T.C.
İSTANBUL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Yüksek Lisans Tezi

TOPLULUK ÖĞRENMESİ YÖNTEMİ İLE MİKRODİZİ VERİ ANALİZİ

Tcharé Adnaane BAWA

Enformatik Anabilim Dalı

Enformatik Programı

DANIŞMAN
Doç. Dr. Çiğdem EROL

Şubat, 2022

İSTANBUL

Bu çalışma, 9.02.2022 tarihinde aşağıdaki jüri tarafından Enformatik Anabilim Dalı, Enformatik Programında Yüksek Lisans tezi olarak kabul edilmiştir.

Tez Jürisi

Doç. Dr. Çiğdem EROL (Danışman)
İstanbul Üniversitesi
Fen Fakültesi

Prof. Dr. Sevinç GÜLSEÇEN
İstanbul Üniversitesi
Enformatik Bölümü

Dr. Öğr. Üyesi Yalçın Özkan
İstinye Üniversitesi
İktisadi, İdari ve Sosyal Bilimler Fakültesi

İntihal Programı Beyanı

20.04.2016 tarihli Resmi Gazete'de yayımlanan Lisansüstü Eğitim ve Öğretim Yönetmeliğinin 9/2 ve 22/2 maddeleri gereğince; Bu Lisansüstü teze, İstanbul Üniversitesi'nin abonesi olduğu intihal yazılım programı kullanılarak Fen Bilimleri Enstitüsü'nün belirlemiş olduğu ölçütlere uygun rapor alınmıştır.

Proje Destekleri

Bu tez, 4590 numaralı Türkiye Sağlık Enstitüleri Başkanlığı (TÜSEB) projesi ile desteklenmiştir.

Tezden Üretilmiş Yayınların Künye Bilgileri

ÖNSÖZ

Yüksek lisans projemin başlangıcından beri beni sıcak bir şekilde karşılayan, her koşulda ulaşılabilirliği sağlayan, bana her zaman destek olan ve tecrübeleriyle öğrenme fırsatı sunan, tezimin gerçekleşmesi sırasında beni motive eden, olumlu düşünceleri ile beni olumlu etkileyen ve ilham veren sabırlı, değerli danışmanım Doç. Dr. Çiğdem SELÇUKCAN EROL'a tüm içtenliğimle teşekkür ederim.

Makine öğrenimi ve biyoenformatik konusunda tecrübesiyle bana yardım eden ve benimle en iyi fikirlerini paylaşan Dr. Yalçın ÖZKAN'a çok teşekkür ederim.

Öğretisi, çalışması ve çalışkanlığıyla beni bu heyecan verici makine öğrenimi ve yapay zekâ dünyası ile tanıştıran Dr. Murat GEZER hocama çok teşekkür ederim.

Bölüm Başkanı ve aynı zamanda hocam olan Prof. Dr. Sevinç GÜLSEÇEN ve tüm öğretim üyelerine, beni bölümlerinde sıcakkanlı bir şekilde karşıladıkları ve bu alandaki bilgilerimi öğrenmemi ve geliştirmemi sağlayan destekleri için çok teşekkür ediyorum.

Hayatımda, her koşulda yanımda olup bana destek olan ve yol gösteren babama ve anneme çok teşekkürler.

Şubat 2022 Tchare Adnaane BAWA

İÇİNDEKİLER

Sayfa No

ŞEKİL LİSTESİ

Sayfa No

TABLO LİSTESİ

Sayfa No

SİMGE VE KISALTMA LİSTESİ

Kısaltmalar **Açıklama**

SNP : Tek Nükleotid Polimorfizmleri

EDA : Dağılım Algoritmalarının Tahmini (*Estimation of Distribution Algorithms*)

NCBI : *The National Center for Biotechnology Information*

SVM : Destek Vektör Makinesi

RBF : Radyal tabanlı fonksiyon (*Radial basis function*)

CART : Sınıflandırma ve Regresyon Ağacı (*Classification and Regression Tree*)

LDA : Lineer Diskriminant Analizi(*Linear Discriminant Analysis*)

PLS : Kısmi En Küçük Kare (*Partial Least Square*)

mRNA : Mesajcı RNA (*messenger RNA*)

cDNA : Tamamlayıcı DNA (*complementary DNA*)

PM : Tam eşleşen (*Perfect Match*)

LOWESS/LOESS: Yerel Ağırlıklı Düzgünleştirme Dağılım Grafikleri (*Locally Weighted Smoothing Scatterplots*)

ÖZET

YÜKSEK LİSANS TEZİ

TOPLULUK ÖĞRENMESİ YÖNTEMİ İLE MİKRODİZİ VERİ ANALİZİ

Tcharé Adnaane BAWA

İstanbul Üniversitesi

Fen Bilimleri Enstitüsü

Enformatik Anabilim Dalı

Danışman: Doç. Dr. Çiğdem EROL

Topluluk öğrenme yöntemi, tek bir algoritmadan elde edilenden daha iyi bir sınıflandırma veya tahmin sonucuna sahip olmak için birçok makine öğrenimi algoritmasını birleştirmekten oluşan bir makine öğrenimi tekniğidir. Son zamanlarda, topluluk öğrenme, çoğunlukla örüntü tanıma, öznitelik seçimi ve sınıflandırma gibi farklı amaçlar için birçok alanda kullanılan yoğun ve yaygın bir makine öğrenimi tekniği olarak görülmektedir. Torbalama (*Bagging*), Yükseltme (*Boosting*), Rasgele Orman (*Random Forest*), İstifleme (*Stacking*) en popüler ve yaygın olarak kullanılan topluluk öğrenme yöntemleridir. Topluluk öğrenmesinin avantajlarından biri, yüksek boyutlu ve karmaşık veri yapılarıyla başa çıkma yeteneğidir, yani topluluk öğrenmesi, veri kümelerinin varyansını azaltmaya yardımcı olur ve daha doğru sonuçlar sağlar.

Mikrodizi analizi, hastalıkların erken tespiti ve sınıflandırılmasına, hastalıklarla ilgili genlerin veya biyobelirteçlerin bulunmasına ve ilaçların bulunmasına yardımcı olabilecek sonuçlar sağlamayı amaçlayan bir biyoenformatik yöntemdir. Bu nedenle, mikrodizi analizi ile elde edilen verinin çok dikkatli bir şekilde analiz edilmesi gerekmektedir. Mikrodizi veri analizinde, hastalık durumlarını tahmin etmek, hastalıkla ilgili genleri bulmak ve ayrıca örüntü tanımak amacıyla makine öğrenmesi algoritmaları kullanılmaktadır. Ancak, bu algoritmaların karşılaştığı yaygın büyük sorunlardan biri, küçük örnek boyutu ve mikrodizi veri setlerinin yüksek boyutluluğudur, çoğunlukla modellerin aşırı uyum göstermesine yol açan yüksek bir varyansın varlığı ile karakterize edilir ve bahsedilen yüksek varyans sorunu, birçok çalışmada vurgulanmıştır.

Bu tezin amacı, topluluk öğrenme yöntemlerini kullanarak mikrodizi verisetlerindeki varyansı azaltmaya çalışmaktır. Bu amaçla NCBI GEO veri tabanında yer alan GSE19804 kodlu küçük hücreli olmayan akciğer kanseri veri seti kullanılmıştır. Bu çalışma çerçevesinde iki tür makine öğrenmesi modeli uygulanmıştır; biri istifleme topluluk öğrenmesine ve diğeri bir algoritmaya dayalıdır. Analizin başında seçilen 12 algoritmadan 7'si yani Radyal Destek Vektör Makinesi, Doğrusal Ayırıcı Analizi, k-En Yakın Komşular, C5.0, CART, Öznitelik Çıkarımlı Sinir Ağları, Genelleştirilmiş Doğrusal Model algoritmaları analizde kullanılmıştır. Sonuç olarak, analizlerimiz istifleme topluluk öğrenme modellerinin basit modellere kıyasla daha düşük varyansa sahip olduğunu göstermiştir.

Şubat 2022, 88 sayfa.

Anahtar kelimeler: Topluluk öğrenme, makine öğrenme, veri madenciliği, mikrodizi veri analizi, küçük hücreli olmayan akciğer kanseri.

SUMMARY

M.Sc. THESIS

MICROARRAY DATA ANALYSIS

WITH

ENSEMBLE LEARNING METHODS

Tchare Adnaane BAWA

İstanbul University

Institute of Graduate Studies in Sciences

Department of Informatics

Supervisor : Assoc. Prof. Dr. Çiğdem EROL

The Ensemble learning method is a machine learning technique that consists of combining many machine learning algorithms in order to have a better classification or prediction result than can be obtained from a single algorithm. Recently, ensemble learning has been shown as an intensive and widespread machine learning technique that is mostly used in many areas for different purposes such as pattern recognition, feature selection, and classification. Bagging, Boosting, Stacking, Random Forest are the most popular and widely used ensemble learning methods. One of the advantages of ensemble learning is the ability to deal with high-dimensional and complex data structures, that is to say ensemble learning helps reduce the variance of datasets and provides more accurate results.

Microarray analysis is a bioinformatics method that aims to provide results that can help early detection and classification of diseases, to find genes or biomarkers related to diseases, and to drugs. Therefore, the data obtained by microarray analysis should be analyzed very carefully. In microarray data analysis, machine learning algorithms are used to predict disease states, to find disease-related genes and also to recognize patterns. However, one of the common major problems faced by these algorithms is the small sample size and the high dimensionality of the microarray datasets, often characterized by the presence of a high variance leading to the overfitting of models, and that high variance problem aforementioned has been highlighted in many studies.

The aim of this thesis is to try to reduce the variance in microarray dataset by using ensemble learning methods. For this purpose, non-small cell lung cancer dataset (GSE19804) in the NCBI GEO database was used. Within the framework of this study, two types of machine learning models were applied; one is based on stacking ensemble learning and the other is based on simple algorithm. Seven of the 12 algorithms selected at the beginning of the analysis, namely Radial Support Vector Machine, Linear Discriminant Analysis, k-Nearest Neighbors, C5.0, CART, Feature Extraction Neural Networks, Generalized Linear Model algorithms were used in the analysis. As a result, stacking ensemble learning models showed lower variance compared to simple algorithm-based models.

February 2022, 88 pages.

Keywords: Ensemble learning, machine learning, data mining, microarray data analysis, non-small lung cancer.

1 GİRİŞ

Makine öğrenmesi teknikleri, farklı veri türlerini analiz etmek için yaygın olarak kullanılmaktadır (Molla ve diğ., 2004; Pirooznia ve diğ., 2008; Panca ve Rustam, 2017; Sontakke ve diğ., 2017; Turgut ve diğ., 2018). Topluluk öğrenmesi, farklı makine öğrenmesi algoritmalarını veya modellerini ortak bir amaca hizmet etmek için birlikte kullanıldığında kullanılan bir terimdir (Opitz ve Maclin, 1999). Geçtiğimiz birkaç yıl içinde, topluluk sistemleri veya topluluk yöntemleri olarak da adlandırılan topluluk öğrenmesi, hesaplamalı zekâ ve makine öğrenmesi topluluğu içinde çok fazla dikkat çekmeye başlamıştır (Khoshgoftaar ve diğ., 2013). Sınıflandırma, öznitelik seçimi, güven tahmini, eksik öznitelik, kademeli öğrenme, hata düzeltme, sınıf dengesizliği ve varyans azaltma gibi çeşitli sorunları çözmek için birçok alanda kullanılırlar- böylece hataları azaltıp doğruluğu artırır (Khoshgoftaar ve diğ., 2013; Eldardiry ve diğ., 2020).

Topluluk öğrenmesi, elde edilen sonucun stabilizasyonuna katkıda bulunan bir tür makine öğrenmesidir ve biyobelirteçlerin keşfi, yüksek boyutlu veriden gen seçimi gibi uygulamalarda dahil olmak üzere, hesaplamalı biyolojinin biyomedikal uygulamalarında önemli bir konusudur (Abeel ve diğ., 2010). Kanser teşhisi için sağlam biyobelirteçlerin tanımlanmasından oluşan bir çalışmada, Abeel ve diğ. (2010) topluluk öznitelik seçme yöntemleriyle (*Ensemble feature selection*) seçilen biyobelirteçlerde yaklaşık %30'a varan bir artış ve sınıflandırma performansında ~ %15'lik bir iyileşme göstererek kanserle ilgili 4 mikrodizi üzerinde bir etki göstermiştir. Sonuç olarak biyobelirteçlerin stabilitesini iyileştirmek için topluluk öğrenmesi yöntemlerin kullanılmasını önermektedir (Abeel ve diğ., 2010).

Birçok makine öğrenimi tekniği, zorlu sorunlardan muzdariptir. Çeşitli faktörler olsa da öznitelik sayıları veya boyutluluk ana faktörlerdendir Veri boyutluluğu, bir öğrenme sisteminin hem eğitim hem de çalışma zamanı aşamalarını etkiler. Bu arada, çok boyutluluk sorunlara neden olabilir. Öznitelik seçimi, boyutluluk problemlerini çözmek için kullanılan bir tekniktir. Alakasız ve gereksiz öznitelikleri kaldırarak yalnızca yararlı öznitelikleri korur ve sınıflandırma performansını iyileştirmek, algoritmayı hızlandırmak ve en alakalı özniteliklerin yorumlanması yoluyla soruna iyi bir içgörü dahil olmak üzere birçok fayda sağlar (Guan ve diğ., 2014). Son zamanlarda, topluluk öğrenme tabanlı öznitelik seçimi olarak adlandırılan, öznitelik seçimi ve

topluluk öğrenmeyi bütünleştiren yeni bir öznitelik seçimi yaklaşımı önerilmiş ve çalışılmıştır. Birçok çalışma, bu toplu öğrenmeye dayalı öznitelik seçme yönteminin yalnızca basit öznitelik seçme işlemlerinden daha iyi performans gösterdiğini göstermiştir (Guan ve diğ., 2014; Bolón-Canedo ve Alonso-Betanzos, 2019).

Mikrodizi veri setleri, genellikle sınıflandırma ve kümeleme gibi makine öğrenimi görevlerinde çeşitli zorluklara neden olan son derece yüksek boyut ve küçük örneklem sayılarından oluşmaktadır (Guan ve diğ., 2014). Kanser mikrodizi veri analizinde önemli sorunlardan biri, DNA mikrodizi sınıflandırmasının doğruluğudur. Tek bir modelin sağladığı sonuca güvenmek yerine, birçok çalışma, farklı uzman sınıflandırıcıların sonuçlarının *(expert classifiers)* birleştirilmesinin daha iyi bir sonuç sağladığını göstermiştir (Safiyari ve Javidan, 2017; Wang ve diğ., 2019). Bununla birlikte, bu yöntemler, hata korelasyonları nedeniyle bazen performans açısından sınırlıdır. Ana zorluk, DNA mikrodizilerinin örneklem sayısı az olması ve öznitelik sayısı fazla olmasıdır (Abdulla ve Khasawneh, 2020). Korelasyon analizi öznitelik seçimine dayalı bir yöntem kullanarak, bileşenleri birbiriyle negatif veya tamamlayıcı şekilde ilişkili olan farklı öznitelik setlerinden öğrenilen iki toplu sınıflandırıcı, üç kıyaslama veri setinde en iyi tanıma oranlarını üretir (Kim ve Cho, 2006). Bununla birlikte, bir sonuç vermek için birlikte katkıda bulunan birçok algoritmaya dayalı modeller kullanmak, DNA mikrodizi veri setinde bulunan varyans problemini azaltmak gibi sonucun birçok yönünü iyileştirmeye yardımcı olabilir.

Bu tezin GENEL KISIMLAR bölümünde öncelikle çalışmanın iki ana konusu olan TOPLULUK ÖĞRENME ve MİKRODİZİ anlatılmaktadır. İlk ana konu olarak, topluluk öğrenme hakkında genel bilgiler, topluluk öğrenmede önişleme dahil veri analizi aşamalarının açıklanması, varyans, analizlerde kullanılan algoritmalar ve diğer makine öğrenmesi yöntemlerinden bahsedilmiştir. Aynı bölümde ikinci ana konu olan Mikrodizi başlığında tanımı, kullanım amaçları, mikrodizi verisinin elde edilme süreci ve mikrodizi verilerinin ön işleme dahil tüm veri analizi sürecinden bahsedilir. Sonrasında bu çalışma çerçevesinde topluluk öğreneme modellerinin oluşturulmasına müdahale algoritmaları, modellerin varyansını değerlendirme yöntemlerine ilişkin bilgilere yer verilmiştir. Genel kısımlar bölümünün sonunda, mikrodizi sayesinde belirli hastalıklarda topluluk öğreneme yöntemlerinin uygulanmasına yönelik yapılan çalışmaları içeren literatürün bir incelemesi verilmiştir. MATERYAL VE YÖNTEM bölümü, mikrodizi verilerinin seçilmesi ve elde

edilmesi, bunların görselleştirilmesi, ön işleme ve veri seti olarak kullanılmaya hazır ifade edilen genlerin elde edilmesi ile ilgili derinlemesine ayrıntılar verir. Daha sonra topluluk öğreneme'nin varyans üzerindeki etkisini ispatlamak için elde edilen veri setine uygulanan farklı yöntemler aktarılmaktadır. BULGULAR bölümünde, R programlama dili kullanılarak gerçekleştirilen analizlerin performanslarının karşılaştırmalı değerlendirmeleri sonucu elde edilen bulgular tablo ve şekil olarak verilmiştir. TARTIŞMA VE SONUÇ bölümünde ise bulgular, literatür ışığında tartışılarak gelecekteki araştırmalar için öneriler sunulmaktadır.

