması, nümerik türde 10 farklı veri kümesi üzerinde test edilmiştir. Kullanılan veri kümeleri %70 eğitim ve %30 test verisi olacak şekilde rastgele bölünmüştür. NC-SVM yöntemi standart SVM yöntemi ile karşılaştırılmıştır. Standart SVM yöntemi tüm eğitim kümesini kullanırken NC-SVM yönteminde azaltılmış eğitim kümesi kullanılmaktadır.

SVM yönteminin uygulanmasındaki önemli adımlardan birisi uygun çekirdek fonksiyonunun belirlenmesidir. Testlerde, önceki bölümlerde değinilen yaygın kullanılan çekirdek fonksiyonları kullanılmıştır. NC-SVM ve standart SVM yöntemlerinin testlerinde lineer, radyal tabanlı fonksiyon, sigmoid ve polinomiyal çekirdek fonksiyonları ayrı ayrı uygulanmış ve her iki yöntem için de en iyi sonucu veren çekirdek fonksiyonu ilgili yöntemin uygulanmasında dikkate alınmıştır.

NC-SVM ve standart SVM yöntemlerinden elde edilen sonuçlar Çizelge 4.2'de verilmiştir. Çizelgede en iyi doğruluk değerleri koyu olarak işaretlenmiştir.

Çizelge 4.2. NC-SVM ve standart SVM için sonuçlar

Veri Kümesi	Standart SVM			NC-SVM		
	Çekirdek*	$n_T(n_{SV})$	Doğruluk	Çekirdek*	$n_T(n_{SV})$	Doğruluk
D1	4	105(15)	**0,96**	4	32(14)	**0,96**
D2	2	125(89)	**0,99**	2	116(84)	**0,99**
D3	4	146(96)	**0,86**	4	115(86)	**0,86**
D4	4	148(112)	**0,71**	4	107(89)	0,67
D5	4	215(113)	**0,75**	4	133(102)	**0,75**
D6	2	235(111)	**0,88**	1	151(89)	0,86
D7	2	246(121)	**0,94**	2	158(112)	**0,94**
D8	3	490(80)	**0,97**	1	349(61)	0,96
D9	2	523(264)	**0,79**	4	364(224)	**0,79**
D10	4	590(259)	**0,84**	4	389(222)	0,83
n_T: Eğitim kümesi örnek sayısı, n_{SV}: Destek vektörlerin sayısı						
*Çekirdek Fonksiyonları:	1–Lineer, 2–Radyal Tabanlı Fonksiyon, 3–Sigmoid, 4–Polinomiyal					
Veri Kümesi:	*D1*–Iris, *D2*–Wine, *D3*–Sonar, *D4*–Glass, *D5*– Haberman, *D6*–E.coli, *D7*–Ionosphere, *D8*–Breast-cancer, *D9*–Transfusion, *D10*–Vehicle					

4.1.3. NC-SVM-B Sınıflandırma Algoritmasının Test Edilmesi

NC-SVM-B sınıflandırma algoritması, nümerik türde 10 farklı veri kümesi üzerinde test edilmiştir. Kullanılan veri kümeleri %70 eğitim ve %30 test verisi olacak şekilde rastgele bölünmüştür. NC-SVM-B yöntemi standart SVM yöntemi ile karşılaştırılmıştır. Standart SVM yöntemi tüm eğitim kümesini kullanırken NC-SVM-B yönteminde azaltılmış eğitim kümesi kullanılmaktadır.

SVM yönteminin uygulanmasındaki önemli adımlardan birisi uygun çekirdek fonksiyonunun belirlenmesidir. Önceki bölümlerde yaygın kullanılan çekirdek fonksiyonları belirtilmiştir. NC-SVM-B ve standart SVM yöntemlerinin testinde lineer, radyal tabanlı fonksiyon, sigmoid ve polinomiyal çekirdek fonksiyonları ayrı ayrı uygulanmış ve her iki yöntem için de en iyi sonucu veren çekirdek fonksiyonu, ilgili yöntemin uygulanmasında dikkate alınmıştır.

NC-SVM-B ve standart SVM yöntemlerinden elde edilen sonuçlar Çizelge 4.3'te verilmiştir. Çizelgede en iyi doğruluk değerleri koyu olarak işaretlenmiştir.

Çizelge 4.3. NC-SVM-B ve standart SVM için sonuçlar

Veri Kümesi	Standart SVM			NC-SVM-B		
	Çekirdek*	$n_T(n_{SV})$	Doğruluk	Çekirdek*	$n_T(n_{SV})$	Doğruluk
D1	4	105(29)	**1,00**	4	23(13)	0,93
D2	4	125(65)	**1,00**	4	33(16)	0,95
D3	4	146(117)	**0,85**	4	98(71)	0,82
D4	4	149(122)	**0,71**	4	60(45)	0,69
D5	4	215(113)	**0,75**	4	54(28)	**0,75**
D6	2	235(111)	**0,88**	1	100(56)	0,85
D7	2	246(77)	0,92	2	145(66)	**0,94**
D8	3	490(80)	**0,97**	1	281(32)	**0,97**
D9	2	523(247)	**0,79**	4	212(100)	0,75
D10	4	590(288)	**0,85**	4	201(129)	0,82
n_T: Eğitim kümesi örnek sayısı, n_{SV}: Destek vektörlerin sayısı						
*Çekirdek Fonksiyonları:	1–Lineer, 2–Radyal Tabanlı Fonksiyon, 3–Sigmoid, 4–Polinomiyal					
Veri Kümesi:	*D1*–Iris, *D2*–Wine, *D3*–Sonar, *D4*–Glass, *D5*– Haberman, *D6*–E.coli, *D7*–Ionosphere, *D8*–Breast-cancer, *D9*–Transfusion, *D10*–Vehicle					

4.1.4. RP-SVM Sınıflandırma Algoritmasının Test Edilmesi

RP-SVM yöntemi; standart SVM, KMSVM (Wang vd. (2005) tarafından önerilen), CART ve *K*NN yöntemleri ile karşılaştırmalı olarak analiz edilmiştir. Bu yöntemlerde, optimum parametrelerin belirlenmesi sınıflandırma performansı açısından önemlidir. *K*NN yöntemi için gereken parametre ve CART yöntemi için gereken ana parametreler ve her bir parametre için uygulanan değer aralıkları Çizelge 4.4'te verilmiştir. *K*NN ve CART yöntemlerinin parametrelerine uygun değerleri belirlemek için scikit-learn kütüphanesi (Pedregosa vd., 2011) ile sağlanan çapraz doğrulanmış grid-arama (cross-validated grid-search) yöntemi uygulanmıştır.

Çizelge 4.4. *K*NN ve CART için gereken parametreler ve değer aralıkları

Yöntem	**Parametre**	**Değer Aralığı**
*K*NN	*K*	[1,30]
CART	*criterion*	{“gini”, “entropy”}
	max_depth	[1,10]
	min_samples_split	[2,10]
	min_samples_leaf	[1,20]

Standart SVM, KMSVM ve RP-SVM yöntemlerinde çekirdek fonksiyonu olarak yaygın kullanılan çekirdek fonksiyonlarından biri olan Radyal Tabanlı Fonksiyon (Radial Basis Function, RBF) kullanılmıştır. RBF çekirdek kullanımı durumunda çekirdek parametresi γ ve maliyet (cost) parametresi C ayarlanmalıdır. Bu parametrelerin belirlenmesi için grid-arama yönteminde, $\gamma \in \{2^{-10}, 2^{-9}, \dots, 2^{4}\}$ ve $C \in \{2^{-2}, 2^{-1}, \dots, 2^{12}\}$ değerleri uygulanmıştır (Bertini vd., 2011). RBF için, scikit-learn kütüphanesi (Pedregosa vd., 2011) ile sağlanan çapraz doğrulanmış grid-arama yöntemi uygulanarak en iyi parametre değerleri aranmıştır.

Testlerde kullanılan veri kümelerinde, hem ikili hem de çok sınıflı veri kümeleri bulunmaktadır. Standart SVM, KMSVM ve RP-SVM yöntemlerinde çok sınıflı sınıflandırma için bire-karşı-bir (one-against-one) metodu kullanılmıştır. Bu yöntem pratik kullanımlar için en uygun yöntemlerden birisidir (Debnath vd., 2004).

KMSVM yöntemi için ayrıca sıkıştırma oranını gösteren CR parametresinin belirlenmesi gerekmektedir. CR parametresi için $CR \in \{10, 20, 30, 40\}$ değerleri kullanılarak parametre ayarlaması yapılmıştır.

RP-SVM yönteminde ise ayrıca CURE kümeleme yönteminden kaynaklanan parametrelerin belirlenmesi gerekmektedir. CURE kümeleme yöntemi için gereken parametreler: k (kümelerin sayısı), c (temsili noktaların sayısı) ve a (daraltma katsayısı) parametreleridir. Guha vd. (2001) CURE kümeleme yönteminin, 0,2’den 0,7’ye kadar tüm a değer aralığı için doğru kümeler bulduğunu, bu aralığın küresel olmayan kümelerin tespiti için iyi bir değer aralığı olduğunu ve aykırı değerlerin etkisini azalttığını belirtmişlerdir. Bu bilgiye dayanarak RP-SVM yönteminde a parametresi için 0,2-0,7 değer aralığı kullanılmıştır.

Ayrıca Guha vd. (2001), çok küçük c değerleri için küme kalitesinin düştüğünü ve 10'dan büyük c değerlerinde CURE algoritmasının doğru kümeleri tespit edebildiğini belirtmiştir. Bu bilgiden yararlanılarak RP-SVM yönteminde c parametresi için Denklem 4.1 önerilmiştir. $n_{sınıf_i}$ veri kümesinin i. sınıfında yer alan veri noktalarının sayısı ve $oran$ ilgili sınıftaki verilerin yüzde kaçının temsili nokta olarak kullanılacağını belirleyen değer olmak üzere c parametresi Denklem 4.1 yardımıyla hesaplanmaktadır.

$$c = \begin{cases} n_{sınıf_i} * oran, & n_{sınıf_i} > 10 \\ n_{sınıf_i}, & 1 \leq n_{sınıf_i} \leq 10 \end{cases} \quad (4.1)$$

Denklem 4.1'e göre veri kümesinde 10'dan fazla örnek içeren sınıflara CURE uygulanır. 10 ve daha az örnek içeren sınıflara CURE uygulanmaz ve bu sınıflardaki örnekler azaltılmış eğitim kümesine doğrudan eklenir. Ayrıca denklemde görüldüğü üzere c parametresi $n_{sınıf_i}$ ve $oran$ parametrelerine bağlıdır. $n_{sınıf_i}$ değeri kullanılan veri kümesinden elde edilirken, $oran$ parametresinin belirlenmesi gerekmektedir. Bu denklemde belirtilen $oran$ parametresi için $oran = \{0{,}2; 0{,}3; \ldots; 0{,}8\}$ değerleri kullanılmıştır. Bu şekilde her bir sınıftaki verinin %20'sinden %80'ine varacak şekilde farklı oranlarda temsili nokta seçimi sağlanmıştır.