2 GENEL KISIMLAR

Bu tez çalışmasının genel kısımlar bölümü, küçük hücreli olmayan akciğer kanseri, mikrodizi ve makine öğrenmesi olmak üzere temel olarak 3 ana kısma ayrılarak aktarılmaya çalışılmıştır.

2.1 KÜÇÜK HÜCRELİ OLMAYAN AKCİĞER KANSERİ

Akciğer kanseri dünyanın birçok ülkesinde önde gelen ölüm nedenidir (Jemal ve diğ., 2010). Histolojik yapısına bağlı olarak küçük hücreli ve küçük hücreli olmayan akciğer kanseri olmak üzere iki gruba ayrılır. Küçük hücreli olmayan akciğer kanseri (NSCLC), yaklaşık %5, beş yıllık sağkalım oranı ile akciğer kanserlerinin yaklaşık %85'ini oluşturan en yaygın olanıdır (Mendoza ve diğ., 2020). NSCLC'nin adenokarsinoma (%40) bronkoalveolar karsinomu da içeren skuamöz hücreli karsinom (%30-35) ve büyük hücreli karsinom (%5-15) olmak üzere 3 alt tipi bulunmaktadır (Ma ve diğ., 2005). Kanser teşhisinden sonra kanserin yayılmasına göre evreleme yapılmaktadır. NSCLC, sıfırdan dörde kadar beş evreye sahiptir. Hastalığın ciddiyeti sıfırdan başlayıp giderek artmaktadır.

NSCLC ile mücadele kapsamında birçok çalışma yapılmış ve çeşitli yöntemler geliştirilmiştir, ancak karşılaştıkları ana engel ilaç direncinin gelişmesi veya hastalığın geç saptanmasıdır (Sun ve diğ., 2012). NSCLC tümörlerinin genomik taraması, hedeflenen tedavilere karşı kazanılmış direncin moleküler mekanizmalarının tanımlanmasını kolaylaştırmaya devam edecektir. Bu nedenle, NSCLC'de yer alan genleri ve rollerini bulmak bu hastalığın üstesinden gelmeye yardımcı olabilir.

Mikrodizi teknikleri ile büyük miktarda veri sağlanarak NSCLC ile mücadeleye yardımcı olunmaktadır (Russo ve diğ., 2003).

2.2 MİKRODİZİ

Mikrodizi, birçok genin aynı anda ifadesini analiz etmek için laboratuvarlarda kullanılan bir tür araçtır. Bir mikrodizi binlerce küçük noktadan oluşur ve bu noktalar prob olarak adlandırılmaktadır. Her prob, iyi bilinen bir DNA dizisi veya geni içerir (Govindarajan ve diğ., 2012). Genellikle, mikrodizi slaytları, gen çipleri, DNA çipleri veya biyoçipler olarak adlandırılır. Her slayta eklenen her DNA molekülü, aynı zamanda transkriptom veya bir grup gen tarafından ifade edilen mesajcı RNA *(mRNA)* transkript seti olarak da bilinen gen

ekspresyonunu saptamak için prob görevi görür. DNA mikrodizisi, protein mikrodizisi, Peptit mikrodizisi, antikor mikrodizisi olmak üzere farklı mikrodizi türleri vardır(Heller, 2002; Templin ve diğ., 2002; Angenendt, 2005; Lin ve diğ., 2016).

Mikrodiziler, hastalık prognozu ve teşhisi, ilaç keşfi, toksikoloji, yaşlanma, beyin hastalığı ve akciğer hastalığı gibi alanlara eşi görülmemiş anahtar erişim sağlar (Özkan ve Erol, 2015). Tarama sonrası mikrodizi görüntü analizinden elde edilen veriler, istatistik, veri madenciliği ve makine öğrenmesi yöntemleri kullanılarak analiz edilebilir. Günümüzde çok sayıda mikrodizi üreticisi bulunmaktadır; en popüler olanları Affymetrix, Agilent ve Illumina'dır. Şekil 2.1'de bir mikrodizi çip örneği görülmektedir.

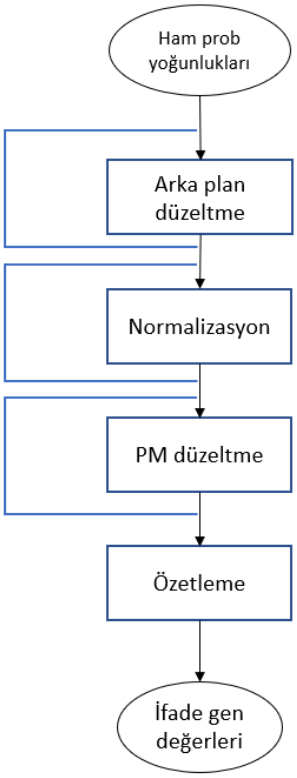
Şekil 2.1: Mikrodizi çip örneği (*URL-1*).

Bir mikrodizi analizinin gerçekleştirilmesi problemin tanımlanması, biyolojik bir şekilde bir deneyin hazırlanması, mikrodizinin oluşturulması, görüntünün analizi ve elde edilen verinin analizi olmak üzere 5 farklı aşamadan oluşmaktadır (Özkan ve Erol, 2015):

1. **Problem tanımı**: örneğin, sigaranın akciğer kanseri üzerinde herhangi bir etkisi var mı?

2. **Biyolojik yolla bir deneyin hazırlanması:** Bu aşamada sigara içen ve içmeyen kişiden numune alınacaktır. Sigara içilen numune deneysel numunedir ve sigara içilmeyen numune kontrol numunesini temsil eder. Bu örnekler daha sonra mikrodizi deneyi için hazır hale getirecektir.
3. **Mikrodizi deneyi**: Birinci faz deneyi sırasında, deneysel ve referans numuneden alınan mRNA örnekleri tamamlayıcı DNA'ya (cDNA) dönüştürülür. Bir floresan probla (örneğin kırmızı ve mavi) etiketlenen her numune daha sonra mikrodizi lamına bağlanmalarını sağlamak için birlikte karıştırılır. Bu işleme hibridizasyon da denir. Daha sonra, slayt üzerine basılan her bir genin ifadesini ölçmek için mikrodizi taranır.
4. **Görüntü işleme:** mikrodizi slaydı oluşturulduktan sonra dijitalleştirme için taranır. Dijitalleşme sırasında gürültü ve hibridizasyon sorunları gibi farklı sorunlar olabilir. Bu sorunların üstesinden gelmek için arka plan düzeltme, normalleştirme gibi farklı yöntemler oluşturulmuştur. Ön işlemeden sonra örnekler bu aşamanın sonunda sayısal verilere dönüştürülür.
5. **Veri analizi:** Elde edilen sayısal ham veri seti daha sonra makine öğrenmesi veri analizi için hazırdır.

Görüntü verisinden, verinin analize hazır hale getirme süreci Şekil 2.2’de yer almaktadır.



Şekil 2.2: Affymetrix mikrodizi ön işleme süreci (Özkan ve Erol, 2015).

Şekil 2.2'te gösterildiği gibi, Affymetrix mikrodizi veri setinin ön işleme süreci 4 farklı adımdan oluşur: Arka plan düzeltme, Normalizasyon, PM *(Perfect Match)* düzeltme ve Özetleme.

Arka plan düzeltme, mikrodizi veri ön işlemenin en önemli adımlarından biridir. Mikrodizi görüntüsü tarandıktan sonra elde edilen verilerde bazen bazı sorunlar olabilir. Örnek olarak, farklı olabilen parlaklık veya bazı probların diğerlerinden daha uzun olabilmesi ve dolayısıyla herhangi bir sinyal vermemesidir. Affymetrix, bu tür sorunların üstesinden gelmek için RMA ve MAS5 gibi farklı arka plan düzeltme yöntemleri sunar (Hochreiter ve diğ. , 2006; Irizarry ve diğ., 2006).

Normalleştirme, ölçülen gen ekspresyon seviyelerini etkileyen bazı varyasyon kaynaklarını ortadan kaldırmayı amaçlayan mikrodizi veri ön işlemenin en önemli adımlarından biridir (Park ve diğ., 2003). Bu varyasyonlar, RNA hibridizasyonunun kalitesinden kaynaklanabilir. En popüler olanları Lowess ya da Loess *(LOcally Weighted Smoothing Scartterplot)* ve nicelik

normalleştirme yöntemleri olan farklı normalleştirme yöntemleri vardır (Calza ve Pawitan, 2010; Özkan ve Erol, 2015).

Mikrodizi oluşturma sırasında, spesifik olmayan hibridizasyonlar gerçekleşebilir ve böylece araştırılan verilerdeki gürültüye katkıda bulunabilir. Bu sorunun üstesinden gelmek için PM (Perfect Match) düzeltmesi yapılabilir. PM düzeltme, mikrodizi verilerindeki gürültüye katkıda bulunan spesifik olmayan hibridizasyonun etkisini azaltmayı amaçlayan bir işlemdir(Babichev ve diğ., 2016). MAS ve Substractum, Affymetrix tarafından sağlanan PM düzeltme yöntemleridir.

Özetleme, mikrodizi ön işleme adımlarının son adımıdır. Her gen için önceden işlenmiş çoklu prob yoğunluklarının tek bir ekspresyon değeriyle birleştirilmesinden oluşur (Hochreiter ve diğ., 2006). Affymetrix mikrodizi çipleri için kullanılan bazı yöntemler *Tukey Bi-Weight* ve Medyan parlatmadır – *Median Polishing* (Özkan ve Erol, 2015).

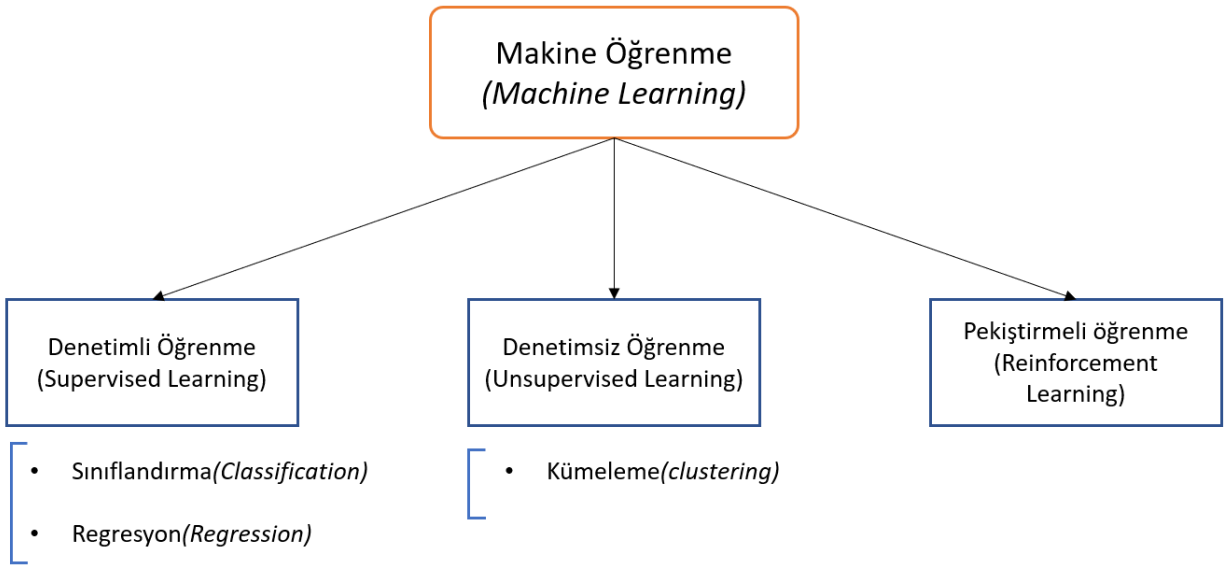
Bu tez çalışmasında mikrodizi deneylerinden elde edilen ve işlenerek gen ifade değerlerine dönüştürülen hazır veri seti kullanılarak, beşinci aşama olan veri analizi aşaması gerçekleştirilmiştir.

2.3 MAKİNE ÖĞRENMESİ

Hem makine öğrenimi hem de veri madenciliği, yararlı bilgiler elde etmek için veri ile ilgilenen bilim dallarıdır. Makine öğrenmesi, büyük veri kümelerini analiz etmeyi ve daha sonra gelecekteki yeni veri kümeleri hakkında tahminlerde bulunmak için bunlardan kalıplar bulmayı amaçlayan bir Yapay Zekâ alt alanıdır (El Naqa ve Murphy, 2015). Teknolojinin evrimi ile makine öğrenimi, eczacılık, sağlık, mekanik, elektronik ve inşaat gibi birçok endüstriye ve alana entegre edilmektedir.

Makine öğrenimi, denetimli (*supervised*), denetimsiz (*unsupervised*) ve pekiştirmeli (*reinforcement*) öğrenme olmak üzere birkaç alandan oluşur (Han ve diğ., 2011; Dey, 2016) (Şekil 2.3). Denetimli öğrenme, örnek girdi-çıktı çiftlerine dayalı olarak bir girdiyi bir çıktıya eşleyen bir işlevi öğrenme görevine sahip makine öğrenmesinin bir alt kategorisidir (Cunningham ve diğ. , 2008; Ashfaq ve diğ., 2017). Bir dizi eğitim örneğinden oluşan etiketli eğitim verilerinden bir işlev çıkarır. Girdi verisi ile model beslenir. Denetimli öğrenmedeki iki temel görev Sınıflandırma ve Regresyondur (Jiang, ve diğ., 2020). Denetimsiz öğrenme,

kullanıcıların modeli denetlemesine/etiketlemesine gerek olmayan bir makine öğrenmesi dalıdır. Bunun yerine, modelin daha önce tespit edilmemiş kalıpları ve bilgileri keşfetmesi için kendi başına çalışmasına izin verir (Han ve diğ., 2011). Esas olarak etiketlenmemiş veri ile ilgilenir. Denetimsiz makine öğreniminin bazı örnekleri, anomali tespiti ve öznitelik seçimidir. Pekiştirmeli öğrenme, kümülatif ödül kavramını en üst düzeye çıkarmak için akıllı aracıların bir ortamda nasıl harekete geçmesi gerektiğiyle ilgili bir tür makine öğrenmesidir (Nian ve diğ., 2020). Şekil 2.2'de de görüldüğü gibi makine öğrenmesi, tahmine yönelik olarak regresyon ve sınıflandırma alt alanlarını içeren denetimli öğrenme, kümeleme yaklaşımını kullanan denetimsiz öğrenme ve pekiştirmeli öğrenme olmak üzere üç alandan oluşmaktadır.

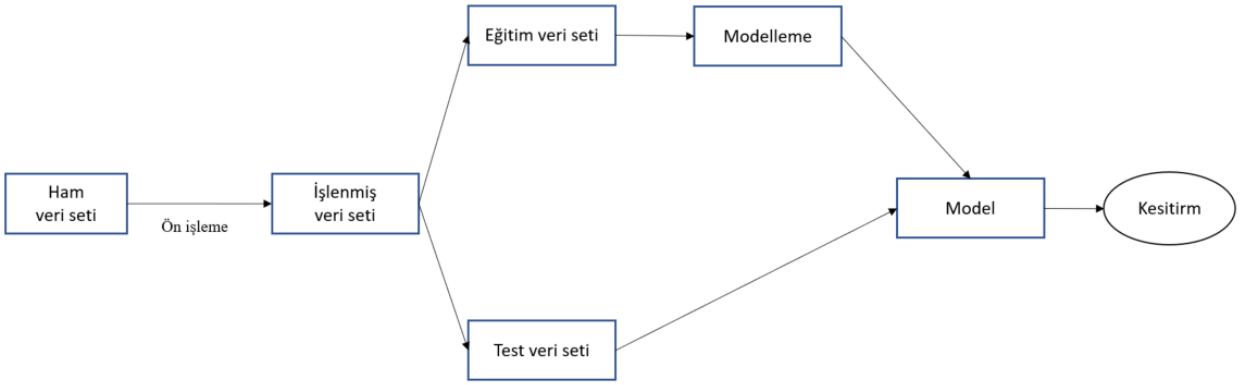


Şekil 2.3: Makine öğrenimi ve onun oluşturan farklı alanlar (Han, Pei ve diğ. , 2011).

2.4 SINIFLANDIRMA

Denetimli makine öğrenmesi, sınıflandırma ve regresyon olmak üzere iki bölümden oluşmaktadır. Regresyon görevleri, modellerin bir veya birden fazla yordayıcı değişken veya özelliğin (x) değerine dayalı olarak sürekli bir sonuç değişkenini (y) tahmin etmesini sağlayan bir makine öğrenmesi yönteminden oluşur (Veloso de Melo ve Banzhaf, 2018). Sınıflandırmaya gelince, önemli veri sınıflarını tanımlayan modelleri çıkaran bir veri analizi biçimidir (Gola ve diğ., 2018). Çoğunlukla sınıflandırıcılar olarak adlandırılan bu tür modeller, verilen verilere dayalı olarak kategorik (ayrık, sırasız) sınıf etiketlerini tahmin eder (Chollet, 2018).

Makine Öğreniminde farklı alanlar vardır ve bunlardan biri olan denetimli sınıflandırmanın amacı, etiketlenmiş veri kümesine dayalı özlü bir model oluşturmaktır. Denetimli öğrenme sınıflandırma görevlerinde, veri kümesindeki her örnek, aynı öznitelik kümesiyle kullanılan algoritma tarafından temsil edilir (Han ve diğ. , 2011). Bu öznitelikler kategorik, sürekli veya ikili olabilir (Kotsiantis ve diğ., 2006). Ortaya çıkan sınıflandırıcı (genellikle model olarak adlandırılır) daha sonra, tahmin edici özniteliklerin değerlerinin bilindiği yeni örneklerin sınıf etiketlerini tahmin etmek için kullanılır. Tipik bir denetimli makine öğrenimi sınıflandırma işlemi, Şekil 2.4'de yer alan adımları içermektedir (Özkan ve Erol, 2015):



Şekil 2.4: Denetimli makine öğrenme süreci (Özkan ve Erol, 2015).

- ***Veri ön işleme***

Verileri (ham veri) topladıktan sonra, makine öğrenmesi algoritmaları için kullanılabilir hale getirmek için işlenmeleri gerekir. Ön işleme adı verilen bu süreç, ham verileri kullanışlı hale getirmek için farklı istatistiksel yöntemlerin kullanılmasından oluşur. Eksik değerlerin doldurulması, gürültülü değerlerin düzeltilmesi, gereksiz özniteliklerin kaldırılması, özniteliklerin kodlanması, verilerin ölçeklenmesi ve aykırı değerlerin tespit edilmesi gibi prosedürler, veri ön işleme sırasında bulunabilir (Al-Jarrah ve diğ., 2014). Bu süreç, model oluşturmanın en zor ve yorucu adımlarından biridir ve bu süreçte başarılı olabilmek için veri bilimciler veya mühendisler veri setlerini çok iyi bilmeleri veya verilerle ilgili alanında uzman bir kişi ile çalışmaları gerekmektedir(Özkan ve Erol, 2015).

- ***Modelleme***

Ön işleme aşamasını tamamladıktan hemen sonra, veri seti artık modelleme için kullanışlıdır. Modelleme için sadece algoritmanın seçimi değil, aynı zamanda o algoritmanın parametresi de

önemlidir çünkü mevcut verilere uygun olmayan bir algoritma veya yanlış parametrelere sahip bir algoritma fazla veya eksik uydurmaya neden olabilir (Han ve diğ., 2011). Bu nedenle seçilen algoritmanın doğru olduğundan ve optimal parametrelerle yapılandırıldığından emin olmak önemlidir. Bunu yapmak için, metrik adı verilen ve mevcut olan iki temel yöntem vardır: yeniden örnekleme yöntemleri *(resampling methods)* (eğitim-testi bölme – *train/test split*, çapraz doğrulama – *cross-validation*, önyükleme – *Bootstrap*) ve olasılık ölçütleri *(probabilistic measures)* (Akaike Bilgi Kriteri, Bayes Bilgi Kriteri, Minimum Açıklama Uzunluğu, Yapısal Risk Minimizasyonu) (Kearns ve diğ., 1997; Brownlee, 2019). Yeniden örnekleme yöntemleri çoğu durumda uygun oldukları için en çok kullanılan yöntemlerdir(Kearns ve diğ., 1997). Yeniden örnekleme modeli seçim metrikleri, veri setini eğitim ve test seti (veya bazen eğitim, doğrulama, test seti) olarak bölmekten oluşur; eğitim seti modelin oluşturulmasına izin verirken, test seti modelin performansının değerlendirilmesine yardımcı olur (Chollet, 2018). Aslında, k-katlı çapraz doğrulama *(cross-validation)*, bekletme veya eğitim testi bölme *(hold-out)*, bir dışarıda bırakma çapraz doğrulama *(Leave One Out Cross Validation ya da LOOCV)*, katmanlı çapraz doğrulama *(stratified cross-validation)*, tekrarlanan çapraz doğrulama *(repeated cross-validation)*, İç içe çapraz doğrulama *(stratified cross-validation)* gibi farklı çapraz doğrulama yöntemleri vardır(Han ve diğ., 2011; Brownlee, 2018). K-katlı çapraz doğrulama, veri kümesini k eşit alt örneğe bölmekten oluşur ve $(k-1)$ veri kümesini eğitmek için kullanırken kalan bir katı değerlendirmek veya test etmek için kullanır; *Hold-out* veya eğitim-test bölünmesi, veri setinin iki parça eğitim ve test setine bölünmesidir (bazı kriterlere göre. Genel olarak eğitim için %80 ve test için %20'dir); bir dışarıda bırakılan çapraz doğrulama, k'nin veri kümesindeki örnek sayısına ayarlandığı zamandır; Katmanlı çapraz doğrulama, her katın belirli bir kategorik değerle (çoğunlukla Sınıflar) aynı oranda gözleme sahip olmasını sağlamaktan oluşur; Tekrarlanan çapraz doğrulama k-katlı çapraz doğrulamanın n kez tekrarlanmasıdır; ve son olarak, çift çapraz doğrulama olarak da adlandırılan iç içe çapraz doğrulama her çapraz doğrulama katının içinde bir çapraz doğrulama olduğunda gerçekleşir (Han ve diğ., 2011).

Performanslı bir model elde etmek için veri bilimciler genellikle hiperparametre ayarlaması veya performanslarına göre farklı algoritmalar arasında karşılaştırma gerçekleştirirler (Han ve diğ., 2011).

- ***Model performans değerlendirme ölçütleri***

Model oluşturulduktan sonra test setine göre model değerlendirilir. Bir modeli değerlendirmek için çeşitli metrikler vardır: Doğruluk değeri, F ölçütü, AUC – ROC, vb (Han ve diğ., 2011; Tavish Srivastava, 2019). Model değerlendirme metriklerinin birçoğunun elde edilmesi için karışıklık (*confusion*) matrisinden yararlanılmaktadır. Bir karışıklık matrisi, N'nin tahmin edilen sınıfın sayılarını temsil ettiği bir matris (N x N) olarak görülebilir. Bir karışıklık matrisi tablosundan doğruluk, pozitif Öngörü Değeri (veya Kesinlik), Negatif Öngörü Değeri, Duyarlılık veya Geri Çağırma, Özgüllük ve F ölçütü gibi birçok model performans değeri elde edilebilir. Doğruluk, tahmin sırasında elde edilen doğru tahminin oranını temsil eder. Pozitif ve Negatif Tahmin Değerleri sırasıyla doğru sınıflandırılan pozitif ve negatif vakaların oranını temsil eder. Duyarlılık, doğru şekilde sınıflandırılmış pozitif vakaları ve Özgüllük, doğru tanımlanmış negatif vakaların oranını temsil eder (Han ve diğ., 2011). F ölçütü (*F-score* ya da *F-measure* olarak da adlandırılan) kesinlik *(Precision)* ve hatırlamanın *(Recall)* birleştirilmesinden oluşan bir yaklaşımdır(Huang ve diğ., 2015). Bu metriklerin elde edildiği karışıklık matrisi ve formülleri Tablo 2.1: Karışıklık matrisi tablosu de gösterilmektedir:

Tablo 2.1: Karışıklık matrisi tablosu.

		Kestirim		
		S1 (+)	S2 (-)	
Gerçek	S1 (+)	A	B	Pozitif öngörme değeri: a/(a+b)
	S2 (-)	C	D	Negatif öngörme değeri: d/(c+d)
		duyarlılık: a/(a+c)	özgüllük: d/(b+d)	

$$Doğruluk = \frac{a + d}{a + b + c + d} \tag{2.1}$$

$$F \text{ ölçütü } = \frac{2 \times precision \times recall}{precision + recall} \tag{2.2}$$

$S1$ ve $S2$ veri setinin farklı sınıflarını ifade etmektedir. Örneğin bir hastalığın tahmin edilmesi söz konusu ise hastalık var yok gibi iki farklı kategoriyi temsil etmektedir.

Bu tez sırasında oluşturulan tüm modeller, yaygın olarak kullanılan doğruluk ölçütü ile değerlendirilmiştir.

- ***Varyans***

İstatistik ve olasılık teorisinde varyans, bir dizi sayının ortalama değerlerinden ne kadar uzağa yayıldığını tahmin eden bir ölçüdür. İstatistik ve bilimde, makine öğrenmesinde önemli bir role sahip olan varyans, model oluşturma sırasında farklı eğitim verilerinin kullanılması durumunda hedef fonksiyonun tahmininin değişeceğinin ölçüsünü temsil eder (Dietterich ve Kong, 1995). Bu, eğitim verilerinin verileri etkilediği anlamına gelir. Bu nedenle, düşük varyans, tahmin edilen veriler arasında küçük bir fark olduğu anlamına gelir ve yüksek varyans, tahmin edilen veriler ile gerçek olan arasında büyük bir fark olduğu anlamına gelir. f_S belirli bir hipotez ise ve $\bar{f}$ ortalama hipotez ise; varyansı hesaplamak için formül aşağıdaki gibidir:

$$Varyans(A, m, x) = E[(f_s - \bar{f}(x)^2] \tag{2.3}$$

A, kullanılan algoritma ve m, S eğitim veri setinin boyutudur (2.3).

Makine öğrenimi modelleri, veri bilimcileri için çok güçlü ve kullanışlı araçlardır. Hangi hastanın hasta olup olmadığı ya da bir krediden önce hangi müşteriye güvenilip güvenilmeyebileceği veya dolandırıcılık tespiti gibi farklı şeyleri ya da durumları tahmin etmelerine yardımcı olabilirler. Kullanışlılığına rağmen, makine öğrenimi modelleri bazen bir takas ilişkisine sahip olan önyargı (bias) ve varyansın kötü ayarlanması nedeniyle eksik ve fazla uyum sorunlarıyla karşı karşıya kalır. Sorun ilk bakışta ciddi görünmeyebilir ama bir hastanın akciğer analizi için hastaneye gittiği ve bunun için bilgisayarlı tomografi (BT) çekildiği ve tanının bir makine öğrenmesi sınıflandırması ile yönlendirildiği bir durumu düşünelim. Hastanın kanser olup olmadığını söylemesi gereken model hastanın normal olduğunu söyledi ve hasta evine gitti, aylarca acı çektikten sonra aynı hasta geri gelir ve doktorlar ileri evrede akciğer kanseri olduğunu öğrenirler. Yani teknik olarak, bu duruma (makine öğrenimi sınıflandırmasında Yanlış Pozitif denir) modelin yanlış tahmin edilmesi neden olur. Bu örnekteki problemin kaynağı birden fazla olabilir ve olası sebeplerden biri, oluşturulan modelin yüksek varyansa sahip olması ve bu nedenle doğrulama seti üzerinde iyi sonuçlar gösterse bile, test veriler üzerinde iyi tahminde bulunma konusunda zayıf doğruluğa sahip olmasına neden olabilir. Bir kanser hastasını sağlıklı olarak teşhis eden veya varken bir sahtekarlık veya güvenlik sorununu tespit edemeyen yetersiz bir model, dikkat edilmezse tehlikeli olabilir. Bir modelin varyansı tamamen ortadan kaldırılamaz, ancak kullanılabilir her yöntem kullanılarak azaltılabilir.

Yaygın olarak *Overfitting* (aşırı öğrenme, ezberleme) olarak adlandırılan yüksek varyans, makine öğrenimi topluluğunun karşılaştığı yinelenen ve en büyük sorunlardan biridir (Zhang ve diğ., 2018; Mehta ve diğ., 2019; Ying, 2019). Yüksek varyans, bir makine öğrenme modelinin test veriler üzerinde iyi tahminde bulunamamasıdır. Bir model eğitim setini o kadar iyi öğrendiğinde olur ki genelleme veya tahminde başarısız olur (Chollet, 2018). Modeli oluşturmak için kullanılan algoritma, veri kümesinin karmaşıklığını idare edemeyecek kadar basit olduğunda veya temel bir veri kümesini öğrenemeyecek kadar karmaşık olduğunda veya veri kümesinde tutarlı bir model oluşturmak için yeterli örneğe sahip olmadığında ortaya çıkabilir. Makine öğrenimi çağının başlangıcından bu yana, veri bilimciler *overfitting* sorununu önlemek için makine öğrenimi modellerinin varyansını mümkün olduğunca (elbette önyargıyı ihmal etmeden) azaltmanın bir yolunu araştırmaya devam ediyorlar. O zamandan beri, eğitim setine daha fazla örnek eklemekten oluşan veri büyütme, hiperparametre ayarlaması *(hyperparameter tuning)*, k-kat *(k-fold)* çapraz doğrulama, veri kümesindeki gereksiz özniteliklerin bırakılmasını içeren öznitelik azaltma, eğitim setine gürültü ekleme (çoğunlukla derin öğrenmede kullanılır), veya topluluk öğrenme yöntemi gibi çeşitli yöntemleri kullanımı oluşturulmuştur (Han ve diğ., 2011; Chollet, 2018).

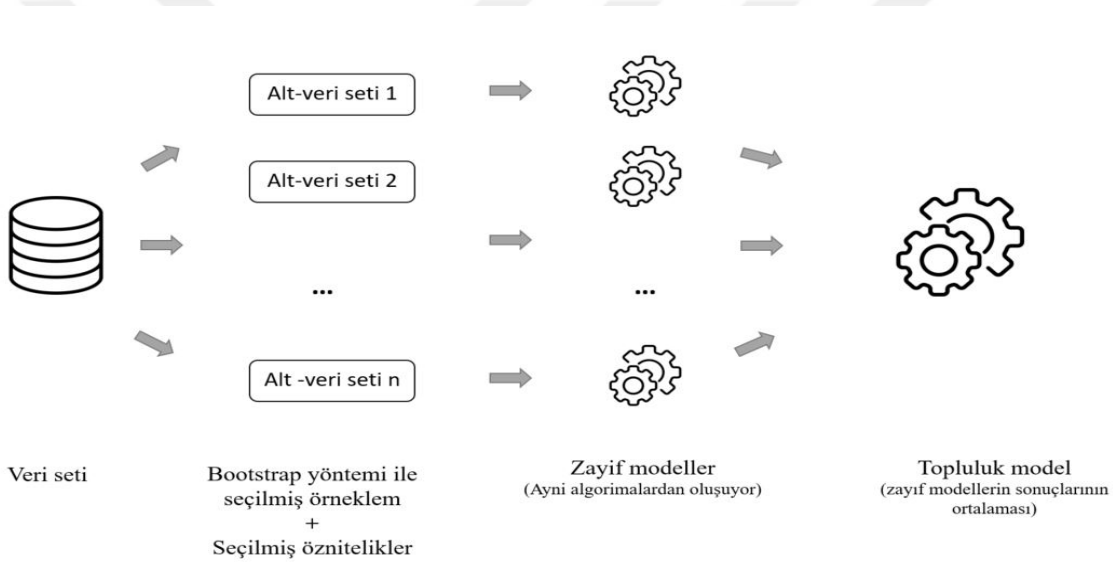
2.5 TOPLULUK ÖĞRENME

Topluluk yöntemleri olarak da adlandırılan topluluk öğrenme, birden çok ayrı modelden gelen tahminleri birleştirmekten oluşan denetimli makine öğrenimi yönteminin bir türüdür. Son zamanlarda topluluk yöntemleri de veri bilimi topluluğunda daha fazla popülerlik kazanmaya başlamıştır; yaygın ve iyi bilinen örnekleri Torbalama *(Bagging)*, Güçlendirme *(Boosting)* ve İstiflemedir *(Stacking)* (Opitz ve Maclin, 1999; Graczyk ve diğ., 2010; Han ve diğ., 2011; Eldardiry, Neville ve Rossi, 2020).

Bir topluluk öğrenme yöntemi oluşturmadan önce, zayıf modeller olarak da adlandırılan alt modellerin seçimi çok önemlidir. Çoğunlukla alt-modeller, farklı şekillerde eğitilmiş aynı algoritmadan oluşur. Elde edilen topluluk modelinin daha sonra "homojen" olduğu söylenir. Bununla birlikte, temel öğrenme modellerini oluşturmak için farklı türde algoritmalar kullanan İstifleme gibi bazı yöntemler de vardır; bu durumda oluşturulan topluluk "heterojen topluluklar modeli" olarak adlandırılır (Zhou, 2009).

Torbalama, genellikle homojen zayıf modellerden oluşan bir topluluk öğrenme yöntemidir (

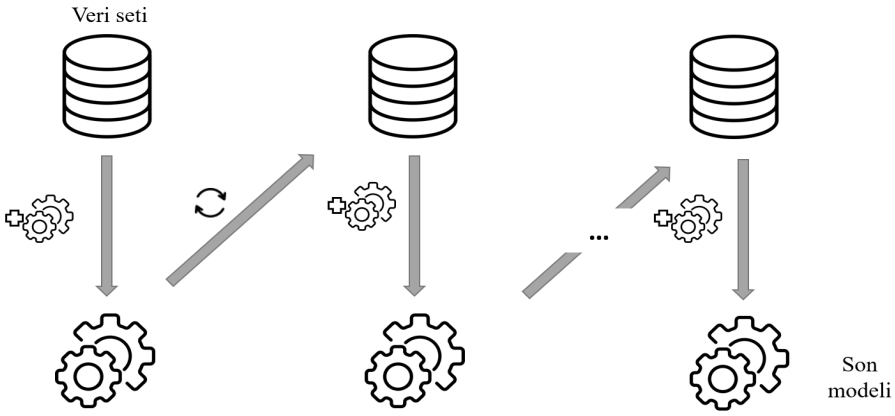
Şekil **2.5**). Bu modeller veri setini birbirinden bağımsız olarak öğrenir ve bir tür deterministik ortalama alma sürecini takiben paralel olarak birleştirilir(Yaman ve Subasi, 2019). Bu şekilde yapmanın sorunu, çok fazla veriye ihtiyaç duyulmasıdır. Bu sorunun üstesinden gelmek için çoğunlukla bootstrap örnekleme yöntemi kullanılır. Torbalamanın arkasındaki fikir çok basit; daha düşük varyansa sahip bir model elde etmek için birkaç bağımsız modele uymak ve tahminlerinin ortalamasını almaktır. Sınıflandırma görevleri için, her modelin tahmini bir oylama olarak görülebilir; nihai tahmin, oyların çoğunluğunu alan sınıftır. Regresyon görevlerinde ise son tahmin, zayıf modellerin tahmininin ortalamasıdır (Bühlmann, 2012).