RP-SVM yönteminde, kümeleme işlemi sınıf bazlı uygulanmış ve her bir sınıf bir küme olarak ele alınmıştır. Bu nedenle küme sayısını gösteren k parametresi $k = 1$ olarak sabitlenmiş ve tüm testlerde aynı değer kullanılmıştır.

RP-SVM yönteminde belirtilen parametrelere en iyi değerlerin belirlenmesi için 10-katlı çapraz doğrulanmış grid arama yöntemi uygulanmıştır.

Deneysel analizde tüm yöntemler katmanlı iç içe 10-katlı çapraz doğrulama (stratified nested 10-fold cross validation) yöntemi ile test edilmiştir. Bu işlem farklı rastgele çekirdek değerleri (random seed value) ile 30 kere tekrar edilmiş ve her bir yöntem için sonuçların ortalaması alınmıştır. Elde edilen sonuçlar Çizelge 4.5'te verilmiştir. Çizelgede en iyi doğruluk değerleri koyu olarak işaretlenmiştir.

Çizelge 4.5. Karşılaştırmalı sonuçlar

Veri Kümesi	İndeks	RP-SVM	Standart-SVM	KMSVM	*K*NN	CART
D1	Acc±Std	95,44±0,99	**95,98±0,95**	94,38±0,02	94,96±0,94	94,27±0,87
	n_T	67,3	135	13.0	135	135
	n_{SV}	25,7	39,6	9,4	-	-
D2	Acc±Std	97,46±0,80	**97,83±0,74**	94,26±0,01	96,38±0,66	92,17±1,73
	n_T	100,8	161	12,9	161	161
	n_{SV}	63,9	87,6	10,5	-	-
D3	Acc±Std	81,15±1,77	**85,27±1,67**	67,04±0,04	85,12±2,14	72,26±2,79
	n_T	106,7	188	16,0	188	188
	n_{SV}	68,2	128,6	12,0	-	-
D4	Acc±Std	66,75±1,77	**70,00±1,68**	54,31±0,02	68,64±1,59	66,73±2,31
	n_T	146,9	193	17,0	193	193
	n_{SV}	102,8	130,3	16,0	-	-
D5	Acc±Std	73,05±1,04	72,37±1,07	72,90±0,01	**74,88±0,67**	72,26±1,41
	n_T	177,1	276	20,7	276	276
	n_{SV}	74,0	148,4	10,7	-	-
D6	Acc±Std	72,11±0,93	**72,42±0,94**	63,65±0,03	65,25±1,38	66,85±1,45
	n_T	227,7	311	30,9	311	311
	n_{SV}	157,4	221,9	18,3	-	-
D7	Acc±Std	93,00±0,82	**94,11±0,58**	93,30±0,01	85,55±0,70	88,38±1,31
	n_T	231,4	316	30,9	316	316
	n_{SV}	74,8	115,1	29,2	-	-
D8	Acc±Std	95,63±0,26	**96,35±0,21**	96,13±0,01	96,04±0,30	93,80±0,51
	n_T	393,7	630	35,7	630	630
	n_{SV}	37,5	81,9	11,2	-	-
D9	Acc±Std	76,64±0,42	76,80±0,25	78,31±0,01	**78,99±0,50**	78,12±0,68
	n_T	228,9	674	64,0	674	674
	n_{SV}	108,0	324,7	15,5	-	-
D10	Acc±Std	83,29±0,97	**84,61±0,78**	70,10±0,01	71,07±0,87	71,54±1,28
	n_T	537,2	762	76,0	762	762
	n_{SV}	216,6	301,7	56,4	-	-

Acc: Doğruluk, Std: Standart Sapma

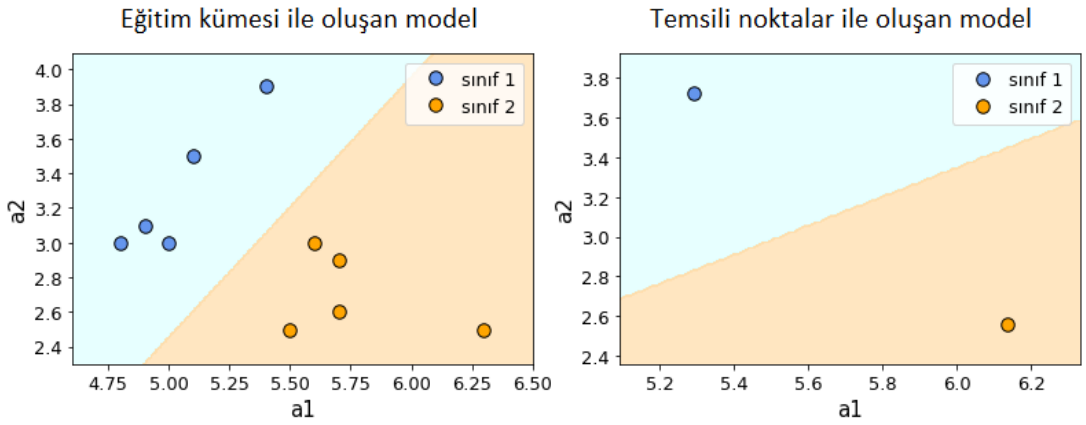
n_T: Eğitim kümesi örnek sayısı, n_{SV}: Destek vektörlerin sayısı

Veri Kümesi: *D1*–Iris, *D2*–Wine, *D3*–Sonar, *D4*–Glass, *D5*– Haberman, *D6*–Bupa, *D7*–Ionosphere, *D8*–Breast-cancer, *D9*–Transfusion, *D10*–Vehicle

Temsili Noktaların Sayısı için Hassasiyet Analizi

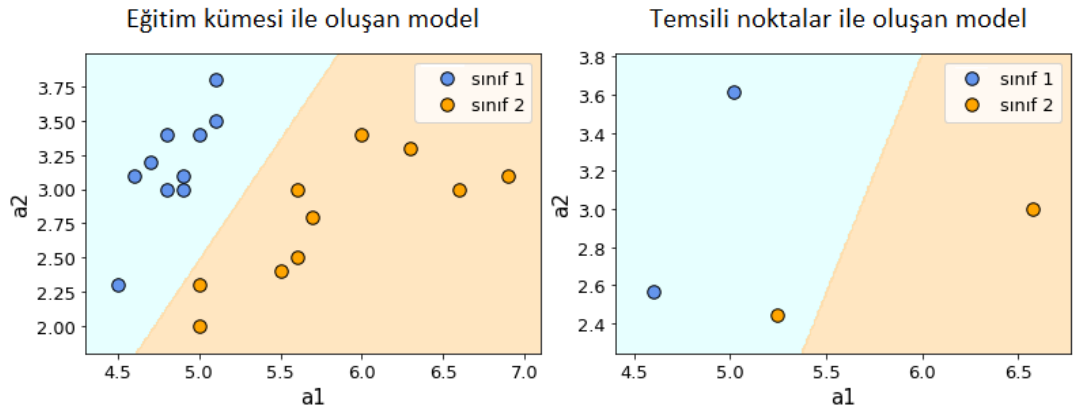
RP-SVM yönteminde temsili noktaların sayısının (c) belirlenmesi için önerilen Denklem 4.1 için hassasiyet analizi yapılmıştır. Hassasiyet analizi için iris veri kümesinden farklı boyutlarda 6 farklı alt örneklem seçilmiştir. Veri kümesi 2 sınıflı olarak ele alınmıştır. Sonuçları geometrik olarak ifade edebilmek için veri kümesindeki özelliklerden sadece *Sepal Length* (a1) ve *Sepal Width* (a2) özellikleri kullanılmıştır. Belirtilen alt örneklemlerin her biri üzerinde standart SVM ve RP-SVM yöntemleri ayrı ayrı uygulanmıştır. Denklem 4.1'in sonuçta oluşan model üzerindeki etkisini gözlemlemek için sadece $n_{sınıf_i}$ parametresi değiştirilmiş, diğer parametreler için sabit değerler kullanılmıştır. Temsili noktaların tespiti için gereken parametrelerde $a = 0{,}3$ ve $oran = 0{,}2$ varsayılan değerleri kullanılmıştır. Elde edilen modeller grafiklerle karşılaştırmalı olarak sunulmuştur.

$n_{sınıf_1} = 5$ ve $n_{sınıf_2} = 5$ veri noktası içeren alt örneklem (*Örneklem1*) ile elde edilen SVM modeli ve bu örneklemden çıkarılan temsili noktalar ile elde edilen SVM modeli Şekil 4.1'de verilmiştir. Bu örneklem için temsili noktaların sayısı $c = 1$ olmaktadır.



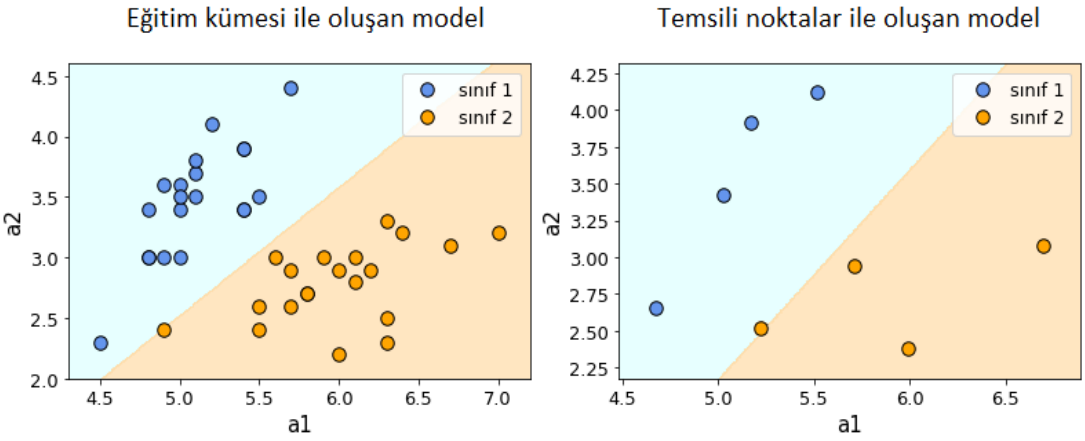
Şekil 4.1. *Örneklem1* için SVM modelleri

$n_{sınıf_1} = 10$ ve $n_{sınıf_2} = 10$ veri noktası içeren alt örneklem (*Örneklem2*) ile elde edilen SVM modeli ve bu örneklemden çıkarılan temsili noktalar ile elde edilen SVM modeli Şekil 4.2'de verilmiştir. Bu örneklem için temsili noktaların sayısı $c = 2$ olmaktadır.