Şekil 2.5: Torbalama modelinin mimarisi (Han ve diğ., 2011; Rocca, 2021).

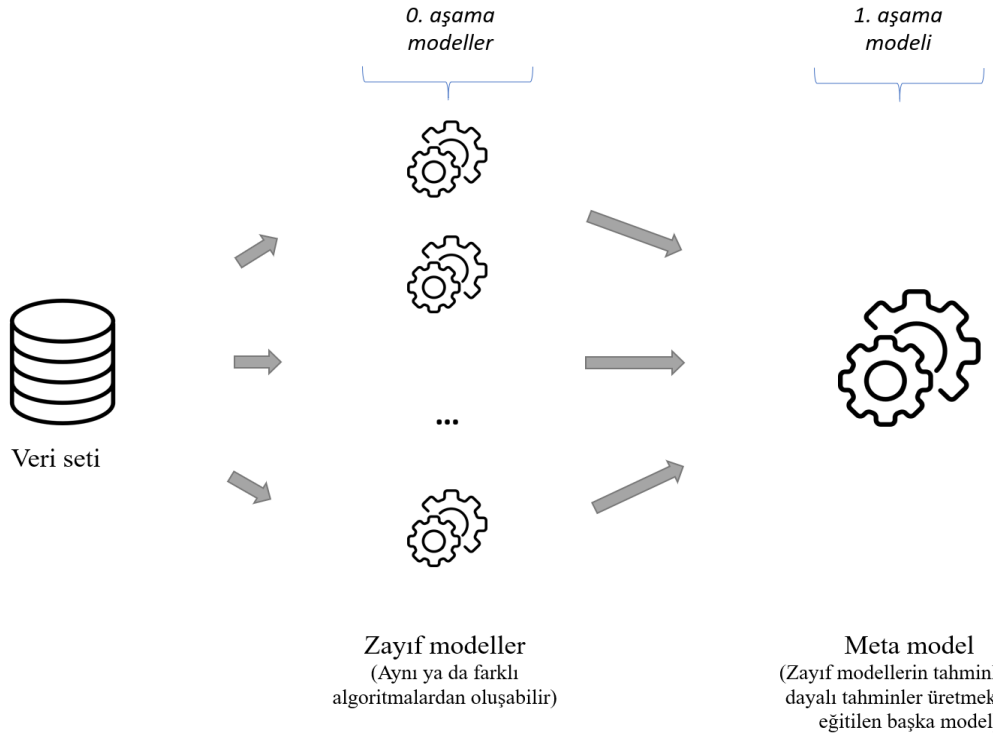
Topluluk modellerini güçlendirmek, genellikle homojen zayıf öğrenicileri dikkate alır, onları çok uyarlanabilir bir şekilde sırayla öğrenir (bir temel model öncekilere bağlıdır) ve sonra bunları deterministik bir strateji izleyerek birleştirir (

Şekil **2.6**). Güçlendirme, sıralı olarak çok sayıda zayıf modeli çok uyarlanabilir bir şekilde uydurmaktan oluşan bir tekniktir: dizideki her model, dizideki önceki modeller tarafından iyi olmayan, ele alınan veri kümesindeki gözlemlere daha fazla önem verecek şekilde yerleştirilir (Verma ve Mehta, 2017). Torbalama gibi güçlendirme, regresyon ve sınıflandırma görevleri için kullanılabilir.



Şekil 2.6: Güçlendirme modelinin mimarisi (Rocca, 2021).

İstifleme tabanlı topluluk öğrenme yöntemi bu tez çerçevesinde kullanılmıştır. Şekil 2.7'de gösterildiği gibi, aynı veri setinde alt modeller (veya *level 0* modeller) adı verilen birden fazla makine öğrenimi modelinden gelen tahminlerin birleştirilmesinden oluşur(Graczyk ve diğ., 2010; Mohammed ve diğ., 2018). Daha sonra son bir model (veya *level 1* modeli) bu tahminlerden en iyi sonuçları elde etmek için bunları en iyi şekilde nasıl birleştireceğini öğrenecektir.



Şekil 2.7: İstifleme modelinin mimarisi (Rocca, 2021).

Bu tez kapsamında, iki farklı istifleme modeli oluşturulmuştur. Her istifleme modelinde sıfırıncı aşama (yani zayıf modeller) ve birinci aşama modeli (yani Meta model) bulunur. Oluşturulan iki istifleme modelinin sıfırıncı aşaması Doğrusal Destek Vektör Makinesi (*Linear SVM*), Radyal Destek Vektör Makinesi (*Radial SVM*), Doğrusal Ayırıcı Analizi (*Linear Discriminant Analysis*), k-En Yakın Komşu (*k-Nearest Neighbors*), CART, Saf Bayes (Naive Bayes), Çok Katmanlı Algılayıcı (*Multi-Layer Perceptron*), C5.0, Kısmi En Küçük Kareler (*Partial Least Squares*), Öznitelik Çıkarımlı Sinir Ağları (*Neural Networks with Feature Extraction*), Rasgele Orman (*Random Forest*), ve Genelleştirilmiş doğrusal model (*Generalized linear model*) algoritmalarından oluşturulmuştur.

Bu tez çalışmasında iki topluluk modeli oluşturuldu. Birinci modelde istifleme aşamasında GLM algoritması kullanılırken ikinci modelde doğrusal SVM algoritması kullanılmıştır. Bu algoritmalar zayıf modelleri güçlendirmesi amacıyla denenmiştir.

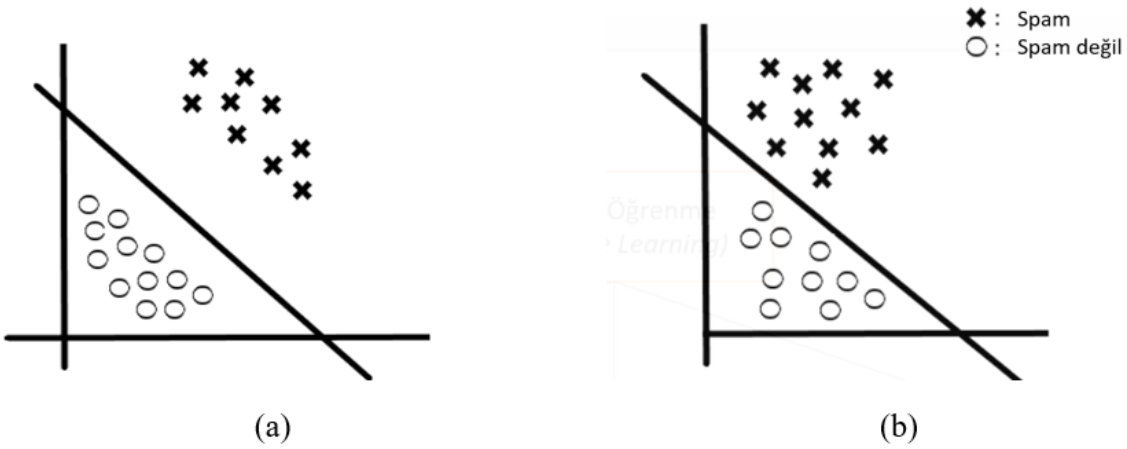
2.6 ALGORİTMALAR

Makine öğrenmesinde kullanılan birçok algoritma bulunmaktadır. Bu bölümde bu tez çalışmasında kullanılan Doğrusal Destek Vektör Makinesi (*Linear SVM*), Radyal Destek Vektör Makinesi (*Radial SVM*), Doğrusal Ayırıcı Analizi (*Linear Discriminant Analysis*), k-En Yakın Komşu, CART, Saf Bayes, Çok Katmanlı Algılayıcı, C5.0, Kısmi En Küçük Kareler, Öznitelik Çıkarımlı Sinir Ağları, Rasgele orman, ve Genelleştirilmiş Doğrusal Model algoritmalara değinilmiştir.

2.6.1 Destek Vektör Makineleri (SVM)

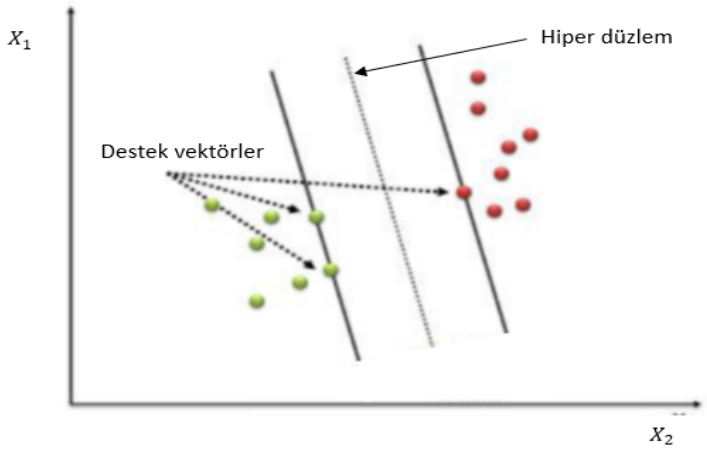
Veri madenciliğinde, Destek Vektör Makineleri (SVM'ler); sınıflandırma, regresyon ve aykırı değer tespit görevleri için kullanılabilen bir dizi denetimli öğrenme yöntemidir. Çekirdek adı verilen farklı işlevlerin (doğrusal, polinom, radyal temel ve sigmoid) kullanımıyla bu yöntem, sınıflandırma görevini doğrusal ve doğrusal olmayan olarak ayırır(Azimi-Pour ve diğ., 2020). Veri kümesini ayırmak için SVM, o işlevin parametrelerini bulmaya çalışır. Çoğunlukla makine öğreniminde kullanılır; veri madenciliğinde de tercih edilmektedir. SVM'nin ana fikri, öznitelikleri farklı alanlara (sınıf = "*Class*") en iyi şekilde ayıran bir hiper düzlem (*hyperplane*) bulmaktır.

Örneğin bir yöneticinin e-postasının “spam” ya da “spam değil” olarak sınıflandırılmasını istediğini düşünelim. Bir çözüm, istenmeyen bir e-posta geldiğinde istenmeyen posta olarak sınıflandırılabilmesi için iki durumu açıkça ayıracak bir fonksiyon (bir hiper düzlem) tasarlamak olabilir. Aşağıdaki şekiller, hiper düzlemin çizildiği iki durumu göstermektedir.



Şekil 2.8: Destek vektör makinesinin illüstrasyonu (Yadav, 2018).

Şekil 2.8 (a) tarafından sunulan iyi bir çözüm olacaktır çünkü Şekil 2.8 (a)'daki e-postalar açıkça sınıflandırılmıştır ve Şekil 2.8 (b)'ye kıyasla sınıflandırma konusunda daha fazla güven sağlıyor. Özetlemek gerekirse, SVM temelde farklı sınıfları açıkça sınıflandıracak bir Optimal hiper düzlem oluşturma fikrinden oluşur (bu durumda bunlar ikili sınıflardır) (Şekil 2.9). Bir hiper düzlem, farklı öznitelikleri ayırt etmek için kullanılan bir fonksiyondur. 2-D'de bu fonksiyon bir çizgidir, oysa 3-D'de düzlem olarak adlandırılır; benzer şekilde daha yüksek boyuttaki noktayı sınıflandıran fonksiyona hiper düzlem denir.



Şekil 2.9: Destek vektör makinesinin açıklaması(Yadav, 2018).

Vapnik tarafından geliştirilen(Boser ve diğ., 1992; Vapnik, 1998), temel destek vektörü (SV) regresyon kavramı önce doğrusal bir modelle tartışıldı ve ardından çekirdek fonksiyonları aracılığıyla doğrusal olmayan bir modele genişletildi. Bir eğitim verisi verildiğinde $\{(x_1, y_1), \ldots, (x_n, y_n), X \in R^n$ ve $Y \in R\}$; SV regresyon denklemi aşağıdaki gibi verilebilir:

$$f(x) = (w, x) + b, w \in X, b \in R \tag{2.4}$$

(w, x) : X'deki skaler çarpma ifade eder.

b : önyargı (*bias*)

ξ SVM'in kayıp fonksiyonunu için düşünülen hassas kayıp fonksiyonu şu şekilde ifade edilmektedir:

$$|\xi|_\varepsilon = |y - f(x)|_\varepsilon = \begin{cases} |y - f(x)| < \varepsilon \; ise \; 0 \\ yoksa \;\; |y - f(x)| - \varepsilon \end{cases} \tag{2.5}$$

SVM regresyonunun temel amacı, minimum kayıp fonksiyonu değeriyle f(x) fonksiyonunu bulmak ve ayrıca mümkün olduğunca düz olmaktır. Minimum bir normun elde edilmesini sağlamak için, daha küçük olan ω değerini bulmak gerekir yani $\|\omega\| = (\omega, \omega)$.Bu nedenle model aşağıdaki dışbükey optimizasyon problemi olarak ifade edilebilir.

$$\min\left(\frac{1}{2}\|\omega\|^2 + C\left(\sum_{i}^{l}\xi_i^* + \sum_{i=1}^{l}\xi_i\right)\right) \tag{2.6}$$

$$\text{Koşullar: } \begin{cases} y_i - f(\omega, x) - b \leq \varepsilon + \xi_i \\ f(\omega, x) + b - y_i \leq \varepsilon + \xi_i \\ \xi_i . \xi_i^* \geq 0, i = 1, \ldots, n \end{cases} \tag{2.7}$$

Bu optimizasyon problemi ikinci dereceden bir programlama problemine dönüştürülebilir ve çözümü şu şekilde verilir:

$$f(x) = \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)k(x_i, x) + b \tag{2.8}$$

$k\ (x_i, x)$: Girdi verilerini n boyutlu uzaya dönüştürmek için çekirdek *(kernel)* fonksiyonudur.

SVM yöntemi için birçok çekirdek fonksiyonu mevcuttur. Bunların arasında doğrusal, polinom, radyal temel fonksiyon (RBF) ve sigmoid vardır, bu tez kapsamında doğrusal be Radyal çekirdekli SVM'ler kullanılmıştır, aşağıdaki gibi sağlanan çekirdek fonksiyonuna sahip fonksiyonlardır:

- Doğrusal: $k(x_i, x) = x_i . x$
- Radyal: $k(x_i, x) = exp(-\gamma \|x_i - x\|^2)$

xi ve x: iki öznitelik vektörü arasındaki Öklid mesafesinin karesi olarak tanınabilir

γ : sadece radyal SVM için kullanılır, model fazla sığdığında(*overfit*) değeri artar ve model yetersiz kaldığında(*underfit*) azalır. Gama parametresi, tek bir eğitim örneğinin etkisinin ne kadar uzağa ulaşacağını tanımlar; düşük değerler "uzak" ve yüksek değerler "yakın" anlamına gelir. Gama parametresi, model tarafından destek vektörleri olarak seçilen örneklerin etki yarıçapının tersi olarak görülebilir.

2.6.2 K-En Yakın Komşu (K-Nearest Neighbour)

K-En Yakın Komşu anlamına gelen KNN, makine öğrenmesinde kullanılan algoritmalardan biridir. İlk olarak 1951'de Evelyn Fix ve Joseph Hodges tarafından geliştirildi; sınıflandırma ve

regresyon işlemleri için kullanılır. Bu algoritmanın arkasındaki ilke, veri kümesindeki en yakın K eğitim örneğinden oluşur ve çıktı, görevin türüne (sınıflandırma veya regresyon) bağlıdır.

S demet (*tuple*) ve n öznitelikleri olan bir veri kümesi için; her demet, n boyutlu uzayda bir noktayı temsil eder. Bu şekilde, tüm eğitim demetleri n-boyutlu desen uzayında saklanır. Bilinmeyen bir demet (bir test seti) verildiğinde, bir k-en yakın-komşu sınıflandırıcı, verilen bilinmeyen demete en yakın olan k eğitim demetleri için desen uzayını arar. Bu k eğitim demeti, bilinmeyen demetin K "en yakın komşusunu" temsil eder. Bilinmeyen grubun sonucu, en yakın K komşunun katıldığı bir oylama ilkesi ile belirlenir. Bu nedenle, K seçilirken tek bir değer kullanılması önerilir.