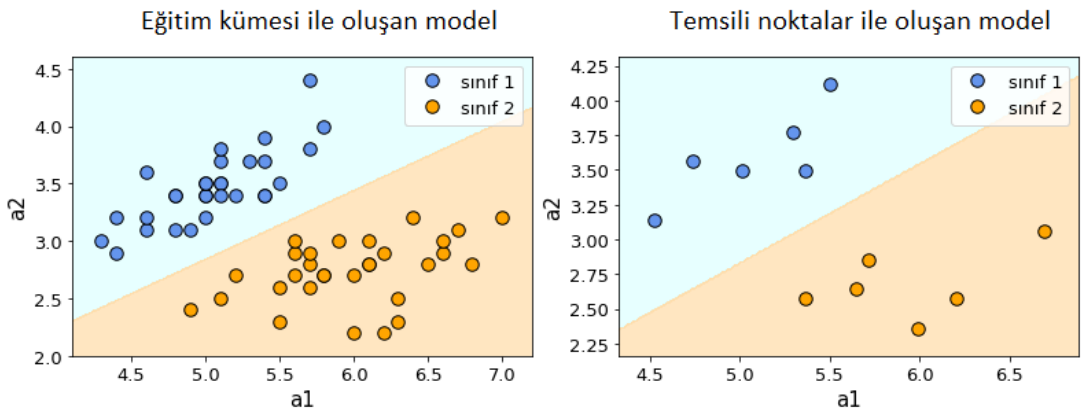
Şekil 4.2. *Örneklem2* için SVM modelleri

$n_{sınıf_1} = 20$ ve $n_{sınıf_2} = 20$ veri noktası içeren alt örneklem (*Örneklem3*) ile elde edilen SVM modeli ve bu örneklemden çıkarılan temsili noktalar ile elde edilen SVM modeli Şekil 4.3'te verilmiştir. Bu örneklem için temsili noktaların sayısı $c = 4$ olmaktadır.



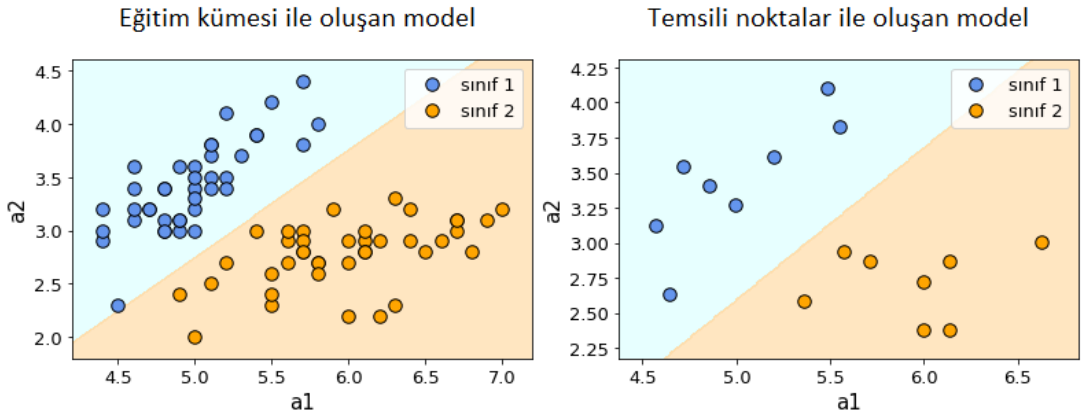
Şekil 4.3. *Örneklem3* için SVM modelleri

$n_{sınıf_1} = 30$ ve $n_{sınıf_2} = 30$ veri noktası içeren alt örneklem (*Örneklem4*) ile elde edilen SVM modeli ve bu örneklemden çıkarılan temsili noktalar ile elde edilen SVM modeli Şekil 4.4'te verilmiştir. Bu örneklem için temsili noktaların sayısı $c = 6$ olmaktadır.



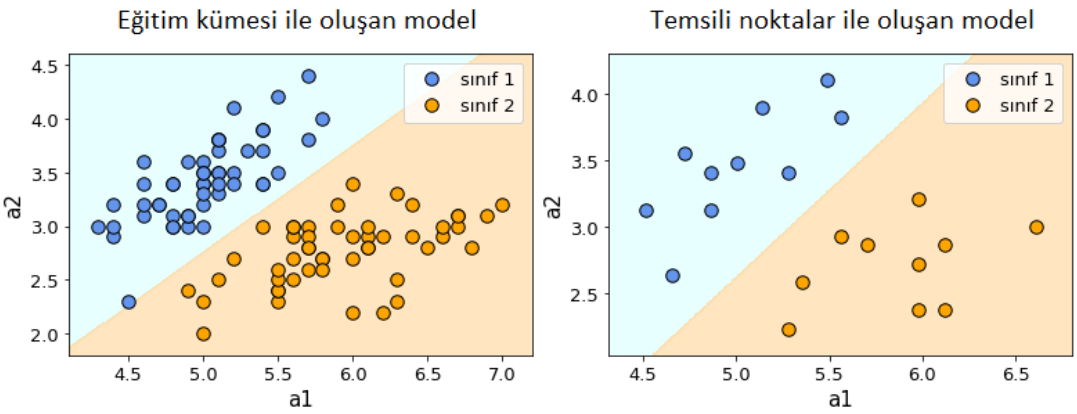
Şekil 4.4. *Örneklem4* için SVM modelleri

$n_{sınıf_1} = 40$ ve $n_{sınıf_2} = 40$ veri noktası içeren alt örneklem (*Örneklem5*) ile elde edilen SVM modeli ve bu örneklemden çıkarılan temsili noktalar ile elde edilen SVM modeli Şekil 4.5'te verilmiştir. Bu örneklem için temsili noktaların sayısı $c = 8$ olmaktadır.



Şekil 4.5. *Örneklem5* için SVM modelleri

$n_{sınıf_1} = 50$ ve $n_{sınıf_2} = 50$ veri noktası içeren alt örneklem (*Örneklem6*) ile elde edilen SVM modeli ve bu örneklemden çıkarılan temsili noktalar ile elde edilen SVM modeli Şekil 4.6'da verilmiştir. Bu örneklem için temsili noktaların sayısı $c = 10$ olmaktadır.



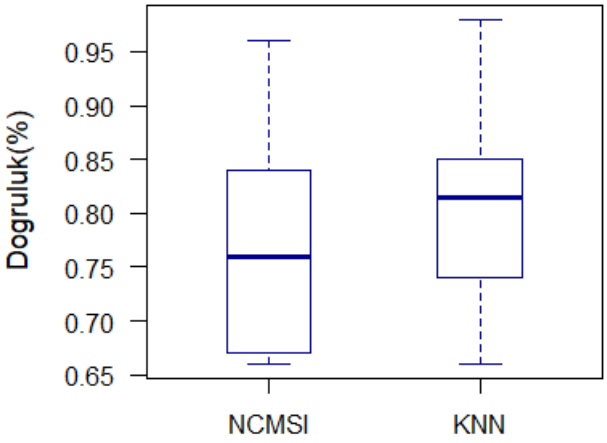
Şekil 4.6. *Örneklem6* için SVM modelleri

Sonuçlar incelendiğinde küçük c değerleri için kümelerin ve oluşan modelin kalitesinin azaldığı görülmektedir. Örneğin, Şekil 4.1 ve Şekil 4.2' de görüldüğü gibi $n_{sınıf_i} \leq 10$ için $c = n_{sınıf_i} * oran$ uygulanması durumunda temsili noktaların sayısı oldukça azalmış ve oluşan modelde ayırıcı hiper düzlem (separating hyperplane) standart SVM yöntemine kıyasla değişmiştir. Bu durum seçilen temsili noktaların, verinin yapısal bilgisini çıkarmak için yeterli olmadığını göstermektedir. Şekil 4.3, Şekil 4.4, Şekil 4.5 ve Şekil 4.6'da ise $n_{sınıf_i} > 10$ için $c = n_{sınıf_i} * oran$ uygulanması durumunda seçilen temsili noktaların verinin geometrisini daha iyi yakaladığı ve standart SVM'ye benzer ayırıcı hiper düzlem oluşturduğu görülmektedir.

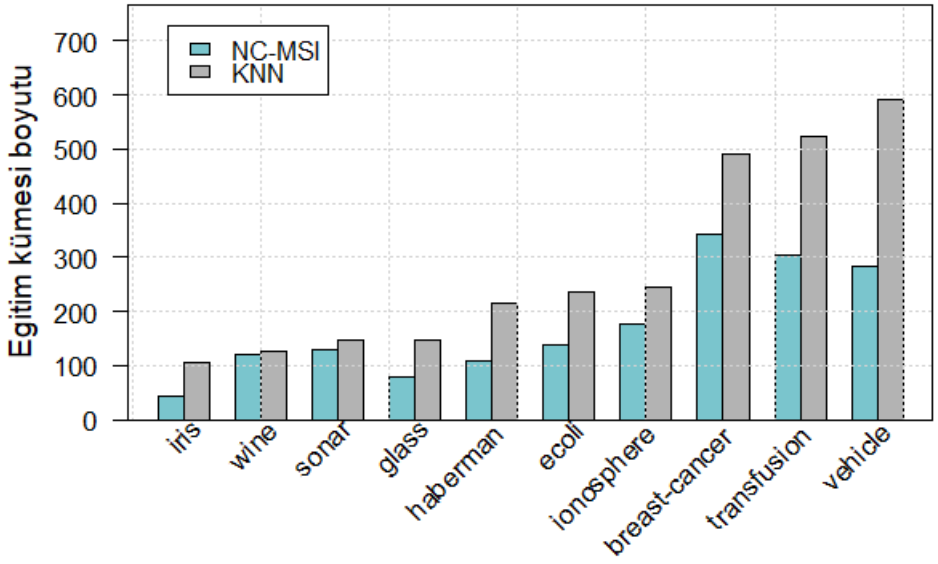
4.2. Sonuçlar ve Tartışma

4.2.1. NC-MSI Sınıflandırma Algoritması için Bulgular

NC-MSI ve *K*NN algoritmaları doğruluk ve eğitim kümesi boyutu açısından değerlendirilmiştir. Şekil 4.7'de doğruluk sonuçları kutu grafiği ve Şekil 4.8'te eğitim kümelerinin boyutu çubuk grafiği ile karşılaştırmalı olarak verilmiştir.