Yeni bir demeti sınıflandırmak için KNN, aşağıda listelenen adımları takip edilebilir:

- **Adım-1:** Komşu K sayısını seçmek;
- **Adım-2:** Eğitim setinde yer alan her bir demet ile yeni demet arasındaki mesafeyi hesaplamak. Mesafe, Manhattan mesafesi veya Öklid mesafesi gibi mesafe ölçüm yöntemleri kullanılarak hesaplanabilir;
- **Adım-3:** Yeni demet ve eğitim demetleri arasındaki en küçük mesafeleri temsil eden K en yakın komşuyu almak;
- **Adım-4:** Bu K komşu arasında, her kategorideki (sınıftaki) veri noktalarının sayısını saymak;
- **Adım-5:** Yeni demeti en yüksek komşu sayısına sahip o kategoriye (sınıfa) atmak.

$$ED(x, y) = \sqrt{\sum_{i=1}^{m} (x_i - y_i)^2} \tag{2.9}$$

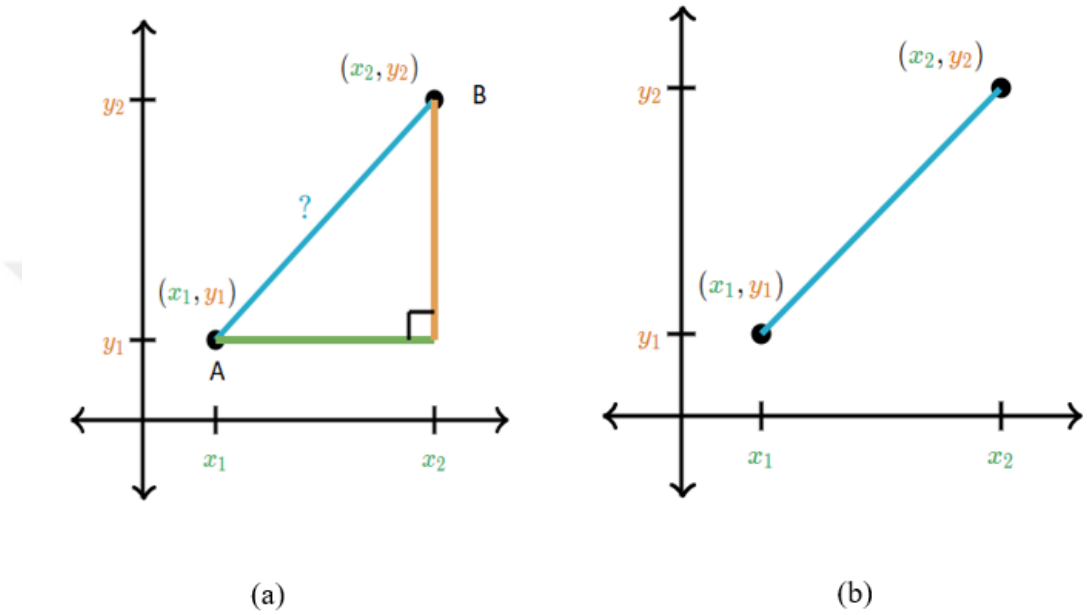
ED (2.9), Öklid Uzaklığı ifade etmektedir(Özkan ve Erol, 2015) ;

x, eğitim veri setinin bir örneklemi ifade ederken, y bir test örneklemi ifade ediyor.

m ise D veri setinin öznitelik sayısı ifade etmektedir.

$$MD(x,y) = \sum_{i=1}^{m} (|x_i - y_i|) \tag{2.10}$$

Manhatan uzaklığı (MD) (2.10), yükarıdaki formulu kullanarak hesaplanılır.



Şekil 2.10: Öklid ve Manhatan uzaklığı illüstrasyon (Sharma, 2019).

Regresyon görevlerinde, yeni çıktının değeri, K en yakın komşunun değerinin ortalaması ile belirlenir.

2.6.3 Karar Ağaçları

Karar ağacı sınıflandırıcıları, ağaç benzeri bir hiyerarşik yapı oluşturarak sınıflandırmayı sağlayan yöntemlerdir. Karar ağacı ilkesine dayanan birçok algoritma vardır. Karar ağacının ardındaki ilke *"divide and conquer"*tir (Quinlan, 1996). Karar ağaçlarının sağladığı en önemli avantajlardan bazıları, gerçek hayat problemlerine kolayca uyum sağlayan algoritmalarını anlama kolaylığı ve hem sayısal hem de kategorik verilerle çalışma fırsatıdır(Bujlow ve diğ., 2012) . Karar ağacı algoritmalarının anahtar bileşeni, yaprak düğümler*(leaf nodes)*, dallar*(branches)*, karar düğümleri *(decision nodes)*, ve en üst düğüm olan düğümü olan kök

düğüm*(root node)* (Larose ve Larose, 2014). Şekil 2.11, bir karar ağacının mimarisini göstermektedir.

Şekil 2.11: Karar ağacı illüstrasyonu (Han ve diğ., 2011).

2.6.3.1 C4.5 ve C5.0 algoritmaları

C5.0 algoritması, C4.5'in geliştirilmiş bir sürümünü temsil eden ve karar ağaçlarına dayalı yeni nesil bir makine öğrenimi algoritmasıdır(Bujlow ve diğ., 2012). C4.5 ile aynı prensibi kullanan C5.0 algoritma bir eğitim setinden oluşturulur ve daha sonra tahmin (veya sınıflandırma) için kullanılır (Han ve diğ., 2011). Eğitim seti, nihai sonuca (tahmin) yol açan kurallar ve dallar oluşturarak ağaç oluşturmaya yardımcı olur. C4.5'in geliştirilmiş bir versiyonu olan C5.0, değişken yanlış sınıflandırma maliyetleri, yeni nitelikler (tarihler, saatler, zaman damgaları, sıralı ayrık nitelikler), artırma (tahminleri iyileştirmek için birkaç karar ağacı oluşturulur ve birleştirilir), örnekleme ve çapraz doğrulama desteği ve değerleri belirli durumlar için eksik veya uygulanamaz olarak işaretleme olasılığı gibi birçok avantaj sunar (Bujlow ve diğ., 2012).

Karar ağaçlarındaki hiyerarşik bölme, "Splitting criterion", "Splitting attribute", "Split point", "splitting subset" olarak da adlandırılan bölme kriterine göre yapılır (Han ve diğ., 2011). Bölme kriteri bölme kuralları *(splitting rules)* veya nitelik seçim ölçütleri *(attribute selection measures)* tarafından belirlenir ve bunlar arasında kazanç oranı *(gain ratio)*, bilgi kazancı *(information gain)* ve Gini indeksi *(Gini index)* olabilir(Maimon ve Rokach, 2010; Han ve diğ., 2011). C5.0, atası C4.5 gibi, kazanç oranı ölçümüne dayalı olarak oluşturulmuştur.

Kazanç oranı hesaplamadan önce bilgi kazancı hesaplamak gerekmektedir. Bilgi kazancı, 1948'de Shannon tarafından tanıtılan "*Shannon entropy*" (entropi, öznitelik seçimlerde kullanılan yöntemler arasında yararlanan bir kavramıdır (Shannon, 1948) kullanılarak hesaplanır ve formülü (2.11) daki gibidir:

$$Entropi(D) = -\sum_{i=1}^{m} p_i \log_2 p_i \tag{2.11}$$

$$\text{Entropi}_{\text{A}}(\text{D}) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} Entropi(D_j) \tag{2.12}$$

Formül (2.11) ve (2.12) dan yola çıkarak, (2.13) formülü ile bilgi kazancı elde edilir.

$$Bilgi\ Kazancı(A) = Entropi(D) - EntropiA(D) \tag{2.13}$$

P_i, verilerimizdeki bir i öğesinin/sınıfının en sık rastlanan olasılığı ifade edilmektedir.

m, sınıf sayısı ifade etmektedir.

A, bir veri setinin sütun ifade ederken D_j , A'nın sütunun özel bir değeridir.

A'nın bilgi kazancı, A niteliğinden yapılacak dallanma ne kadar bilgi edileceğini göstermektedir. En yüksek bilgi kazancını sağlayan, yani başka bir deyiş ile kazancı maksimize edebilecek X niteliği seçilir (Özkan ve Erol, 2015).

Bilgi kazancı hesapladıktan sonra (formül referansı) formülü kullanılarak bilgi bölme işlemi yapılır ve ardından (ikinci formül) ile kazanç oranı hesaplanır.

$$Bölme\ bilgisi_A(D) = -\sum_{i=1}^{k} \frac{|D_i|}{|D|} \log_2 \left(\frac{|D_i|}{|D|} \right) \tag{2.14}$$

$$Kazanç\ oranı(A) = \frac{Bilgi\ kazancı(A)}{Bölme\ bilgisi_A(D)} \tag{2.15}$$

Kazanç oranı her öznitelik için ayrı ayrı hesaplanır ve ardından dal oluşturma, en yüksek kazanç oranına sahip öznitelikten başlar.

2.6.3.2 CART (Classification And Regression Trees) Algoritması

C4.5 ve C5.0 gibi CART (Sınıflandırma ve Regresyon Ağaçları anlamına gelir) algoritması, Sınıflandırma ve Regresyon görevleri için kullanılan Ağaç tabanlı bir makine öğrenme algoritmasıdır(Han ve diğ., 2011). İlk olarak Breiman tarafından önerilen, karar ağacını sadece iki gruba ayırabilen ikili ağaçlar üretir (Breiman ve diğ., 1984). Bilgi kazancı ve Kazanç oranını kullanan ID3, C4.5 ve C5.0'ın aksine, C5.0 CART algoritması, bir öznitelik seçmek için bir kirlilik ölçüsü olarak Gini indeksini kullanır. R programında rpart olarak kullanılan CART algoritma, aşağıdaki süreci izleyerek çalışır:

- **Adım 1:** Gini İndeksi (2.16), her bir sınıfın olasılıklarının karelerinin hesaplanan toplamının çıkarılmasıyla hesaplanacaktır.

$$Gini(D) = 1 - \sum_{i=1}^{m} p_i^2 \tag{2.16}$$

Örneğin ilk bölme için gini indeksi şu şekilde hesaplanır:

$$Gini_A(D) = \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2) \tag{2.17}$$

i sınıf sayısını temsil eder; $D1$ ve $D2$, D'nin (veya D'yi alabilen farklı bir değerin) alt kümeleridir; ve Pi, i olasılığını temsil eder

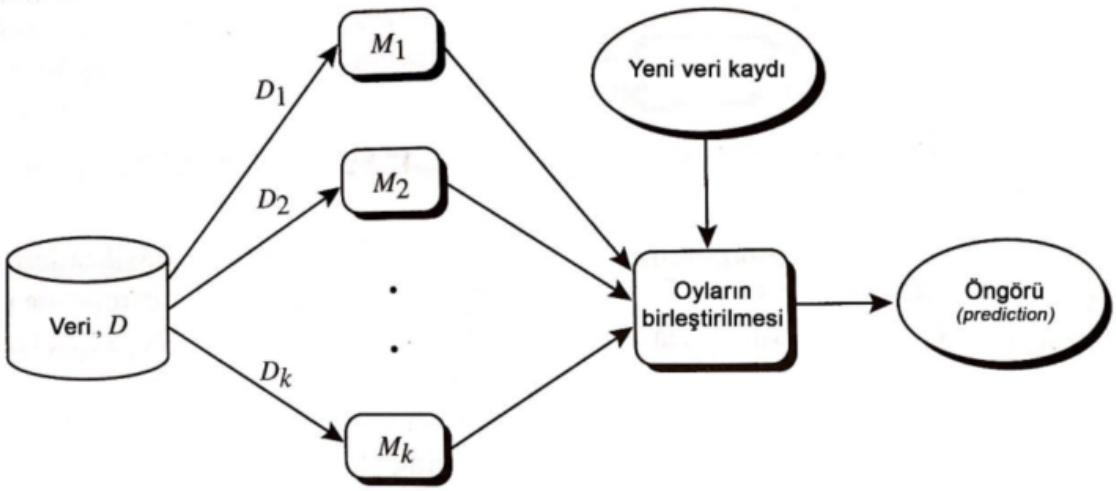
- **Adım 2:** Bundan sonra, en küçük Gini indeksine sahip bölme seçilecektir.

- **Adım 3:** Ağacın yapımına kadar aynı işlem tekrarlanacaktır.

2.6.4 Rastgele Orman

Rastgele orman sınıflandırıcısı, her sınıflandırıcının girdi vektöründen bağımsız olarak örneklenen rastgele bir vektör kullanılarak oluşturulduğu ağaç sınıflandırıcılarının bir kombinasyonundan oluşan bir topluluk öğrenme türüdür(Han ve diğ., 2011). Torbalama, Yükseltme, İstifleme gibi, birincil amacı doğruluğu artırmaktır. Rastgele Orman

algoritmasındaki her ağaç, bir girdi vektörünü sınıflandırmak için en popüler sınıf için bir birim oy verir (Breiman, 1999). Rastgele Orman algoritmasının bir örneği aşağıdaki şekilde gösterilmiştir. Rastgele Orman, rastgele nitelik seçimi ile Torbalama topluluğu yöntemi ilkesi kullanılarak oluşturulabilir; ancak Rastgele Orman algoritması, her bölme için çok daha az özniteliği dikkate alması nedeniyle Torbalama veya Hızlandırmadan daha hızlı çalışabilir (Han ve diğ., 2011). Bahsedilmesi gereken önemli bir şey, Rastgele Ormanların değişken öneme sahip dahili tahminler vermesidir.



Şekil 2.12: Rastgele orman sınırlandırcının mimarisi (Han, Pei ve Kamber, 2011).

Şekil 2.12'de, Rastgele Orman algoritmasının çalışma prensibini göstermektedir (Han ve diğ., 2011). Rastgele Orman birçok karar ağacından (alt model dense) oluşur ve sonucu bu ağaçların oylama prensibine dayanır. Karar ağaçları büyütmek için CART metodolojisi kullanılır(Breiman, 1999).

Örneğin verilen bir eğitim seti D, d demet*(tuples)*; rastgele orman topluluk öğrenme için k karar ağacı oluşturmak için genel prosedür aşağıdaki gibidir:

Her yineleme (*iteration*) i *için*$(i = 1, 2, \ldots, k)$, D'den Bootstrap örnekleme *(bootstrap resampling)* yöntemi kullanılarak d demetlerinden oluşan bir D_i eğitim seti örneklenir. F, her bir düğümdeki bölünmeyi belirlemek için kullanılacak öznitelik sayısı ise; F, mevcut

özniteliklerin sayısından çok daha küçük olacaktır. Bir karar ağacı sınıflandırıcısı M_i oluşturmak için, düğümdeki bölünmeye aday olarak her düğümde F nitelikleri rastgele seçilir.

2.6.5 Saf Bayes (*Naive Bayes*)

Saf Bayes (*Naive Bayes*) sınıflandırma yöntemi, Bayes teoremini sınıf değişkeninin değeri verilen her öznitelik çifti arasında koşullu bağımsızlık "Saf" (*Naive*) varsayımıyla uygulamaya dayanan denetimli bir öğrenme algoritmasıdır (Bayes, 1763; Zhang ve Su, 2004). Bayes'in teoreminin ifade ettiği ilişki şu şekildedir: verilen sınıf değişkeni Y ve bağımlı öznitelik X = {x_1, …. , x_n} (X_1'ler birbirinden bağımsız):

$$P(Y|X\ ) = \frac{P(Y)P(X|Y)}{P(X)} \tag{2.18}$$

$P(Y|X)$: X'yi bilen Y'nın olasılığıdır. $\boldsymbol{C}$ bir sınıfına ait olma iddiası, hipotez **Y** ile aynı şekilde tanımlanır. Bir **X** veri setinin bilinmeyen **C** sınıfı, bu ilişkiye göre hesaplanır.

$P(X|Y)$: **Y**'yi bilen **X** öznitelik değerlerinin olasılığıdır. **Y**'nin bilinen niteliklerine göre sınıflardan birine ait olma olasılığının hesaplanmasıdır. Koşullu olasılıklar, her öznitelik değeri için ayrı ayrı hesaplanmalıdır.

Örnek: İki sınıftan (C1="kanser" ve C2="normal") ve 3 öznitelikten (X_1, X_2 ve X_3) oluşan bir veri kümesi varsayılırsa, her bir özniteliğe göre X'in "kanser" ve "normal" sınıflarına ait olma olasılıklarının hesaplanması gerekmektedir. Öyle bir durumda aşağıdaki olasılık hesaplamaları yapılması gerekmemektedir ((2.19), (2.20)):

$$P(X_1|sınıf = kanser), P(X_2|sınıf = kanser), P(X_3|sınıf = kanser) \tag{2.19}$$

$$P(X_1|sınıf = normal), P(X_2|sınıf = normal), P(X_3|sınıf = normal) \tag{2.20}$$

$P(X \mid sınıf = kanser)$ ve $P(X \mid sınıf = normal)$ her X değerinin "Kanser" ve "Normal" sınıflara ait olma olasılıklarıdır. Her bir X özniteliği için yukarıdaki hesaplanan değerlerin çarpımına eşittir.

P(Y): Y'ye ait önsel olasılıktır. X'in herhangi bir sınıfa ait olma olasılığıdır ((2.21), (2.22)) o yüzden her şeyden önce olasılığın ne olduğu bilinmektedir.

$$P(sınıf = kanser) \tag{2.21}$$

$$P(sınıf = normal) \tag{2.22}$$

$P(X)$: X'e ait önsel olasılıktır. Tüm sınıflar için sabittir.