Şekil 4.7. NCMSI ve KNN yöntemleri için doğruluk değerlerinin karşılaştırılması



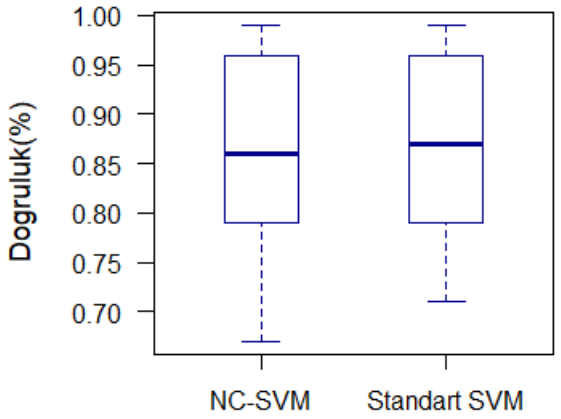
Şekil 4.8. NC-MSI ve *K*NN yöntemleri için eğitim kümesi boyutunun karşılaştırılması

Şekil 4.7 ve Şekil 4.8'de verilen grafikler incelendiğinde tüm veri kümeleri için NC-MSI yönteminin, kullanılan eğitim kümesinin boyutunu ciddi oranda azalttığı görülmektedir. Bu azaltma ile birlikte NC-MSI yöntemi doğruluk açısından düşüşlere neden olmaktadır. Ancak bazı veri kümelerinde küçük oranlarda düşüşler olduğu veya doğruluk değerinin korunabildiği görülmektedir. Örneğin, NC-MSI yöntemi Vehicle veri kümesinde, eğitim kümesi boyutunu 590 örnekten 284 örneğe düşürerek eğitim kümesi boyutunda %50'yi aşan bir oranda azaltma yapmış ve buna rağmen *K*NN ile aynı doğruluğu elde etmiştir. Diğer bir veri kümesi olan Ionosphere veri kümesinde ise NC-MSI yönteminin, eğitim veri kümesi boyutunu 246 örnekten 176 örneğe düşürerek veri kümesi boyutunda yaklaşık %28 azaltma yaptığı ve buna rağmen *K*NN yönteminden daha yüksek doğruluk elde ettiği görülmektedir. Bu sonuçlar, veri

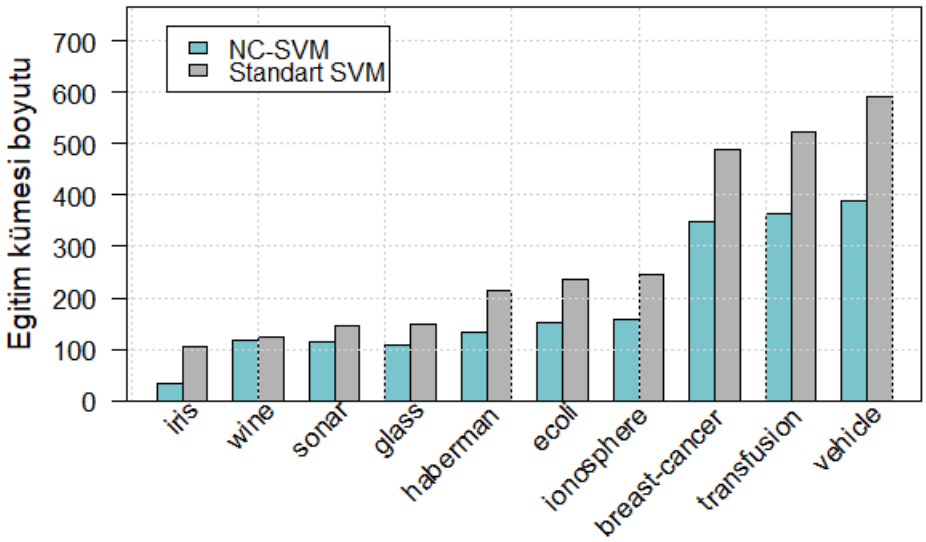
kümesine göre değişkenlik göstermekle birlikte önerilen yöntemin, veri kümesinin sınıflandırma için anlamlı olan bölümünü etkin şekilde tespit edebildiğini ortaya koymaktadır. Bununla birlikte, kümeleme yönteminin sınıflandırmada kullanımının yeni sınıflandırma yöntemlerinin geliştirilmesi için önemli olabileceğini göstermektedir.

4.2.2. NC-SVM Sınıflandırma Algoritması için Bulgular

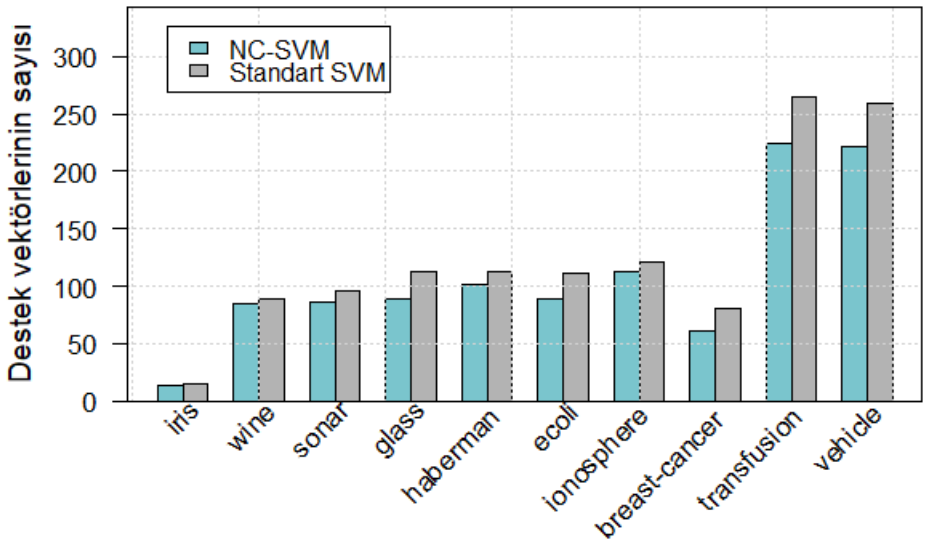
NC-SVM ve standart SVM algoritmaları doğruluk, eğitim kümesinin boyutu ve model geliştirmede kullanılan destek vektörlerinin sayısı açısından karşılaştırılmıştır. Şekil 4.9'da doğruluk değerleri için kutu grafiği verilmiştir. Şekil 4.10'da eğitim kümesinin boyutu ve Şekil 4.11'de destek vektörlerin sayısı için çubuk grafikleri karşılaştırmalı olarak verilmiştir.



Şekil 4.9. NC-SVM ve standart SVM yöntemleri için doğruluk değerlerinin karşılaştırılması



Şekil 4.10. NC-SVM ve standart SVM yöntemleri için eğitim kümesi boyutunun karşılaştırılması



Şekil 4.11. NC-SVM ve standart SVM yöntemleri için destek vektörlerinin sayısının karşılaştırılması

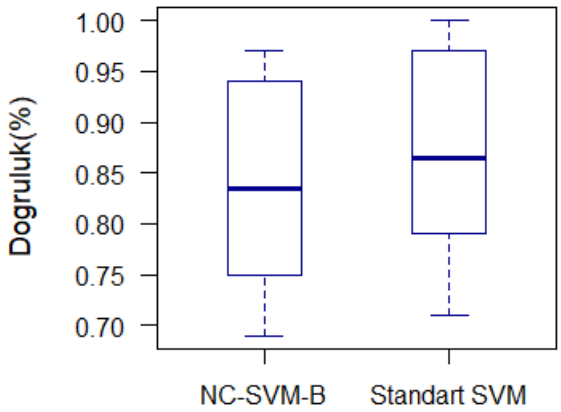
Şekil 4.9, Şekil 4.10 ve Şekil 4.11'de verilen grafikler incelendiğinde NC-SVM yönteminin tüm veri kümelerinde, eğitim kümesi örneklerinin sayısını önemli ölçüde azalttığı ve model oluşturmada daha az destek vektörü kullandığı görülmektedir. Eğitim kümesi boyutu ve destek vektörlerinin sayısındaki bu azaltmaya rağmen NC-SVM yöntemi; Iris, Wine, Sonar, Haberman, Ionosphere ve Transfusion veri kümelerinde standart SVM yöntemi ile aynı doğruluk değerlerini elde etmiştir. Örneğin; Iris veri kümesinde standart SVM yöntemi tüm eğitim kümesini yani 105 örneği kullanırken, NC-SVM yöntemi eğitim kümesini 32 örneğe düşürerek ciddi bir azaltma sağlamış ve sonuçta standart SVM yöntemi ile aynı doğruluk değerini elde etmiştir. Vehicle ve Breast-cancer veri kümelerinde ise NC-SVM yönteminin doğruluk değerinde oldukça az bir düşüşe neden olduğu görülmektedir.

Bu sonuçlar, veri kümesine göre değişkenlik göstermekle birlikte önerilen NC-SVM yönteminin, veri kümesinin sınıflandırma için anlamlı olan bölümünü etkin şekilde tespit edebildiğini göstermektedir. Bununla birlikte, kümeleme yönteminin veri kümesindeki yapılanmaları etkin şekilde tespit ederek sınıflandırma yöntemine adapte edebildiğini ortaya koymaktadır.

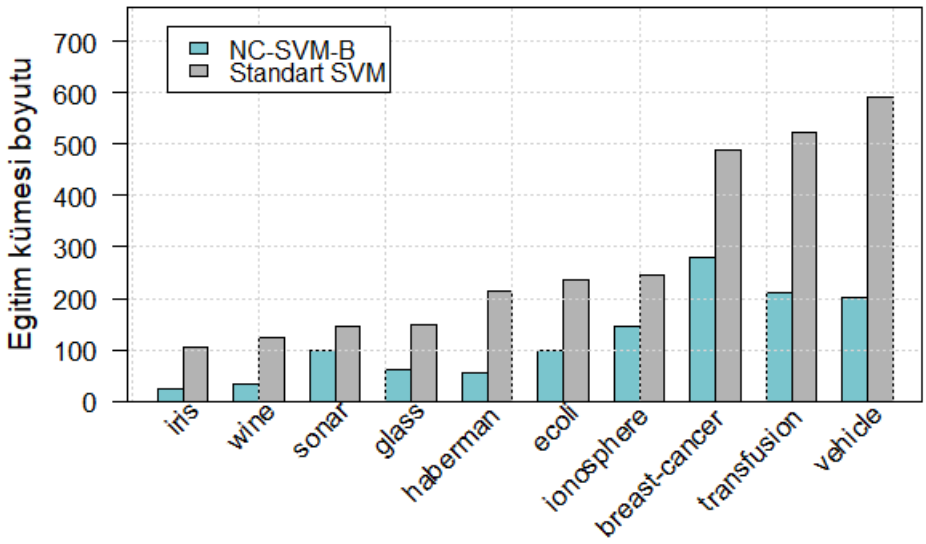
4.2.3. NC-SVM-B Sınıflandırma Algoritması için Bulgular

NC-SVM-B ve standart SVM algoritmaları doğruluk, eğitim kümesinin boyutu ve model geliştirmede kullanılan destek vektörlerin sayısı açısından karşılaştırılmıştır. Şekil 4.12'de doğruluk değerleri için kutu grafiği verilmiştir.

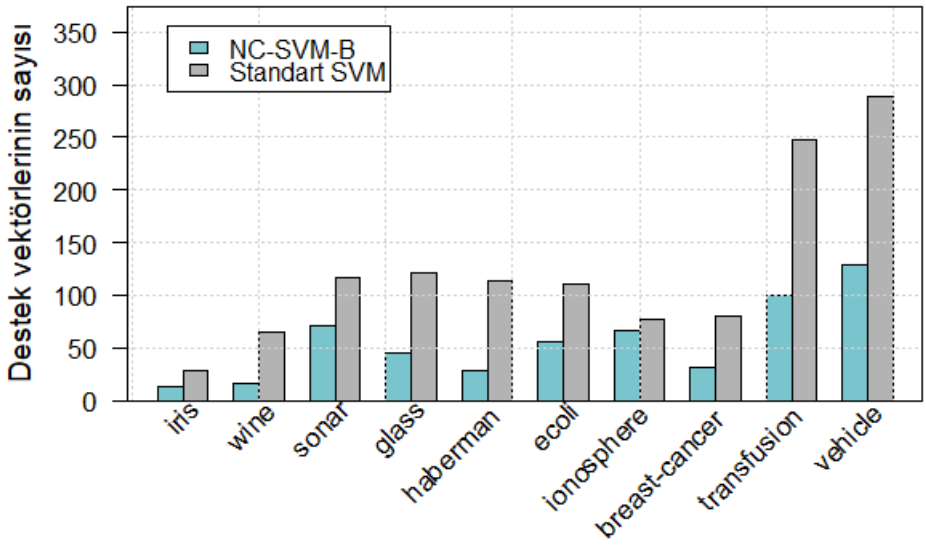
Şekil 4.13'te eğitim kümesinin boyutu ve Şekil 4.14'te destek vektörlerin sayısı çubuk grafikleri ile karşılaştırmalı olarak verilmiştir.