Sınıfların olasılıkları hesaplamak için (2.23) formülünden yararlanılır.

$$P(Yi|X) = \frac{P(X|Yi)P(Yi)}{P(X)} \tag{2.23}$$

X'e ait nitelik değerlerinin birbirinden bağımsız olduğu (*class-conditional independent*) kabul edilerek (2.23) formülü yerine aşağıdaki formülü (2.24) kullanılabilir.

$$P(X|Y_i) = \prod_{k=1}^{n} P(x_k|Y_i) = P(x_i|Y_i)P(x_2|Y_i) \dots P(x_n|Y_i) \tag{2.24}$$

X'e ait sınıf değerini hesaplamak için yapılan işlemlerin amacı $P(X|Y_i)$ değerinin maksimize edilmesidir. Bu durum en büyük sonrasal sınıflandırma yöntemi (*maximum posteriori hypothesis*) adı verilir şöyle ki X, $P(X|Y_i)$ $P(Y_i)$ ifadesinin maksimum olduğu sınıfa aittir. Aşağıdaki formülü kullanılarak elde edilir.

$$argmax\{P(X|Y_i)P(Y_i)\} \tag{2.25}$$

Başka bir deyiş ile eğer X'in değerine göre Yi sınıfına ait olma olasılığı Yj sınıfına ait olma olasılığında büyükse X, Yi sınıfına aittir (2.26)

$$P(X|Y_i)P(Y_i) > P(X|Y_j)P(Y_j) \Rightarrow P(X) \epsilon Y_i \qquad 1 \leq j \leq m, j \neq i \tag{2.26}$$

Yukarıda belirtilen hesaplamalar, X değerleri kategorik olduğunda kullanılır. Nümerik veriler söz konusu olduğunda, olasılıkları hesaplamak için aşağıdaki formülü kullanılır. Sayısal değerlerin Gauss dağılımını gösterdiğini varsayarsak; μ sayısal değer ile gösterilen sınıfın

ortalama değeri ise, σ sayısal değer ile gösterilen sınıfa ait standart sapma ise, işlem formül XX ile gerçekleştirilir:

$$P(X_k|C_i) = f\left(x_i, \mu_{Y_i}, \sigma_{Y_i}\right) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{(x-\mu)^2}{2\sigma^2}} \tag{2.27}$$

2.6.6 Lineer Diskriminant Analizi

Lineer Diskriminant Analizi (*Linear Discriminant Analysis* = LDA), boyutluluk azaltma ve veri sınıflandırma için yaygın olarak kullanılan bir makine öğrenme algoritmasıdır (El-Bendary ve diğ., 2015). Sonuç sınıflarının frekanslarının dengesiz olduğu ve performanslarının rastgele seçilmiş test verileri üzerinde incelendiği durumları kolaylıkla ele alır. Tüm sınıfların lineer olarak ayrılabilir olduğunu varsayarak, LDA algoritması, sınıflandırmanın bazı metriklere (Öklid mesafesi gibi) dayalı dönüştürülmüş bir uzayda gerçekleştirilebilmesi için her sınıfı en iyi şekilde ayıran bir lineer dönüşüm bulmaya çalışır (El-Bendary ve diğ., 2015; Vaibhaw ve diğ., 2020).

Verilen y_i bir veri seti, $x\_i, i \in \{1, .., n\}$, ve $y_i \in \{1, 2, \ldots., k\}$ sınıf etiketidir, k sınıf sayısıdır ve x_i bir öznitelik veya tahmin vektörüdür, amaç sınıfların mümkün olduğunca ayrıldığı tahmin uzayında en iyi yönü bulun.

Matematiksel olarak, LDA uygulaması, dağılım matrisi analizi yoluyla gerçekleştirilir. Yani tüm sınıfların tüm örnekleri için iki ölçü tanımlanmıştır (Li ve diğ., 2006):

- Her sınıfın ortalaması ve örneği arasındaki mesafeyi hesaplayın (Sınıf içi – *within-class*- varyans):

$$S_w = \sum_{j=1}^{K} \sum_{i=1}^{N_j} \left(x_i^j - \mu_j\right)\left(x_i^j - \mu_j\right)^T \tag{2.28}$$

x_i^j Sınıf j'nın i. Örneğidir; μ_j, j sınıfın ortalamasıdır, K sınıf sayısıdır ve K , j sınıfındaki örnek sayısıdır;

- Farklı sınıfların ortalaması arasındaki mesafe olan sınıflar arasındaki ayrılabilirliği hesaplayın (sınıflar arası- *between-class*- varyans olarak adlandırılır):

$$S_b = \sum_{j=1}^{k} (\mu_j - \mu)(\mu_j - \mu)^T \tag{2.29}$$

Formül (2.29)'de μ, tüm sınıfların ortalamasını temsil eder. Bu yöntem, bir veri setinde sınıflar arası mesafenin sınıf içi mesafeye oranını maksimize ederek maksimum ayırt edebilirliği garanti eder.

Maksimizasyon fonksiyonu şu şekildedir (2.30):

$$\max\left(\frac{\delta|S_b|}{\delta|S_w|}\right) \tag{2.30}$$

2.6.7 Kısmi En Küçük Kareler

İsveçli istatistikçi Herman OA Wold tarafından tanıtılan Kısmi en küçük kareler(Partial Least Squares= PLS), boyut küçültme yöntemine dayanan bir makine öğrenme algoritmasıdır ve yeni tahmin edicileri (temel bileşen – principal component – olarak da adlandırılır) denetimli bir şekilde seçmesi farkıyla temel bileşenler regresyonu ile aynı yöntemi kullanır (Wold, 1975; Wold ve diğ., 1984). Sınıflandırma ve regresyon görevleri ile boyut küçültme teknikleri için de kullanılan *PLS* algoritma, araştırılan sistem veya sürecin bir dizi temel Gizli Vektörler (bileşenler veya puan vektörleri olarak da adlandırılır) tarafından yönlendirildiği temel varsayımına dayanır. X ve Y'nin Gizli Vektörleri arasındaki ikili kovaryansı maksimize edecek şekilde hem X hem de Y'yi bir alt uzaya yansıtarak Gizli Vektörleri çıkarır. Gizli Vektörlerin karşılıklı ortogonalliğini sağlamak için, bu prosedür, puan vektörlerine dayalı birinci derece yaklaşımlarıyla açıklanan bilgileri X ve Y'den çıkaran deflasyon şeması kullanılarak iteratif olarak gerçekleştirilir. Verilen iki veri kümesi X ve Y; çok değişkenli *PLS*'nin altında yatan genel modellin formülü aşağıdaki gibidir (2.31)(2.32):

$$X = TP^T + E \tag{2.31}$$

$$Y = UQ^T + F \tag{2.32}$$

Burada X bir $n \times m$ tahmin edici matrisidir, Y bir tahmin edilen (yanıt) $n \times p$ matrisidir; U ve T, sırasıyla Y'nin projeksiyonları (Y puanları) ve X'in projeksiyonları (X puanı, bileşen veya

faktör matrisi) olan $n \times l$ matrisleridir; P ve Q sırasıyla $m \times l$ ve $p \times l$ ortogonal yükleme matrisleridir ve E ve F matrisleri, özdeş dağılımlı rastgele normal ve bağımsız değişkenler olduğu varsayılan hata terimleridir. X ve Y'nin ayrıştırmaları, T ve U matrisleri arasındaki kovaryansı maksimize edecek şekilde yapılır.

2.6.8 Genelleştirilmiş Doğrusal Model

İlk olarak Nelder ve Wedderburn (1972) tarafından sunulan Genelleştirilmiş Doğrusal Model (*Generalized Linear Model* = GLM), sınıflandırma ve regresyon görevleri için kullanılan Doğrusal Regresyonun bir uzantısıdır. Doğrusal Regresyondan farklı olarak GLM, X özniteliklerine dayanarak Y sonucunun bir kategori (kanser ve sağlıklı), bir sayı (çocuk sayısı), bir olayın meydana gelme zamanı (zaman) olabileceği gerçeği gibi birçok varsayımı içerir. Bir makinenin arızalanması) veya çok yüksek değerlere sahip çok çarpık bir sonuç (hane geliri)(Molnar, t.y.).

Genelleştirilmiş doğrusal model, verilen X bağımlı değişkenleri, her bir Y sonucunun, diğerlerinin yanı sıra normal, binom, Poisson ve gama dağılımlarını içeren geniş bir olasılık dağılımları sınıfı olan üstel bir ailedeki belirli bir dağılımdan üretildiğini varsayar (Dobson ve Barnett, 2018). Dağılımın ortalama μ'si, aşağıdaki fonksiyon aracılığıyla X bağımsız değişkenlerine bağlıdır:

$$E(Y|X) = \mu = g^{-1}(X\beta) \tag{2.33}$$

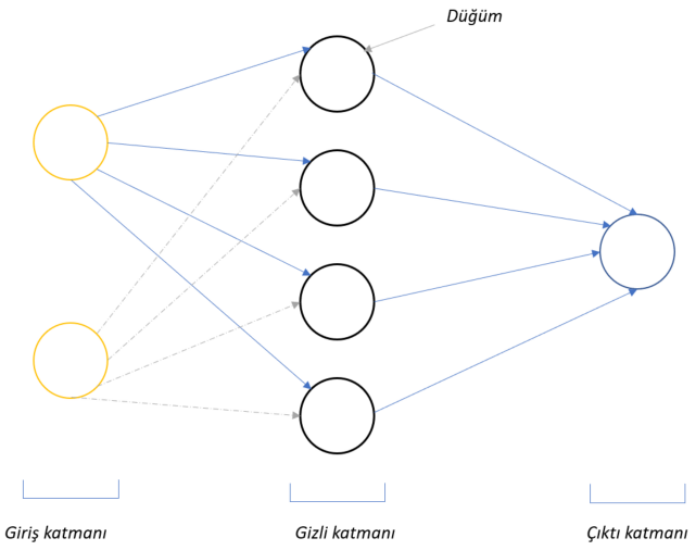
Burada $E(Y|X)$, Y'nin X koşuluna bağlı beklenen değeridir; $X\beta$, bilinmeyen β parametrelerinin doğrusal bir kombinasyonu olan doğrusal tahmin edicidir; g bağlantı fonksiyonudur(Gotway ve Stroup, 1997; 'Generalized linear model', 2021; Molnar, t.y.).

2.6.9 Yapay Sinir Ağları

Artificial Neural Networks- ANN olarak da adlandırılan Yapay Sinir Ağları, yapısı insan beyninin işleyişinden ilham alan makine öğrenme algoritmalarıdır (Zou ve diğ., 2009). Sınıflandırma ve regresyon gibi farklı makine öğrenimi görevlerinde yer alabilirler. Makine öğrenmesinde birçok sınıflandırma algoritması olduğu gibi; Sinir Ağları ile bilgisayar, etiketlenmiş eğitim verilerini analiz ederek bir görevi gerçekleştirmeyi öğrenir. İnsan beyni gibi

Sinir Ağları, ağırlık adı verilen önem faktörleriyle birbirine bağlanan binlerce düğümden (*Neuron* veya *Unit* olarak da adlandırılır) oluşur (Faraggi ve Simon, 1995).

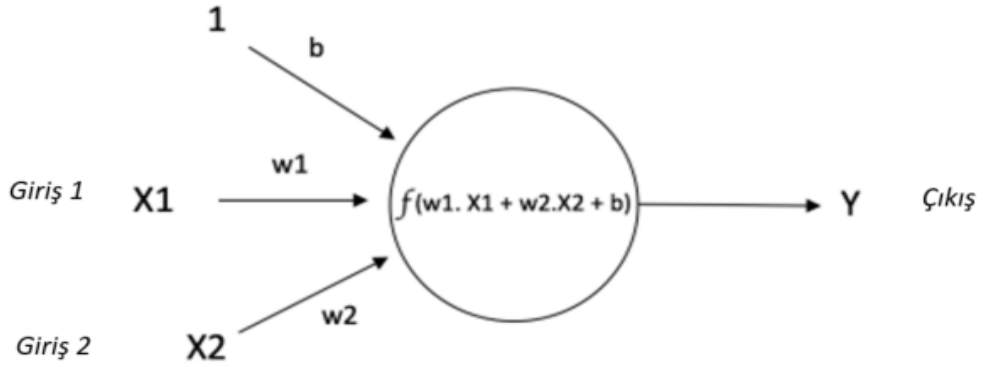
Tipik bir sinir ağı genellikle farklı katmanlardan oluşur (her katman kendi girişlerinde farklı dönüşümler gerçekleştirir) ve 3 tipe bölünebilir: bir girdi katmanı, gizli katmanlar ve bir çıktı katmanı. Bir veri seti temsil eden sinyaller ilk katmandan (Giriş katmanı) hareket eder, gizli katmanları geçer (Kara kutu olarak kabul edilebilecek bu katman, algoritmanın hata payı minimum olana kadar ağırlıklarda ince ayar yaptığı yerdir. Düğümler ağırlıklarla birbirine bağlanır) ve son olarak çıktı katmanına (sınıflandırmanın yapıldığı yer) ulaşır (Zou, ve diğ., 2009; Chollet, 2018; 'Artificial neural network', 2021). Şekil 2.13, bir Temel Sinir Ağı örneğini göstermektedir.



Şekil 2.13: Basit sinir ağı (ujjwalkarn, 2016).

Sinir ağındaki her düğüm, girdi(ler) ve çıktıya sahip doğrusal olmayan bir f fonksiyonundan (genellikle Aktivasyon Fonksiyonu olarak adlandırılır) oluşur. Şekil 2.14'den; matematiksel olarak bu fonksiyon şu şekilde tanımlanabilir: $f = f(w1.X1 + w2.X2 + b)$

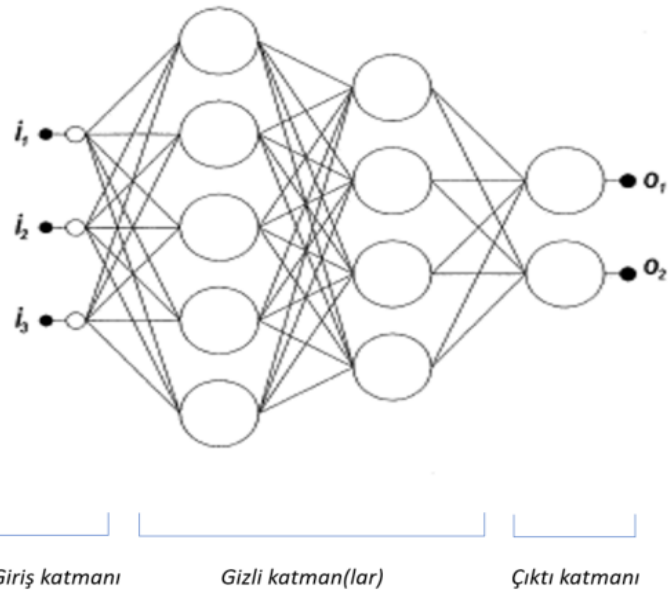
$w1$ ve $w2$, $X1$ ve $X2$ girişlerine atanan ağırlığı temsil eder. Aynı zamanda, bu düğümle ilişkili (yanlılığı temsil eden) ağırlığı b olan fazladan 1 girişi vardır.



Şekil 2.14: Tek düğüm (ujjwalkarn, 2016).

Çok Katmanlı Algılayıcı (MLP), bir veya daha fazla gizli katman içeren (bir giriş ve bir çıkış katmanı dışında) bir tür beslemeli *(feedfoward)* ağdır (Gardner ve Dorling, 1998). Tek katmanlı bir algılayıcı yalnızca giriş ve çıkış katmanını içerebilir ve yalnızca doğrusal işlevleri öğrenebilirken, çok katmanlı bir algılayıcı da doğrusal olmayan işlevleri öğrenebilir ve yeni, test verilerle sunulduğunda doğru bir şekilde genelleme yapmak üzere eğitilebilir. MLP algoritması, tahmin, fonksiyon yaklaşımı veya model sınıflandırma gibi çok çeşitli görevlere uygundur.

Çok katmanlı algılayıcı, aşağıdaki şekilde gösterilen basit birbirine bağlı nöronlardan oluşan bir sistemden oluşur (Gardner ve Dorling, 1998). Giriş katmanında, gizli katmana beslenen X1, X2 ve 1 düğümleri vardır. Daha önce de belirtildiği gibi, gizli katmanda, sonuç olarak çıktı katmanına beslenmek için birçok tür hesaplama gerçekleştirilir, bu hesaplama sonuçları Çok Katmanlı Algılayıcının çıktılarını temsil eder.



Şekil 2.15: İki gizli katmana sahip çok katmanlı bir algılayıcı (Gardner ve Dorling, 1998).

Temel bileşen analizi (PCA), orijinal veri kümesindeki bilgilerin çoğunu korumaya çalışırken büyük veri kümelerinin boyutsallığını azaltmayı veya karmaşıklıklarını basitleştirmeyi amaçlayan matematiksel bir algoritmadır (Ringnér, 2008; Jolliffe ve Cadima, 2016).

n nitelik, değişken veya boyuttan oluşan bir veri kümesi için; PCA, verileri temsil etmek için en iyi kullanılabilecek k n-boyutlu ortogonal vektörleri arar (burada $k \leq n$). Orijinal veriler böylece çok daha küçük bir alana yansıtılır ve bu da boyutsallık azalmasına neden olur. PCA alternatif ve daha küçük bir değişken seti oluşturarak özniteliklerin özünü birleştirir(Han ve diğ., 2011). İlk veriler daha sonra bu daha küçük kümeye yansıtılabilir. PCA genellikle daha önce şüphelenmeyen ilişkileri ortaya çıkarır ve böylece normalde sonuçlanmayacak yorumlara izin verir.

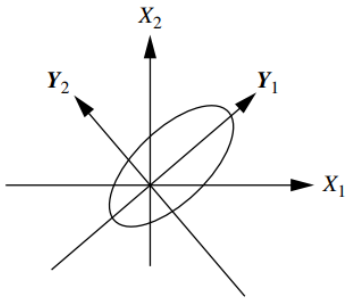
PCA algoritması 4 aşamaya ayrılabilir (Han ve diğ., 2011):

İlk olarak, her özniteliğin aynı aralıkta kalmasını sağlamak için veri kümesi normalleştirilir.

İkinci olarak, k ortonormal vektörler, normalleştirilmiş girdi verileri için bir temel sağlayan PCA tarafından hesaplanır. Temel Bileşenler olarak adlandırılan bu vektörler, her biri diğerine dik bir yönü gösteren birim vektörlerdir. Nihai girdi verileri, temel bileşenlerin doğrusal bir birleşimidir.

Üçüncüsü, azalan “önem” veya güç sırasına göre sıralanan temel bileşenler, esasen veriler için yeni bir eksen seti olarak hizmet eder ve varyans hakkında önemli bilgiler sağlar. Aşağıdaki Şekil 2.16, X1 ve X2 eksenlerine orijinal olarak eşlenen veri seti için ilk iki ana bileşeni, Y1 ve Y2'yi göstermektedir. Bu bilgi, verilerdeki kalıpları veya grupları tanımlamaya yardımcı olur.