Şekil 4.12. NC-SVM-B ve standart SVM yöntemleri için doğruluk değerlerinin karşılaştırılması



Şekil 4.13. NC-SVM-B ve standart SVM yöntemleri için eğitim kümesi boyutunun karşılaştırılması



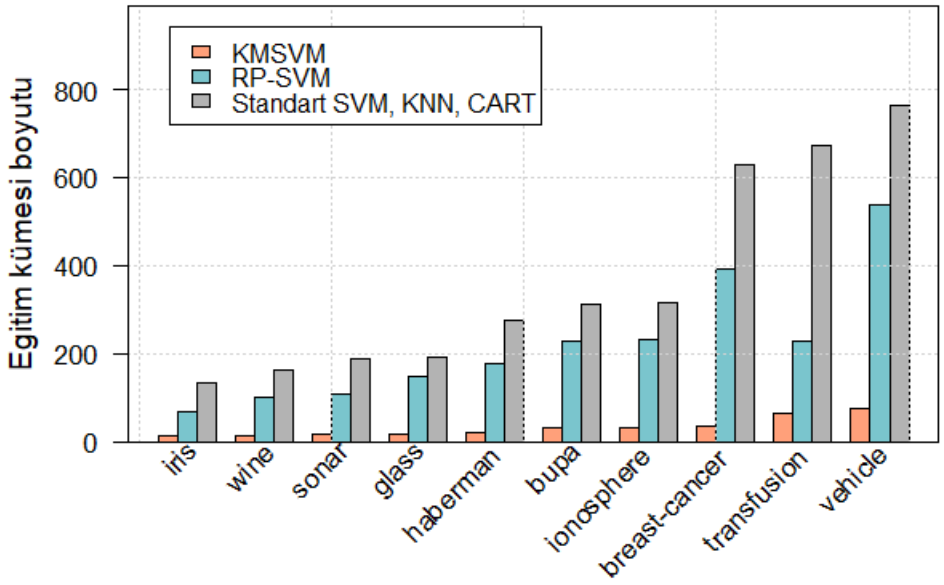
Şekil 4.14. NC-SVM-B ve standart SVM yöntemleri için destek vektörlerinin sayısının karşılaştırılması

Şekil 4.12, Şekil 4.13 ve Şekil 4.14'te verilen grafikler incelendiğinde NC-SVM-B yönteminin tüm veri kümelerinde eğitim kümesi örneklerinin sayısını önemli ölçüde azalttığı ve daha az destek vektörü kullandığı görülmektedir. - Bu azaltma ile birlikte NC-SVM-B yöntemi, bazı veri kümelerinde doğruluk açısından düşüşlere neden olmaktadır. Ancak bazı veri kümelerinde, küçük oranlarda düşüşler olduğu veya doğruluk değerinin korunduğu görülmektedir. Örneğin, NC-SVM-B yöntemi Haberman veri kümesinde eğitim kümesi boyutunu 215 örnekten 54 örneğe düşürerek eğitim kümesi boyutunda yaklaşık %75 oranında azaltma yaptığı ve buna rağmen standart SVM ile aynı doğruluğu elde edebildiği görülmektedir. Breast-cancer veri kümesinde ise NC-SVM-B yöntemi, eğitim kümesinin boyutunu 490 örnekten 281 örneğe düşürerek yaklaşık %43 oranında azaltma yaparken SVM ile aynı doğruluğu elde etmektedir. Ionosphere veri kümesinde ise NC-SVM-B yönteminin, eğitim kümesinin boyutunu 246 örnekten 145 örneğe düşürerek yaklaşık %41 oranında azaltma yaparken dikkat çeken bir sonuçla standart SVM'den daha yüksek doğruluk değeri elde ettiği görülmektedir.

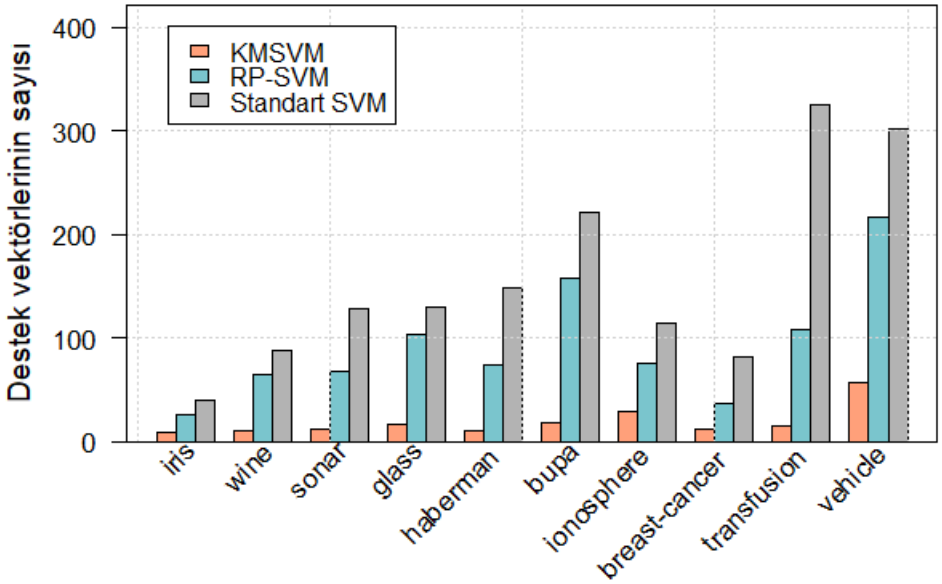
Bu sonuçlar, NC-SVM-B yönteminin özellikle bazı veri kümelerinde, veri kümesinin sınıflandırma için önemli olan yapısal bilgisini/doğal yapılanmalarını etkin şekilde tespit edebildiğini ve bu bilgiyi/yapılanmaları tüm eğitim kümesi yerine kullanarak etkin bir sınıflandırma sağlayabildiğini göstermektedir. Sonuçlar veri kümesine göre farklılık göstermektedir.

4.2.4. RP-SVM Sınıflandırma Algoritması için Bulgular

Elde edilen sonuçlar doğruluk, eğitim kümesinin boyutu ve model geliştirmede kullanılan destek vektörlerin sayısı açısından karşılaştırılmıştır. Şekil 4.15'te eğitim kümesinin boyutu ve Şekil 4.16'da destek vektörlerin sayısı için çubuk grafikleri verilmiştir.



Şekil 4.15. KMSVM, RP-SVM, Standart SVM, *K*NN ve CART yöntemleri için eğitim kümesi boyutunun karşılaştırılması



Şekil 4.16. KMSVM, RP-SVM, Standart SVM, *K*NN ve CART yöntemleri için destek vektörlerinin sayısının karşılaştırılması

Standart SVM, *K*NN ve CART yöntemleri tüm eğitim kümesini kullanırken RP-SVM ve KMSVM yöntemleri eğitim kümesinde azaltma yaparak daha az eğitim örneği kullanmaktadır. Bununla birlikte RP-SVM ve KMSVM yöntemleri standart SVM yöntemine kıyasla daha az destek vektörü kullanmaktadır. Sonuçlar doğruluk açısından değerlendirildiğinde RP-SVM yönteminin eğitim kümesinin boyutunu ve destek vektörlerin sayısını ciddi oranda azaltırken standart SVM yöntemi ile benzer doğruluk değeri elde edebildiği görülmektedir. Ayrıca RP-SVM yöntemi, daha az eğitim örneği kullanarak *K*NN ve CART yöntemlerinden daha iyi doğruluk elde edebilmiştir. Öte yandan, KMSVM yönteminin en az eğitim örneği ve en az destek vektörü kullanan yöntem olduğu ve bazı veri kümelerinde doğruluk açısından iyi sonuçlar elde ederken bazılarında ciddi düşüşlere sebep olduğu görülmektedir. RP-SVM yönteminin, KMSVM yöntemine kıyasla daha az veri azaltma yaptığı fakat tüm veri kümelerinde doğruluk açısından iyi sonuçlar elde ettiği için daha kararlı bir yöntem olduğu görülmektedir.

Uygulanan yöntemlerden elde edilen sonuçları karşılaştırmak için, parametrik olmayan bir istatistiksel hipotez testi olan Wilcoxon İşaretli Sıralar Testi (Wilcoxon Signed Rank Test) uygulanmıştır. Bu test birden fazla veri kümesi üzerinde uygulanan iki yöntemin karşılaştırılmasını sağlamaktadır. Önerilen RP-SVM yöntemi, Wilcoxon İşaretli Sıralar Testi uygulanarak standart SVM, KMSVM, *K*NN ve CART yöntemleri ile karşılaştırılmıştır.

Wilcoxon İşaretli Sıralar Testi için sıfır hipotezi (H_0) şu şekilde tanımlanmıştır:

"$H_0 = Her\ iki\ yöntem\ de\ eşit\ derecede\ iyi\ performans\ gösterir$."

Testten elde edilen p değerleri Çizelge 4.6'da verilmiştir. Bu çizelgede $a = 0{,}05$ anlamlılık düzeyinin (significance level) altındaki p değerleri kalın olarak işaretlenmiştir.

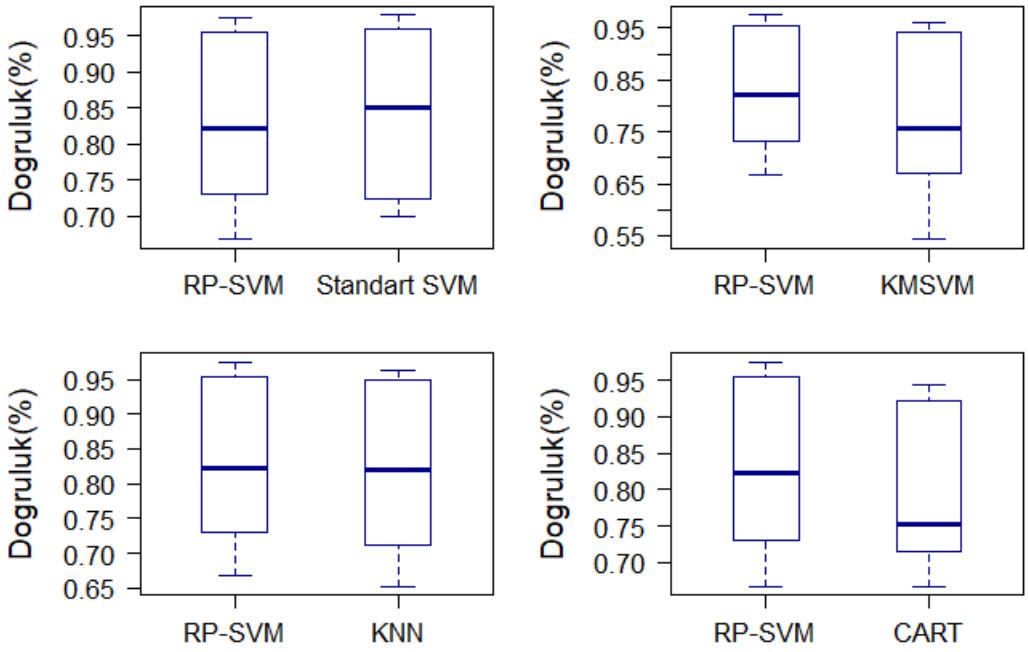
Çizelge 4.6. Wilcoxon işaretli sıralar testi sonuçları

	p değerleri			
Yöntemler	Standart SVM	KMSVM	*K*NN	CART
RP-SVM	**0.01953**	0.08398	0.6953	**0.01367**

Çizelge 4.6'da yer alan sonuçlar, $a \geq 0{,}05$ anlamlılık düzeyi ile RP-SVM yönteminin standart SVM ve CART yöntemlerinden daha iyi performans gösterdiğini

doğrulamaktadır. KMSVM ile karşılaştırma sonucunda elde edilen p değeri ise RP-SVM yönteminin KMSVM yönteminden daha iyi olduğunu söylemek için güçlü bir kanıt olmadığını göstermektedir. Ancak bazı veri kümelerinde (bkz. Çizelge 4.5'te Sonar, Glass ve Vehicle veri kümeleri edilen sonuçlar) KMSVM yönteminin RP-SVM yönteminden oldukça kötü sonuçlar elde ettiği görülmektedir. Bu durum Şekil 4.17'de verilen kutu grafiklerinde de görülmektedir. RP-SVM ve *K*NN için ortalama performansta, $p = 0{,}6953$ değeri ile istatistiksel olarak önemli bir fark yoktur.

Yöntemlerden elde edilen doğruluk sonuçları Şekil 4.17'de kutu grafikleri ile karşılaştırmalı olarak verilmiştir.



Şekil 4.17. RP-SVM yönteminin diğer yöntemlerle karşılaştırılması

Benzer çalışmalar incelendiğinde, K-ortalamalar (Wang vd., 2015; Lee vd., 2007; Chen ve Pan, 2010; Yao vd., 2017; Bang ve Jhun, 2014), BIRCH (Yu vd., 2003; Horng vd., 2011) ve k-uzamsal medyanlar (Bang vd., 2010) gibi kümeleme yöntemlerinin SVM yöntemine adapte edildiği yöntemler görülmektedir. Bu yöntemlerden K-ortalamalar kümeleme yönteminin aykırı değerlere karşı hassas olduğu bilinmektedir (Bang ve Jhun, 2014). CURE kümeleme yöntemi ise daraltma özelliği ile aykırı değerlere karşı dirençlidir. Ayrıca gerçek hayat veri kümelerinde, kümeler farklı şekillerde olabilmektedir. BIRCH kümeleme yöntemi farklı boyutlarda ve küresel olmayan kümeler için uygun değilken (Guha vd., 2001), CURE kümeleme yöntemi farklı boyutlarda ve küresel olmayan kümelerin tespitinde başarılıdır. Ancak

literatürde, etkin bir kümeleme yöntemi olan CURE kümeleme yönteminin, benzer amaçlarla SVM yöntemine adapte edildiği bir çalışma yer almamaktadır. Bu açıdan önerilen RP-SVM yöntemi ile yeni bir yaklaşım oluşturulmuştur.

5. SONUÇ

Bu çalışmada, veri kümesinden elde edilen doğal yapılanmaların/yapısal bilginin makine öğrenmesi sürecine etkisi araştırılmıştır. Bu kapsamda, sınıflandırma problemi ele alınmış ve sınıflandırma öncesinde kümeleme işleminden yararlanan iki aşamalı sınıflandırma yöntemleri üzerinde durulmuştur. Kümeleme yönteminin sınıflandırma öncesi bir ön işlem olarak uygulanmasının temel motivasyonu, kümeleme yardımıyla veri kümesindeki önemli örüntüleri, en anlamlı bilgiyi veya doğal kümelenmeleri tespit etmektir. İlk olarak, örnekler arası benzerliklere dayanan ve Benzerlik Tabanlı Doğal Kümeler (SNC) olarak adlandırılan yeni bir kümeleme yöntemi önerilmiştir. SNC kümeleme yöntemi temel alınarak 3 farklı sınıflandırma yaklaşımı tasarlanmıştır. Bu yaklaşımlardan ilkinde, SNC ile tespit edilen en benzer örnekler sınıflandırma için kullanılmış ve bu yöntem Doğal Kümelere Dayalı En Benzer Örnekler (Natural Clusters-based Most Similar Instances, NC-MSI) olarak adlandırılmıştır. İkinci yaklaşımda, örnekler arası benzerliklerle tespit edilen homojen doğal kümeleri kaldırarak SVM uygulanmış ve bu yöntem Doğal Kümelere Dayalı Destek Vektör Makinesi (Natural Clusters-based Support Vector Machine, NC-SVM) olarak adlandırılmıştır. Son olarak üçüncü yaklaşımda, doğal kümeler farklı şekilde kullanılmış ve tespit edilen doğal kümelerin tamamı kaldırıldıktan sonra SVM uygulanmıştır. Bu şekilde sadece sınırlardaki örneklerin SVM yönteminde kullanılmasını amaçlayan bu yöntem ise Doğal Kümelere Dayalı Destek Vektör Makinesi-Sınırlar (Natural Clusters-based Support Vector Machine-Boundaries, NC-SVM-B) olarak adlandırılmıştır.

Önerilen yöntemler gerçek hayat veri kümeleri üzerinde iyi bilinen benzer yöntemler ile karşılaştırmalı olarak test edilmiştir. NC-MSI, NC-SVM ve NC-SVM-B algoritmaları sınıflandırma için kullanılan eğitim kümesinin boyutunu azaltmaktadır. NC-MSI algoritması için elde edilen sonuçlar, özellikle bazı veri kümelerinde bu algoritmanın etkin sonuçlar elde ettiğini göstermektedir. NC-SVM ve NC-SVM-B algoritmaları için elde edilen sonuçlar ise kümeleme aşamasının, eğitim kümesinde önemli bir azaltma sağlayabildiğini ve eğitim kümesinin sınıflandırma için

kullanılabilecek bölümünü tespit edebildiğini göstermektedir. Ayrıca NC-SVM ve NC-SVM-B algoritmaları, eğitim kümesini ve destek vektörlerinin sayısını önemli ölçüde azaltırken standart SVM algoritması ile benzer veya yakın performans elde edebilmektedir.

Diğer bir yaklaşımda ise veri kümesinin yapısal bilgisinin elde edilmesi ve bu bilginin ilgili denetimli öğrenme algoritmasında eğitim kümesi yerine kullanılması için mevcut kümeleme yaklaşımları araştırılmıştır. Bu doğrultuda, RP-SVM olarak adlandırılan yeni bir sınıflandırma yöntemi önerilmiştir. RP-SVM yöntemi, özellikle küresel olmayan kümelerin tespitinde başarılı olan ve temsili noktalara dayanan CURE kümeleme algoritmasını temel almaktadır. CURE kümeleme algoritması yardımıyla etiketsiz veri kümesini en iyi temsil eden noktalar yani veri kümesinin yapısal bilgisi elde edilmiştir. Elde edilen yapısal bilgi, SVM yönteminin eğitimi için tüm eğitim kümesi yerine kullanılmıştır. Çeşitli gerçek hayat veri kümeleri üzerinde yapılan testlerde eğitim kümesinin %50'den fazlaya varan oranlarda azaltılmış ve azaltılan veri eğitim kümesi yerine SVM yönteminin eğitimi için kullanılmıştır. RP-SVM yöntemi; standart SVM, KMSVM, *K*NN ve CART yöntemleri ile karşılaştırılmıştır. Uygulanan tüm veri kümelerinde, RP-SVM daha az eğitim örneği ile eğitilmiş ve daha az destek vektör kullanılarak model oluşturulmuştur. RP-SVM yöntemi eğitim kümesinin boyutunu ve destek vektörlerin sayısını oldukça azaltırken başarılı sonuçlar elde etmiştir. Bu sonuçlar RP-SVM yönteminin veri kümesini temsil eden noktaları oldukça başarılı şekilde belirleyebildiğini göstermektedir. Bu temsili noktalar veri kümesinin yapısal bilgisinin temsili olarak düşünülebilir.

Çalışma kapsamında önerilen SNC kümeleme yöntemi, kümeleme adımlarına yapılacak düzenlemelerle geliştirilebilecek bir yaklaşım sunmaktadır. Bu yöntemden elde edilen doğal kümeler üç faklı yaklaşım ile sınıflandırma sürecine dahil edilmiştir. Bunlar dışında daha farklı yaklaşımlar izlenerek sınıflandırma yöntemlerinin etkinliği artırılabilir. Ayrıca önerilen yaklaşımlar, farklı makine öğrenmesi yöntemlerine adapte edilerek farklı problemler için uygulanabilir. Bu doğrultuda, farklı yöntemlerin geliştirilmesi için bir zemin oluşturulmuştur. Bununla birlikte, CURE kümeleme yaklaşımına dayalı olarak önerilen RP-SVM yöntemi, eğitim kümesini etkin şekilde azaltmaktadır. RP-SVM yönteminde izlenen yaklaşım farklı makine öğrenmesi yöntemlerine adapte edilebilir.

Sonuç olarak, bu tez kapsamında veri kümesinden daha etkin yararlanmak amacıyla araştırılan ve geliştirilen yaklaşımlar bu alandaki yeni çalışmalara zemin oluşturabilecek niteliktedir. Makine öğrenmesi alanında araştırma ve geliştirmeye açık yaklaşımlar ortaya koyulmuştur. Elde edilen bulgular veri kümesinin daha etkin kullanılabileceğini göstermekte ve bu açıdan büyük veri üzerindeki çalışmalar için de potansiyel oluşturulmaktadır.

KAYNAKLAR

Aburomman, A. A. ve Reaz, M. B. I. (2017). A novel weighted support vector machines multiclass classifier based on differential evolution for intrusion detection systems. *Information Sciences*, 414, 225-246.

Aggarwal, C. C. (2015). *Data mining: the textbook*. Springer.

Aggarwal, C. C. (2014). Instance-Based Learning: A Survey. *Data Classification: Algorithms* and Applications, 157.

Almasi, O. N. ve Rouhani, M. (2016). Fast and de-noise support vector machine training method based on fuzzy clustering method for large real world datasets. *Turkish Journal of Electrical Engineering & Computer Sciences*, 24(1), 219-233.

Angiulli, F. ve Narvaez, E. (2018). Pruning strategies for nearest neighbors competence preservation learners. *Neurocomputing*, 308, 8-20.

Arslan, G., Karabulut, B. ve Unver, H. M. (2020). On Using Structural Patterns in Data For Classification. *Advances and Applications in Statistics*, 65(1), 33–56. doi:http://dx.doi.org/10.17654/AS065010033.

Awad, M. ve Khanna, R. (2015). *Efficient learning machines: theories, concepts, and applications for engineers and system designers*. Springer nature.