Dördüncüsü, düşük varyanslı bileşenlerden oluşan daha zayıf bileşenler ortadan kaldırılarak veri boyutu azaltılabilir ve en güçlü ana bileşenler kullanılarak, orijinal verilerin iyi bir yaklaşımı yeniden oluşturulabilir.



Şekil 2.16: Temel bileşenler Analizi (Han ve diğ., 2011).

PCA algoritması, orijinal veri kümesindeki bilgilerin çoğunu korurken, yüzlerce değişkenli veri kümelerini işlemek için uygundur.

2.6.9.1 PCA sinir ağları / Öznitelik çıkarmalı sinir ağları

Öznitelik çıkarmalı sinir ağları olarak da adlandırılan PCA Sinir Ağları (*PCA neural networks / Neural networks with feature extraction*) hem PCA hem de Sinir Ağlarının birleştirilmesinin sonucudur. Sinir ağlarının girdisi olarak PCA'nın sonucunu kullanmaktan oluşur (Ranaie ve diğ., 2018). Başka bir deyişle, PCA Sinir Ağı, bir sinir ağı içinde uygulanan PCA'nın sonucudur. Bu işlem geri döndürülemez olduğundan, PCA Sinir Ağlarında verilerin indirgenmesi sadece giriş değişkenleri için yapılır (Liu ve diğ., 2007).

2.7 LİTERATÜR TARAMASI

Tez konusu ila ilgili *"Ensemble learning" AND "nsclc" AND "microarray" AND "variance"* anahtar kelimeleriyle *scholar.google.com* adresinde yapılan kaynak araştırmasında 30 makale

bulunmuştur. Bu makaleler incelendiğinde sadece 4 makalenin tez konusu ile benzerlik gösterdiği belirlenerek aşağıda detayları sunulmuştur;

Fishel ve diğ. (2007) tarafından yürütülen çalışma, bir dizi sınıflandırıcı geni bulmayı amaçlamaktadır ve topluluk öğrenme kavramından ilham aldılar. Bunun için, sınıflandırma sorusunu yanıtlamak için ifade düzeyi kullanılabilecek bir dizi sınıflandırıcı genini bulmaktan oluşan yeni bir meta-analiz tekniği önerdiler. Akciğer kanseri mikrodizi veri kümelerine uygulanan önerdikleri yöntemler, çok az sayıda üst sıradaki gen kullanılarak başarılı bir sınıflandırma göstermiştir (Fishel ve diğ., 2007).

Küçük hücreli olmayan akciğer kanseri için yeni görüntü belirteçleri bulma çalışmaları sırasında; Wang ve diğ. (2014), yeni görüntü belirteçleri kullanarak NSCLC bilgisayar destekli teşhis ve hayatta kalma analizi için entegre bir çerçeve (*framework*) önermektedir. Çerçeveleri, hücre tespiti, bölütleme, sınıflandırma, görüntü belirteçlerinin keşfi ve hayatta kalma analizi gibi farklı bölümleri içerir. Bu çerçevenin bir değerlendirmesinde, yüksek boyutlu verileri işleyebilen sekiz farklı sınıflandırma tekniği arasında, rastgele orman ve AdaBoost topluluğu öğrenme modellerinin NSCLC için en iyi sınıflandırma performansına sahip olduğunu göstermektedir (Wang ve diğ., 2014).

Rastgele orman, eğitim zamanında çok sayıda karar ağacı oluşturarak çalıştığı için bir topluluk öğrenme yöntemi olarak kabul edilir. Xiao ve diğ. (2014), tahmine dayalı modelleri mevcut hastaların özniteliklerine etkin bir şekilde uyarlayabilen ve bu nedenle tahmin doğruluğunu önemli ölçüde iyileştirebilen yeniden ağırlıklı bir rastgele orman modeli (*reweighted Random Forest*-RWRF) önerdi. Başka bir deyişle, RWRF eğitim ve test verileri arasındaki potansiyel heterojenliği hesaba katmak için ek hasta bilgileri mevcut olduğunda her karar ağacının ağırlığını günceller. Bir akciğer kanser veri setinde test edilen önerdikleri model, çalışmalarının başında %0,82 iken, genel olarak %0,92 doğruluk göstermiştir (Xiao ve diğ., 2014).

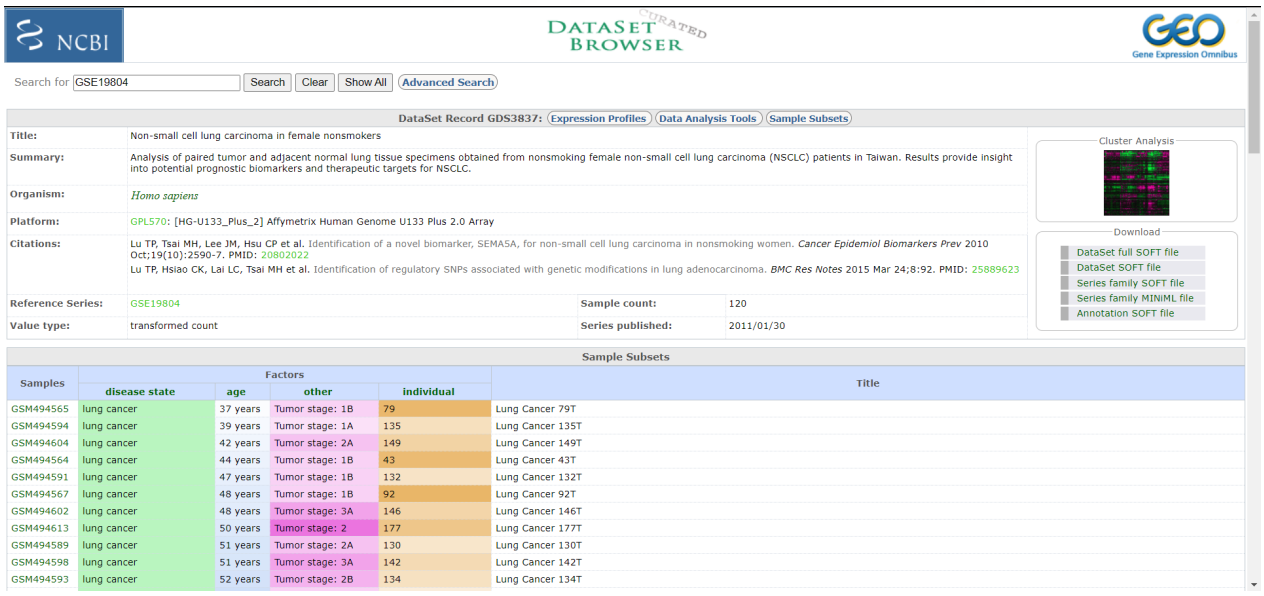
Huang ve diğ. (2016), Rastgele orman algoritmasının parametreleştirilmesinin önemini kanıtlamaya çalıştılar. Onlara göre Random Forest, parametreleri dikkatle seçildiğinde daha iyi performans gösteriyor. Bu hipoteze dayanarak, bir mikrodizi veri kümesi sınıflandırma görevini çözmek için rastgele ormanı parametrelendirdikleri bir çalışma yürüttüler. Sonuç olarak, rastgele ormanın parametreleştirilmesi, tahmin doğruluğu ile yüksek oranda ilişkilidir (Huang ve Boutros, 2016).

3 MALZEME VE YÖNTEM

3.1 VERİ

Bu tez kapsamında, Şekil 3.1'de gösterildiği gibi NCBI (*The National Center for Biotechnology Information*) *https://www.ncbi.nlm.nih.gov/geo/* halka açık web sitesinden bulunan *Gene Expression Omnibus* (GEO) veri tabanında yer alan GSE19804 seri referans numaralı DNA mikrodizi veri seti kullanılmıştır. 120 örnekten oluşan bu mikrodizi veri seti, Tayvan'da yaşayan sigara içmeyen Küçük Hücreli Olmayan Akciğer Kanseri (Non-Small Cell Lung Cancer - NSCLC) kadın hastalarından elde edilen eşleştirilmiş tümör ve komşu (*adjacent*) normal akciğer dokusu örneklerinin bir analizi ile elde edilmiştir (Lu ve diğ., 2010, 2015).

Mikrodizi veri seti, Affymetrix U133plus2.0 ifade dizileri modeli kullanılarak oluşturuldu ve her ikisi de sigara içmeyen Tayvanlı kadınlardan oluşan 60 NSCLC örneğinden ve 60 kontrol örneğinden oluştu, bu da onu dengeli bir veri seti haline getirmektedir. Mikro dizilerin NCBI'nin GDS tarayıcısına yüklemeden önce gerçekleştirilen kalite kontrolleri, bu veri setini araştırmamız için uygun hale getirmektedir.



Şekil 3.1: GEO veri tabanından GSE19804 veri setin erişimin ekran görüntüsü.

Orijinal veri seti, farklı hastaları tanımlayan 120 örneklemden (satır) ve ölçülen farklı genleri tanımlayan 54675 adet öznitelikten (sütun) oluşmaktadır. Yoğunluğa göre alınan nitelikler nümerik değerlerden oluşmaktadır. Örnek görüntüsü Şekil 3.2'de bulunmaktadır,

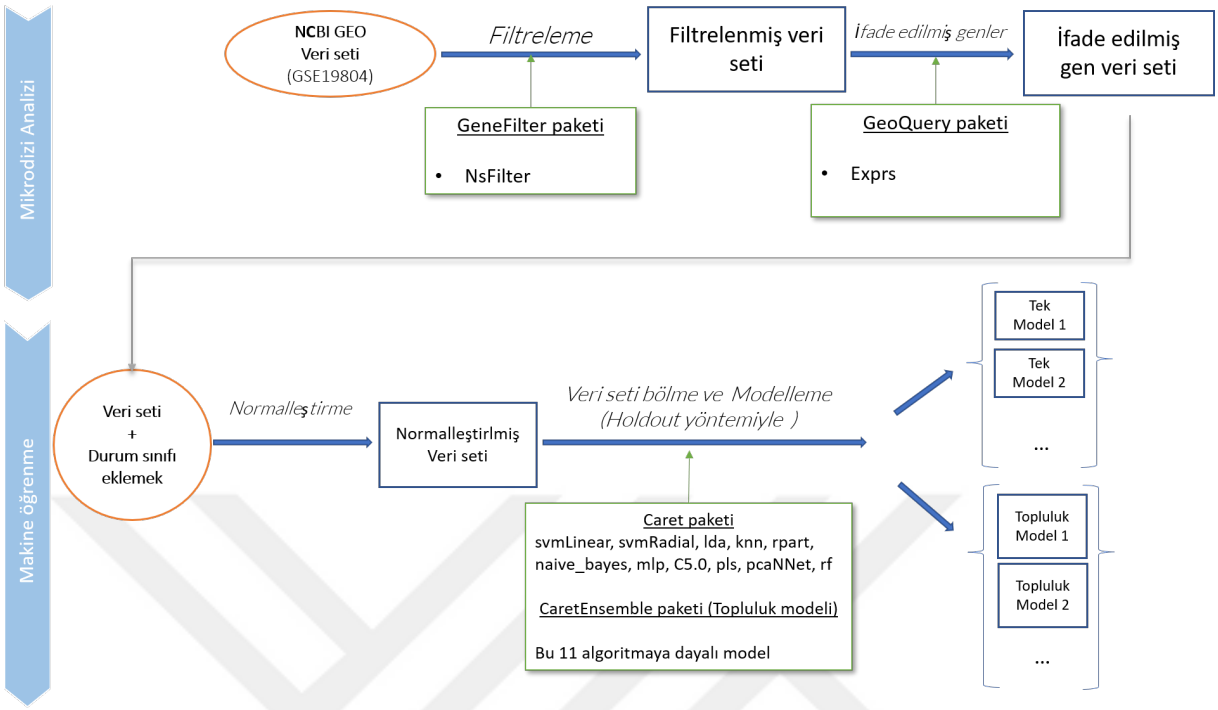
	A	B	C	D	E	F	G	H	I	J	K	L
1		1007_s_at	1053_at	117_at	121_at	1255_g_at	1294_at	1316_at	1320_at	1405_i_at	1431_at	1438_at
2	GSM494565	3.522369672	3.076798513	3.398528233	3.023761316	2.021306129	3.149358312	2.472205876	2.632143952	3.358761485	2.02609559	2.7008963
3	GSM494594	3.503374179	2.877316416	3.112262713	3.137362034	1.863285719	3.29059839	2.534541217	2.620673477	3.391135303	2.146474844	2.9176141
4	GSM494604	3.363255197	3.011689127	2.971892763	3.07747118	1.961936834	3.053666768	2.473371747	2.707016029	3.415744472	2.192529148	2.8993793
5	GSM494564	3.518862415	2.964105535	2.874331301	3.125129626	2.131358536	3.136959487	2.507942242	2.493256166	3.51215102	2.186816773	2.8445725
6	GSM494591	3.662490364	3.14333355	3.053120549	3.12760724	1.911727496	3.330357135	2.485042551	2.536304044	2.972849749	2.042317825	3.0870337
7	GSM494567	3.528083671	3.001011846	2.915334919	3.062578139	1.946433243	3.178646395	2.479630399	2.531404542	3.48610937	2.070391589	2.8186512
8	GSM494602	3.586980861	3.07367033	2.844427702	3.049374193	1.923347039	3.053120549	2.458588926	2.614403057	3.300962734	2.099577178	2.8807329
9	GSM494613	3.555938828	2.948770725	2.939300184	3.100838276	1.86409242	3.143301683	2.417743484	2.579848309	3.034328201	2.131310981	2.7865333
10	GSM494589	3.558512553	3.062694623	3.006296528	3.015462764	1.886918658	3.093165914	2.525283485	2.540492072	2.965789794	2.187591964	2.9129388
11	GSM494598	3.558659484	3.274096978	3.096077976	3.049070961	1.948426947	3.032801767	2.463122967	2.700185004	3.252709819	2.102303768	2.855737
12	GSM494593	3.665961186	2.97377629	2.975093997	3.062034348	1.951802925	3.121141383	2.496401373	2.677880378	3.264060909	2.286967494	3.0127296
13	GSM494583	3.507031342	2.993708066	2.942171893	3.130612116	1.863129017	3.119523555	2.485426999	2.626485921	2.914814008	2.068588267	2.8131096
14	GSM494612	3.584060534	2.886349645	2.755670869	3.134489873	1.949162025	2.900214876	2.412084483	2.659377338	3.086558242	2.151367852	2.9848233
15	GSM494558	3.484498223	2.860208338	2.836909209	3.075768055	1.999484201	3.252998396	2.432316263	2.510625272	3.417244936	2.158552709	2.6771504
16	GSM494556	3.580481893	2.818956537	2.712784129	3.076985846	1.899817024	3.204423535	2.498458405	2.647505493	3.209148626	2.026457396	2.7570415
17	GSM494559	3.573326048	2.984061162	2.835694451	3.116702544	1.884283102	3.050898333	2.554699311	2.523688719	3.146847167	2.196702453	2.6392049
18	GSM494571	3.432664716	2.849593323	3.05690479	3.13581171	1.856398002	3.207631066	2.552618053	2.505086491	3.377665179	2.145717094	2.8107654
19	GSM494614	3.647602329	3.379995171	2.86282087	3.03976309	1.867759829	3.124015696	2.411310852	2.598414676	2.957011069	2.043716793	2.7915713
20	GSM494603	3.502291559	3.048601638	3.089546949	3.060331916	1.863606504	3.090171856	2.466161241	2.564809855	3.372408896	2.078313853	2.8171421
21	GSM494568	3.447048833	3.031837471	3.048852873	3.065893918	1.934362442	3.102837323	2.464148877	2.586678607	3.363213138	2.097831621	2.7330130
22	GSM494572	3.460598418	2.870660068	2.848771496	3.211718074	1.877923871	3.240660446	2.475367274	2.557081168	3.14661535	2.046494228	3.1865552
23	GSM494600	3.546116584	3.009337512	3.174803965	3.128121138	1.876260943	3.142011208	2.53918059	2.471146568	3.10376499	2.162798951	2.8209113
24	GSM494562	3.425136327	3.293392201	3.151443219	3.148717576	2.053047861	3.000163586	2.428435427	2.704208585	3.192353	2.08947041	2.960646
25	GSM494615	3.570376504	2.925764075	2.968601981	3.029338929	1.860207061	3.117990324	2.381282542	2.534571076	3.124636701	2.074312127	2.6873729
26	GSM494582	3.609317644	3.085587251	2.764636247	3.085894798	1.943050912	2.99102719	2.575351726	2.777975819	2.901757615	2.893886767	3.3211437
27	GSM494599	3.493876899	3.034502174	2.892647434	2.99995496	1.860322991	3.124666516	2.483944919	2.702570244	3.427364139	2.172968626	2.8821995
28	GSM494610	3.542406646	3.048288187	3.286030542	3.16040679	2.047841707	3.170609585	2.456255967	2.642381078	3.557777671	2.117263497	2.8941676

Şekil 3.2: Orijinal veri setinin MS Excel'deki örnek görünümü.

3.2 MİKRODİZİ VERİSETİNİN ÖN İŞLENMESİ

Veri önişleme, algoritmaların performansını artırmak için verileri hazırlamak, temizlemek ve daha kullanışlı hale getirmek için kullanılan veri madenciliği sürecinde önemli bir adımdır (Han ve diğ., 2011; Wikipedia, 2021).