Bang, S. ve Jhun, M. (2014). Weighted support vector machine using k-means clustering. *Communications in Statistics-Simulation and Computation*, 43(10), 2307-2324.

Bang, S., Koo, J. Y. ve Jhun, M. (2010). Support vector machine using k-spatial medians clustering and recovery process. *Communications in Statistics-Simulation and Computation*, 39(7), 1422-1434.

Barto, A. G. ve Dietterich, T. G. (2004). Reinforcement learning and its relationship to supervised learning. *Handbook of learning and approximate dynamic programming*, 10, 9780470544785.

Bastani, H., Zhang, D. ve Zhang, H. (2020). Applied machine learning in operations management. *Available at SSRN 3736466*.

Baştanlar, Y. ve Özuysal, M. (2014). Introduction to machine learning. miRNomics: microRNA biology and computational analysis. *Methods in Molecular Biology (Methods and Protocols)*, 1107, 105-28.

Berrar, D. (2019). Cross-validation. In: Ranganathan, S., Gribskov, M., Nakai, K., Schönbach, C. (eds.) *Encyclopedia of Bioinformatics and Computational Biology*, 542–545. Academic Press, Oxford. doi:https://doi.org/10.1016/B978-0-12-809633-8.20349-X.

Bertini, J. R., Jr., Zhao, L. ve Lopes, A. A. (2013). An incremental learning algorithm based on the K-associated graph for non-stationary data classification. *Information Sciences*, 246, 52-68.

Bertini, J. R, Jr., Zhao, L., Motta, R. ve de Andrade Lopes, A. (2011). A nonparametric classification method based on k-associated graphs. *Information Sciences*, 181(24), 5435-5456.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer, New York.

Bonaccorso, G. (2017). *Machine learning algorithms*. Packt Publishing Ltd.

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

Burges, C. J. (1998). A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2), 121-167.

Byun, H. ve Lee, S. W. (2002). Applications of support vector machines for pattern recognition: A survey. *In International workshop on support vector machines* (pp. 213-236). Springer, Berlin, Heidelberg. August.

Camacho, D. M., Collins, K. M., Powers, R. K., Costello, J. C. ve Collins, J. J. (2018). Next-generation machine learning for biological networks. *Cell*, 173(7), 1581-1592.

Carleo, G., Cirac, I., Cranmer, K., Daudet, L., Schuld, M., Tishby, N., Vogt-Maranto, L. ve Zdeborová, L. (2019). Machine learning and the physical sciences. *Reviews of Modern Physics*, 91(4), 045002.

Çelik, S. ve Yılmaz, O. (2018). Prediction of body weight of Turkish tazi dogs using data mining Techniques: Classification and Regression Tree (CART) and multivariate adaptive regression splines (MARS). *Pakistan Journal of Zoology*, 50(2).

Chen, J. ve Pan, F. (2010). Clustering-based geometric support vector machines, p. 207–217. *In Proceedings of the Life System Modeling and Intelligent Computing*, Springer, Berlin, Heidelberg.

Chitrakar, R. ve Chuanhe, H. (2012). Anomaly detection using Support Vector Machine classification with k-Medoids clustering. *In 2012 Third Asian Himalayas International Conference on Internet* (pp. 1-5). IEEE. November.

Cunningham P., Cord M. ve Delany S. J. (2008) Supervised Learning. In: Cord M., Cunningham P. (eds) Machine Learning Techniques for Multimedia. Cognitive Technologies. Springer, Berlin, Heidelberg. doi: https://doi.org/10.1007/978-3-540-75171-7_2

Debnath, R., Takahide, N. ve Takahashi, H. (2004). A decision based one-against-one method for multi-class support vector machine. *Pattern Analysis and Applications,* 7(2), 164–175.

Dimitriadou, E., Hornik, K., Leisch, F., Meyer, D., Weingessel, A. ve Leisch, M. F. (2009). Package 'e1071'. *R Software package*, avaliable at https://cran.r-project.org/web/packages/e1071/e1071.pdf.

Dua, D. ve Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

Dy, J. G. ve Brodley, C. E. (2004). Feature selection for unsupervised learning. *Journal of machine learning research*, 5(Aug), 845-889.

Ezugwu, A. E., Shukla, A. K., Agbaje, M. B., Oyelade, O. N., José-García, A. ve Agushaka, J. O. (2020). Automatic clustering algorithms: a systematic review and bibliometric analysis of relevant literature. *Neural Computing and Applications*, 1-60.

Ferri, C., Hernández-Orallo, J. ve Modroiu, R. (2009). An experimental comparison of performance measures for classification. *Pattern Recognition Letters*, 30(1), 27-38.

Gaertler, M. (2005). Clustering. In Network analysis (pp. 178-215). Springer, Berlin, Heidelberg.

Gambella, C., Ghaddar, B. ve Naoum-Sawaya, J. (2021). Optimization problems for machine learning: A survey. *European Journal of Operational Research*, 290(3), 807-828.

Gan, J., Li, A., Lei, Q. L., Ren, H. ve Yang, Y. (2017). K-means based on active learning for support vector machine. *In 2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS)* (pp. 727-731). IEEE, Wuhan, China, 24-26 May.

Ghahramani, Z. (2003). Unsupervised learning. *In Summer School on Machine Learning* (pp. 72-112). Springer, Berlin, Heidelberg, February.

Greco, S., Matarazzo, B. ve Slowinski, R. (2001). Rough sets theory for multicriteria decision analysis. *European journal of operational research*, 129(1), 1-47.

Gu, Q. ve Han, J. (2013). Clustered support vector machines. *In Artificial intelligence and statistics* (pp. 307-315). PMLR, April.

Guha, S., Rastogi, R. ve Shim, K. (1998). CURE: an efficient clustering algorithm for large databases. *ACM Sigmod record*, 27(2), 73-84.

Guha, S., Rastogi, R. ve Shim, K. (2001). Cure: an efficient clustering algorithm for large databases. *Information systems*, 26(1), 35-58.

Han, J., Pei, J. ve Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier.

Harrington, P. (2012). *Machine learning in action*. Simon and Schuster.

Harris, T. (2015). Credit scoring using the clustered support vector machine. *Expert Systems with Applications*, 42(2), 741-750.

Horng, S. J., Su, M. Y., Chen, Y. H., Kao, T. W., Chen, R. J., Lai, J. L. ve Perkasa, C. D. (2011). A novel intrusion detection system based on hierarchical clustering and support vector machines. *Expert systems with Applications*, 38(1), 306-313.

Hossin, M. ve Sulaiman, M. N. (2015). A review on evaluation metrics for data classification evaluations. *International journal of data mining & knowledge management process*, 5(2), 1.

Hu, L. Y., Huang, M. W., Ke, S. W. ve Tsai, C. F. (2016). The distance function effect on k-nearest neighbor classification for medical datasets. *SpringerPlus*, 5(1), 1-9.

Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L. ve Herrera, F. (2011). Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. Journal of Multiple-Valued Logic & Soft Computing, 17, 255-287.

Jordan, M. I. ve Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245), 255-260.

Bertini Jr., J. R., Nicoletti, M. C. ve Zhao, L. (2017). Attribute-based Decision Graphs: A framework for multiclass data classification. *Neural Networks*, 85, 69-84.

Karabulut, B., Arslan, G. ve Ünver, H. M. (2021). Classification Based on Structural Information in Data. *Arabian Journal for Science and Engineering*, 1-15. doi:https://doi.org/10.1007/s13369-021-06177-3.

Karypis, G., Han, E. H. ve Kumar, V. (1999). Chameleon: Hierarchical clustering using dynamic modeling. *Computer*, 32(8), 68-75.

Kavzoglu, T. ve Colkesen, I. (2009). A kernel functions analysis for support vector machines for land cover classification. *International Journal of Applied Earth Observation and Geoinformation*, 11(5), 352-359.

Kayaalp, N. ve Arslan G. (2014). A fuzzy Bayesian classifier with learned Mahalanobis distance. *International Journal of Intelligent Systems*, 29(8), 713-726.

Khandelwal, M., Armaghani, D. J., Faradonbeh, R. S., Yellishetty, M., Abd Majid, M. Z. ve Monjezi, M. (2017). Classification and regression tree technique in estimating peak particle velocity caused by blasting. *Engineering with* computers, 33(1), 45-53.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. *In International Joint Conference on Artificial Intelligence (IJCAI)*, 14(2), 1137-1145.

Kotsiantis, S. B., Zaharakis, I. ve Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, 160(1), 3-24.

Kwon, D., Kim, H., Kim, J., Suh, S. C., Kim, I. ve Kim, K. J. (2019). A survey of deep learning-based network anomaly detection. *Cluster Computing*, 22(1), 949-961.

Larose, D. T. (2015). *Data mining and predictive analytics*. John Wiley & Sons.

Learn, S. (2017). Cross-validation: evaluating estimator performance. Available at: https://scikit-learn.org/stable/modules/cross_validation.html Erişim tarihi: 21 Haziran 2020].

Lee, S. J., Park, C., Jhun, M. ve Koo, J. Y. (2007). Support vector machine using K-means clustering. *Journal of the Korean Statistical Society*, 36(1), 175-182.

Lemm, S., Blankertz, B., Dickhaus, T. ve Müller, K. R. (2011). Introduction to machine learning for brain imaging. *Neuroimage*, 56(2), 387-399.

Lopes, A. A., Bertini J. R., Jr., Motta, R. ve Zhao, L. (2009). Classification based on the optimal k-associated network. *In International Conference on Complex Sciences* (pp. 1167-1177). Springer, Berlin, Heidelberg, February.

Tipping, M. E. (2000). The relevance vector machine. *In Advances in neural information processing systems*, 652-658.

Tipping, M. E. (2001). Sparse Bayesian learning and the relevance vector machine. *Journal of machine learning research*, 1(Jun), 211-244.

Jones, M. T. (2017). Models for machine learning, *IBM*, Available at: https://developer.ibm.com/articles/cc-models-machine-learning/

Mak, K. K., Lee, K. ve Park, C. (2019). Applications of machine learning in addiction studies: A systematic review. *Psychiatry research*, 275, 53-60.

Markoulidakis, I., Rallis, I., Georgoulas, I., Kopsiaftis, G., Doulamis, A. ve Doulamis, N. (2021). Multiclass Confusion Matrix Reduction Method and Its Application on Net Promoter Score Classification Problem. *Technologies*, 9(4), 81.

Meng, T., Jing, X., Yan, Z. ve Pedrycz, W. (2020). A survey on machine learning for data fusion. *Information Fusion*, 57, 115-129.

Mohammadi, M., Raahemi, B., Mehraban, S. A., Bigdeli, E. ve Akbari, A. (2015). An enhanced noise resilient K-associated graph classifier. *Expert Systems with Applications*, 42(21), 8283-8293.

Mohri, M., Rostamizadeh, A. ve Talwalkar, A. (2018). *Foundations of machine learning*. MIT press.

Morgan, J. (2014). Classification and Regression Tree Analysis. Technical Report No. 1, *Boston University School of Public Health Department of Health Policy & Management*, 8 May.

Novaković, J. D., Veljović, A., Ilić, S. S., Papić, Ž. ve Milica, T. (2017). Evaluation of classification models in machine learning. *Theory and Applications of Mathematics & Computer Science*, 7(1), 39-46.

Olivier, C., Schölkopf, B. ve Alexander, Z. (2006). *Semi-Supervised Learning (Adaptive Computation and Machine Learning)*, MIT Press, Cambridge.

Paper D. (2020). *Introduction to Scikit-Learn. In: Hands-on Scikit-Learn for Machine Learning Applications*. Apress, Berkeley, CA. doi: https://doi.org/10.1007/978-1-4842-5373-1_1.

Parvandeh, S., Yeh, H. W., Paulus, M. P. ve McKinney, B. A. (2020). Consensus features nested cross-validation. *Bioinformatics*, 36, 3093–3098.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... ve Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of machine Learning Research*, 12, 2825-2830.

Pitombo, C. S., de Souza, A. D. ve Lindner, A. (2017). Comparing decision tree algorithms to estimate intercity trip distribution. *Transportation Research Part C: Emerging Technologies*, 77, 16-32.

Reddy, G. T., Reddy, M. P. K., Lakshmanna, K., Kaluri, R., Rajput, D. S., Srivastava, G., ve Baker, T. (2020). Analysis of dimensionality reduction techniques on big data. I*EEE Access*, 8, 54776-54788.

Refaeilzadeh, P., Tang, L., & Liu, H. (2009). Cross-validation. *Encyclopedia of database systems*, 5, 532-538.

Reich, Y. ve Barai, S. V. (1999). Evaluating machine learning models for engineering problems. *Artificial Intelligence in Engineering*, 13(3), 257-272.

Rokach, L. ve Maimon, O. (2005). *Clustering methods. In Data mining and knowledge discovery handbook* (pp. 321-352). Springer, Boston, MA.

Salimi, A., Faradonbeh, R. S., Monjezi, M. ve Moormann, C. (2018). TBM performance estimation using a classification and regression tree (CART) technique. *Bulletin of Engineering Geology and the Environment*, 77(1), 429-440.

Sarker, I. H. (2021). Machine learning: Algorithms, real-world applications and research directions. *SN Computer Science*, 2(3), 1-21.

Saxena, A., Prasad, M., Gupta, A., Bharill, N., Patel, O. P., Tiwari, A., ... ve Lin, C. T. (2017). A review of clustering techniques and developments. *Neurocomputing*, 267, 664-681.

Sayed, G. I. ve Hassanien, A. E. (2017). Moth-flame swarm optimization with neutrosophic sets for automatic mitosis detection in breast cancer histology images. *Applied Intelligence*, 47(2), 397-408

Shalev-Shwartz, S. ve Ben-David, S. (2014). *Understanding machine learning: From theory to algorithms*. Cambridge university press.

Shamsollahi, M., Badiee, A. ve Ghazanfari, M. (2018). Using Combined Descriptive and Predictive Methods of Data Mining for Coronary Artery Disease Prediction: a Case Study Approach. *Journal of AI and Data Mining*, 7(1), 47-58.

Sinaga, K. P. ve Yang, M. S. (2020). Unsupervised K-means clustering algorithm. *IEEE Access*, 8, 80716-80727.

Soofi, A. A. ve Awan, A. (2017). Classification Techniques in Machine Learning: Applications and Issues. *Journal of Basic and Applied Sciences*, 13, 459-465.

Tharwat, A. (2020). Classification assessment methods. *Applied Computing and Informatics*, 17(1), 168-192. doi: https://doi.org/10.1016/j.aci.2018.08.003.

Verbraeken, J., Wolting, M., Katzy, J., Kloppenburg, J., Verbelen, T. ve Rellermeyer, J. S. (2020). A survey on distributed machine learning. *ACM Computing Surveys (CSUR)*, 53(2), 1-33.

Viswanath, P. ve Sarma, T. H. (2011). An improvement to k-nearest neighbor classifier. *In 2011 IEEE Recent Advances in Intelligent Computational Systems* (pp. 227-231). IEEE, September.

Wainer, J. ve Cawley, G. (2021). Nested cross-validation when selecting classifiers is overzealous for most practical applications. *Expert Systems with Applications*, 182, 115222.

Wang, J., Wu, X. ve Zhang, C. (2005). Support vector machines based on K-means clustering for real-time business intelligence systems. *International Journal of Business Intelligence and Data Mining*, 1(1), 54–64.

Wickham, M. (2018). *Practical Java Machine Learning: Projects with Google Cloud Platform and Amazon Web Services*. Apress.

Widera, P., Welsing, P. M., Ladel, C., Loughlin, J., Lafeber, F. P., Dop, F. P., ... ve Bacardit, J. (2020). Multi-classifier prediction of knee osteoarthritis progression from incomplete imbalanced longitudinal data. *Scientific reports*, 10(1), 1-15.

Widodo, A. ve Yang, B. S. (2007). Support vector machine in machine condition monitoring and fault diagnosis. *Mechanical systems and signal processing*, 21(6), 2560-2574.

Witten, I. H. ve Frank, E. (2002). Data mining: practical machine learning tools and techniques with Java implementations. *ACM Sigmod Record*. 31(1), 76–77.

Xiang, S., Nie, F. ve Zhang, C. (2008). Learning a Mahalanobis distance metric for data clustering and classification. *Pattern recognition*, 41(12), 3600-3612.

Xie, H., Zhang, L., Lim, C. P., Yu, Y., Liu, C., Liu, H. ve Walters, J. (2019). Improving K-means clustering with enhanced firefly algorithms. *Applied Soft Computing*, 84, 105763.

Xu, N. (2019). Understanding the reinforcement learning. *In Journal of Physics: Conference Series*, 1207(1), 012014. IOP Publishing.

Yao, Y., Liu, Y., Yu, Y., Xu, H., Lv, W., Li, Z. ve Chen, X. (2013). K-SVM: An Effective SVM Algorithm Based on K-means Clustering. *Journal of Computers*, 8(10), 2632-2639.

Yu, H., Yang, J. ve Han, J. (2003). Classifying large datasets using SVMs with hierarchical clusters. *In Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM. pp. 306–315.

Zacharis, N. Z. (2018). Classification and regression trees (CART) for predictive modeling in blended learning. *IJ Intelligent Systems and Applications*, 3, 1-9.

Zhang, B., Wei, Z., Ren, J., Cheng, Y. ve Zheng, Z. (2018). An empirical study on predicting blood pressure using classification and regression trees. *IEEE access*, 6, 21758-21768.

Zhang, X. D. (2020). *A matrix algebra approach to artificial intelligence* (pp. 1-820). Springer.

Zhou, Z. H. ve Li, M. (2010). Semi-supervised learning by disagreement. *Knowledge and Information Systems*, 24(3), 415-439.

ÖZGEÇMİŞ

Adı Soyadı : Bergen KARABULUT

Yabancı Dil : İngilizce

Eğitim Durumu :

Lisans : Erciyes Üniversitesi, 2008-2013
Bilgisayar Mühendisliği

Yüksek Lisans : Kırıkkale Üniversitesi, 2014-2016
Bilgisayar Mühendisliği

Çalıştığı Kurum/Kurumlar ve Yıl/Yıllar :

- Kırıkkale Üniversitesi, Bilgisayar Mühendisliği Bölümü, 2014-2019
  Araştırma Görevlisi
- TÜBİTAK BİLGEM, 2019-halen
  Uzman Araştırmacı

Yayınları (SCI-Expanded)

1. Karabulut, B., Arslan, G. ve Ünver, H. M. (2021). Classification Based on Structural Information in Data. *Arabian Journal for Science and Engineering*, 1-15. https://doi.org/10.1007/s13369-021-06177-3
2. Ayan, E., Karabulut, B. ve Ünver, H. M. (2021). Diagnosis of Pediatric Pneumonia with Ensemble of Deep Convolutional Neural Networks in Chest X-Ray Images. *Arabian Journal for Science and Engineering*, 1-17. https://doi.org/10.1007/s13369-021-06127-z

Yayınları (Diğer)

1. Arslan, G., Karabulut, B. ve Unver, H. M. (2020). On Using Structural Patterns in Data for Classification. *Advances and Applications in Statistics*, 65(1), 33–56. http://dx.doi.org/10.17654/AS065010033

2. Karabulut, B., Ergüzen, A. ve Ünver, H. M. A linear time pattern based algorithm for n-queens problem. *Politeknik Dergisi*, 1-1. doi:https://doi.org/10.2339/politeknik.762967

3. Karabulut, B., Arslan, G. ve Ünver, H. M. (2019). A Weighted Similarity Measure for k-Nearest Neighbors Algorithm. *Celal Bayar University Journal of Science*, 15(4), 393-400.

4. Cihan, Ş., Karabulut, B., Arslan, G. ve Cihan, G. (2018). Koroner arter hastalığı riskinin veri madenciliği yöntemleri ile incelenmesi. *International Journal of Engineering Research and Development*, 10(1), 85-93.

5. Cihan, Ş., Karabulut, B., Kokoç M., Arslan, G. ve Gürel, G. (2019). Analysis of Cryotherapy Treatment of Verruca by Machine Learning. *International Scientific and Vocational Studies Journal*, 3(2), 56-66.

6. Karabulut, B., Cihan, Ş., Ünver, H. M. ve Ergüzen, A. (2018). Electrolab: A New Dataset For Educational Data Mining. *Technological Applied Sciences*, 13(4), 318-328.

7. Karabulut, B. ve Cihan, Ş. (2018). Educational Data Mining Application for Increasing Quality in Engineering Education. *The Online Journal of Quality in Higher Education*, 5(3), 7.

Bildiriler

1. Unver, H. M. ve Karabulut, B. (2015). Keyboard Design Approach Based on Word Analysis for the Right-Handed. *The IRES 21st International Conference*, Amsterdam, Holland.

2. Karabulut, B., Cihan, Ş., Ünver, H. M. ve Ergüzen, A. (2017). Creating Laboratory Dataset for Educational Data Mining. *I. International Scientific and Vocational Studies Congress*, Cappadocia, Turkey, 05-08 October.

3. Cihan, Ş., Karabulut, B., Ergüzen, A. ve Ünver, H. M. (2017). Analyzing Students' Programming Performance Using CRISP-DM Model. *I. International Scientific and Vocational Studies Congress*, Cappadocia, Turkey, 05-08 October.

4. Karabulut, B., Kokoç, M., Ünver, H. M. ve Ersöz, S. (2017). Turkish Keyboard Layout using Clustering and Association Rules. *III. International Symposium on Multidisciplinary Studies (ISMS)*, Ankara, Turkey, 10-11 November.

5. Karabulut, B. ve Ünver, H. M. (2017). Word Completion: A Literature Review, *III. International Symposium on Multidisciplinary Studies (ISMS)*, Ankara, Turkey, 10-11 November.

6. Unver, H. M., Karabulut, B. ve Kökver, Y. (2017). Development of an Ergonomic Keyboard with Current Letter Layout for Turkish Language. *International Conference on Science, Technology, Engineering and Management*, Vienna, Austria.

Araştırma Alanları :

Veri Madenciliği, Makine Öğrenmesi, İstatistiksel Veri Analizi, E-İmza Teknolojileri