Veri setinin orijinal versiyonu analiz için elde edilmiştir; DNA mikrodizi veri setlerde olan tekrarlanan öznitelikler ve ifade edilmeyen genler kaldırıldı ve ardından sonuç sütunu (yani hastalık durumu) eklendi. Mikrodizi önişleme yöntemleriyle tekrarlanan sütunlar kaldırılarak veri seti R programlama dili ile analize uygun hale getirilmiştir.

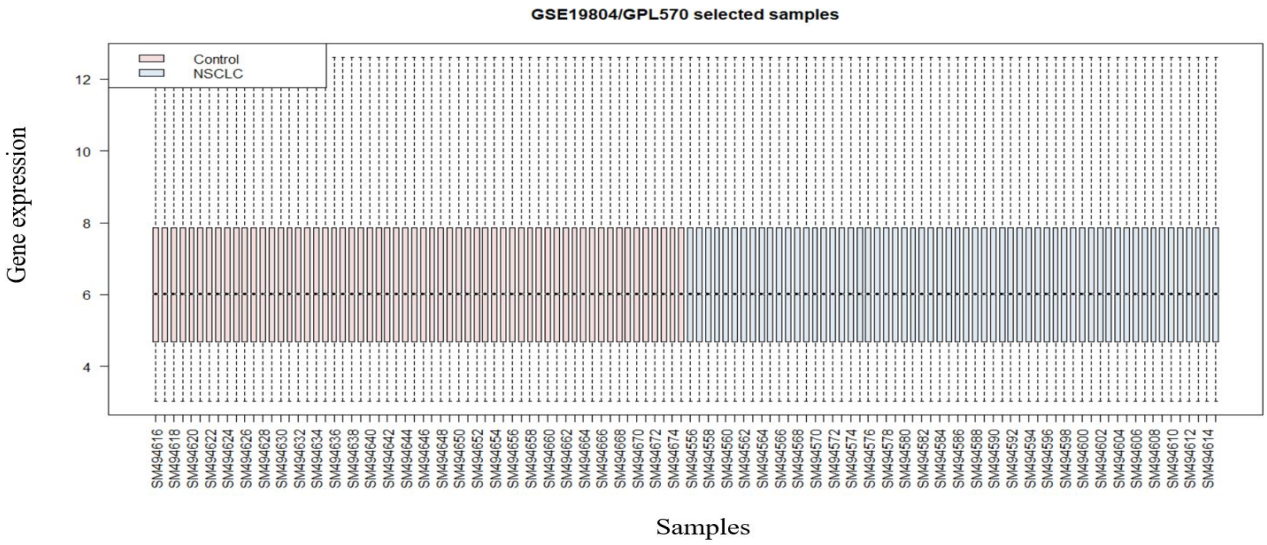


Şekil 3.3: Tez analiz süreci.

Bu tez sırasında yapılan analiz Şekil 3.3'de göründüğü gibi mikrodizi önişleme ve makine öğrenimi bölümü olmak üzere iki bölüme ayrılmıştır.

3.2.1 Mikrodizi Veri seti Analizi

Normalleştirme, numuneler arasındaki sistematik biyolojik farklılıkları daha net görmek için çipler arasındaki sistematik teknik farklılıkları telafi etmek olan bir yöntemdir (Fujita ve diğ., 2006). NCBI'dan elde edilen veri kümesi Şekil 3.4'de göründüğü gibi halihazırda normalleştirilmiştir, bu nedenle Arka plan düzeltme, mikrodizi normalleştirme, Mükemmel Eşleşme (PM) düzeltmesi gibi mikro dizi ön işleme adımları uygulanmamıştır. İfade edilen gen veri seti, Bioconductor'un Geoquery paketi kullanılarak elde edildi (Davis ve Meltzer, 2007). Elde edilen ifade edilmiş gen veri seti, tekrarlanan genlerden (öznitelikler) oluşturuldu. Bu sorunun üstesinden gelmek için, Genefilter paketindeki nsFilter() işlevi kullanıldı ve tekrarlanan öznitelikler kaldırıldı (Gentleman *ve diğ.*, 2021).



Şekil 3.4: Normalleştirilmiş veri kümesinin kutu grafiği.

3.2.1.1 Veri seti filtrelenmesi

Filtreleme, mikrodizi veri analizinde en önemli adımlardan biridir. Filtreleme, mikrodizi veri kümesinden (bu çalışmadaki genler) tekrarlayan özniteliklerin kaldırılmasından oluşan bir işlemdir. 54675 öznitelik ve 120 örnekten oluşan orijinal veri setinin filtreleme işleminden sonra, 10087 özelliğe ve 120 örneğe indirgenmiştir (Şekil 3.5). Filtreleme işlemi, genefilter paketi tarafından sağlanan nsFilter () fonksiyonu kullanılarak gerçekleştirilmiştir (Gentleman ve diğ., 2021).

```
> dim(eset)
Features  Samples
   54675      120
> annotation(eset) <- "hgu133plus2.db"
> filtrelenmis <- nsFilter(eset)$eset
> dim(filtrelenmis)
Features  Samples
   10087      120
>
```

Şekil 3.5: Filtrelenmiş veri seti.

3.2.1.2 Karar sınıfını içeren sütun oluşturulması

Orijinal veri setinin ek açıklamasında yer alan bilgilere göre "Durum" adlı bir sütun oluşturuldu (Şekil 3.6). Aslında, veri setinin ek açıklaması iki kategorize edilmiş veri setinden oluşuyordu: kanser hastası için "lung cancer" ve hasta olmayan hasta için "control". Adından da anlaşılacağı

üzere "Durum" sütununun amacı, normal hastaları Kanserli hastalardan ayırmaktır. Bu nedenle, iki kategorik değer içerir: kanser hastası için 'lung cancer' ve kanser olmayan için 'control' (Şekil 3.7).

```
> colnames(pData(eset2))
[1] "sample"        "disease.state" "age"           "other"         "individual"    "description"
> pData(eset)$disease.state
  [1] lung cancer lung cancer lung cancer lung cancer lung cancer lung cancer lung cancer lung cancer lung cancer
 [10] lung cancer lung cancer lung cancer lung cancer lung cancer lung cancer lung cancer lung cancer lung cancer
 [19] lung cancer lung cancer lung cancer lung cancer lung cancer lung cancer lung cancer lung cancer lung cancer
 [28] lung cancer lung cancer lung cancer lung cancer lung cancer lung cancer lung cancer lung cancer lung cancer
 [37] lung cancer lung cancer lung cancer lung cancer lung cancer lung cancer lung cancer lung cancer lung cancer
 [46] lung cancer lung cancer lung cancer lung cancer lung cancer lung cancer lung cancer lung cancer lung cancer
 [55] lung cancer lung cancer lung cancer lung cancer lung cancer lung cancer control     control     control
 [64] control     control     control     control     control     control     control     control     control
 [73] control     control     control     control     control     control     control     control     control
 [82] control     control     control     control     control     control     control     control     control
 [91] control     control     control     control     control     control     control     control     control
[100] control     control     control     control     control     control     control     control     control
[109] control     control     control     control     control     control     control     control     control
[118] control     control     control
Levels: control lung cancer
```

Şekil 3.6: Orijinal veri setinin ek açıklamasında yer alan bilgiler.

```
> durum <- pData(eset)$disease.state
> table(durum)
durum
    control lung cancer
         60          60
> levels(durum)
[1] "control"     "lung cancer"
> levels(durum)[levels(durum)=="lung cancer"] <- "lung_cancer"
> levels(durum)[levels(durum)=="control"] <- "control"
> table(durum)
durum
    control lung_cancer
         60          60
```

Şekil 3.7: Durum sütunu oluşturulması.

3.2.1.3 Veri kümesinin öznitelik ölçeklendirmesi

Makine öğreniminde öznitelik ölçeklendirme, bir makine öğrenimi modeli oluşturmadan önce verilerin ön işlenmesi sırasında en önemli adımlardan biridir. Ölçeklendirme, iki model arasında büyük bir fark yaratabilir. En yaygın öznitelik ölçekleme teknikleri normalleştirme ve standardizasyondur. Normalleştirme, iki sayı arasındaki değerleri sınırlamak için kullanılırken (tipik olarak [0,1] veya [-1,1] arasında), standardizasyon verileri sıfır ortalamaya ve 1 varyansına dönüştürür, verileri birimsiz (*Unitless*) yapar (Becker ve diğ., 1988; Dzierżak, 2019). Bu tez kapsamında kullanılan öznitelik ölçeklendirme yöntemi standartlaştırmadır. Analiz edilen veri setine ait normalleştirmeden önce (Şekil 3.8) ve sonra (Şekil 3.9), filtrelenmiş veri kümesinin ilk 25 özniteliğinin özeti Şekil 3.8 ve Şekil 3.9 da yer almaktadır.

```
> summary(cbind(durum, as.data.frame(filtrelenmisVeri[1:120 , 1:25])))
        durum      204639_at       203440_at       222880_at       207078_at       222161_at       228647_at
 control    :60   Min.   :2.407   Min.   :2.228   Min.   :2.066   Min.   :1.746   Min.   :1.842   Min.   :2.660
 lung_cancer:60   1st Qu.:2.841   1st Qu.:2.370   1st Qu.:2.262   1st Qu.:2.041   1st Qu.:1.956   1st Qu.:2.910
                  Median :2.906   Median :2.416   Median :2.386   Median :2.330   Median :2.067   Median :3.023
                  Mean   :2.919   Mean   :2.461   Mean   :2.441   Mean   :2.300   Mean   :2.103   Mean   :3.005
                  3rd Qu.:3.006   3rd Qu.:2.533   3rd Qu.:2.599   3rd Qu.:2.489   3rd Qu.:2.235   3rd Qu.:3.106
                  Max.   :3.282   Max.   :2.924   Max.   :3.001   Max.   :3.231   Max.   :2.598   Max.   :3.263
   236514_at       209027_s_at      222922_at       214631_at       203256_at       210458_s_at      235798_at
 Min.   :2.190   Min.   :2.358   Min.   :2.011   Min.   :1.605   Min.   :2.682   Min.   :1.970   Min.   :1.961
 1st Qu.:2.375   1st Qu.:3.183   1st Qu.:2.159   1st Qu.:1.767   1st Qu.:2.846   1st Qu.:2.284   1st Qu.:2.481
 Median :2.437   Median :3.257   Median :2.276   Median :1.834   Median :3.030   Median :2.461   Median :2.643
 Mean   :2.477   Mean   :3.232   Mean   :2.321   Mean   :1.859   Mean   :3.037   Mean   :2.489   Mean   :2.593
 3rd Qu.:2.540   3rd Qu.:3.330   3rd Qu.:2.440   3rd Qu.:1.926   3rd Qu.:3.212   3rd Qu.:2.720   3rd Qu.:2.736
 Max.   :3.007   Max.   :3.454   Max.   :3.316   Max.   :2.316   Max.   :3.527   Max.   :2.947   Max.   :2.931
  222314_x_at      1570040_at      1557961_s_at     244741_s_at      1559045_at      239894_at       240929_at
 Min.   :2.235   Min.   :2.206   Min.   :1.753   Min.   :2.315   Min.   :1.771   Min.   :2.229   Min.   :2.517
 1st Qu.:2.435   1st Qu.:2.372   1st Qu.:2.258   1st Qu.:2.820   1st Qu.:1.920   1st Qu.:2.347   1st Qu.:2.716
 Median :2.518   Median :2.454   Median :2.435   Median :2.979   Median :1.980   Median :2.423   Median :2.810
 Mean   :2.529   Mean   :2.445   Mean   :2.421   Mean   :2.917   Mean   :1.985   Mean   :2.424   Mean   :2.804
 3rd Qu.:2.577   3rd Qu.:2.499   3rd Qu.:2.618   3rd Qu.:3.064   3rd Qu.:2.041   3rd Qu.:2.490   3rd Qu.:2.885
 Max.   :3.265   Max.   :2.860   Max.   :2.852   Max.   :3.256   Max.   :2.482   Max.   :2.688   Max.   :3.116
   239799_at       235174_s_at      236118_at       229691_at       1559870_at
 Min.   :2.173   Min.   :2.436   Min.   :2.444   Min.   :2.499   Min.   :2.501
 1st Qu.:2.479   1st Qu.:2.977   1st Qu.:2.727   1st Qu.:2.732   1st Qu.:2.730
 Median :2.559   Median :3.048   Median :2.831   Median :2.802   Median :2.809
 Mean   :2.568   Mean   :3.024   Mean   :2.809   Mean   :2.808   Mean   :2.813
 3rd Qu.:2.652   3rd Qu.:3.121   3rd Qu.:2.917   3rd Qu.:2.891   3rd Qu.:2.878
 Max.   :2.895   Max.   :3.299   Max.   :3.195   Max.   :3.161   Max.   :3.163
>
```

Şekil 3.8: Normalleştirmeden önce filtrelenmiş veri kümesinin ilk 25 özniteliğinin özeti.

```
> sonVeri <- cbind(durum, as.data.frame(scale(filtrelenmisVeri, center = TRUE, scale = TRUE)))
> summary(sonVeri[1:120 , 1:25])
        durum      204639_at         203440_at         222880_at         207078_at         222161_at
 control    :60   Min.   :-3.58721   Min.   :-1.6008   Min.   :-1.6309   Min.   :-1.79159   Min.   :-1.4772
 lung_cancer:60   1st Qu.:-0.54374   1st Qu.:-0.6240   1st Qu.:-0.7796   1st Qu.:-0.83668   1st Qu.:-0.8299
                  Median :-0.08922   Median :-0.3059   Median :-0.2383   Median : 0.09728   Median :-0.2025
                  Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.0000
                  3rd Qu.: 0.60715   3rd Qu.: 0.4971   3rd Qu.: 0.6884   3rd Qu.: 0.61157   3rd Qu.: 0.7481
                  Max.   : 2.53797   Max.   : 3.1784   Max.   : 2.4352   Max.   : 3.01437   Max.   : 2.8047
   228647_at         236514_at         209027_s_at       222922_at         214631_at         203256_at
 Min.   :-2.5529   Min.   :-1.8798   Min.   :-5.4502   Min.   :-1.4750   Min.   :-2.0924   Min.   :-1.55752
 1st Qu.:-0.7035   1st Qu.:-0.6663   1st Qu.:-0.3101   1st Qu.:-0.7728   1st Qu.:-0.7559   1st Qu.:-0.83592
 Median : 0.1372   Median :-0.2631   Median : 0.1510   Median :-0.2171   Median :-0.2101   Median :-0.03219
 Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.00000
 3rd Qu.: 0.7529   3rd Qu.: 0.4140   3rd Qu.: 0.6060   3rd Qu.: 0.5668   3rd Qu.: 0.5546   3rd Qu.: 0.76521
 Max.   : 1.9167   Max.   : 3.4784   Max.   : 1.3780   Max.   : 4.7356   Max.   : 3.7682   Max.   : 2.14344
  210458_s_at       235798_at         222314_x_at       1570040_at        1557961_s_at      244741_s_at
 Min.   :-2.0873   Min.   :-3.0688   Min.   :-2.04060   Min.   :-1.9867   Min.   :-2.63003   Min.   :-2.9300
 1st Qu.:-0.8255   1st Qu.:-0.5449   1st Qu.:-0.65348   1st Qu.:-0.6067   1st Qu.:-0.64420   1st Qu.:-0.4684
 Median :-0.1125   Median : 0.2438   Median :-0.07602   Median : 0.0739   Median : 0.05339   Median : 0.3020
 Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.0000
 3rd Qu.: 0.9275   3rd Qu.: 0.6958   3rd Qu.: 0.33759   3rd Qu.: 0.4438   3rd Qu.: 0.77469   3rd Qu.: 0.7175
 Max.   : 1.8401   Max.   : 1.6403   Max.   : 5.11776   Max.   : 3.4426   Max.   : 1.69486   Max.   : 1.6492
   1559045_at        239894_at         240929_at         239799_at         235174_s_at       236118_at
 Min.   :-2.10805   Min.   :-2.121887   Min.   :-2.29937   Min.   :-3.06948   Min.   :-3.9525   Min.   :-2.2203
 1st Qu.:-0.64411   1st Qu.:-0.836766   1st Qu.:-0.70628   1st Qu.:-0.69016   1st Qu.:-0.3127   1st Qu.:-0.5000
 Median :-0.05405   Median :-0.007632   Median : 0.04617   Median :-0.06586   Median : 0.1609   Median : 0.1313
 Mean   : 0.00000   Mean   : 0.000000   Mean   : 0.00000   Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.0000
 3rd Qu.: 0.55044   3rd Qu.: 0.720417   3rd Qu.: 0.64438   3rd Qu.: 0.65541   3rd Qu.: 0.6500   3rd Qu.: 0.6536
 Max.   : 4.88734   Max.   : 2.874799   Max.   : 2.48907   Max.   : 2.54738   Max.   : 1.8509   Max.   : 2.3452
   229691_at
 Min.   :-2.35007
 1st Qu.:-0.57333
 Median :-0.04836
 Mean   : 0.00000
 3rd Qu.: 0.63261
 Max.   : 2.68275
```

Şekil 3.9: Normalleştirmeden sonra filtrelenmiş veri kümesinin ilk 25 özniteliğinin özeti.

3.2.2 İşlenmiş Veri Seti

Ön işlem sonrası 120 örnek (satır) ve 54675 öznitelikten (sütun) oluşan veri seti 120 örnek ve 10088 öznitelik(sınıflandırma sınıfından oluşan durum sütunu ile) haline gelmiştir. Sonuç olarak veri seti, 10087 sayısal sütun ve sınıflandırma sınıfına karşılık gelen 1 kategorik sütundan oluşmuştur.

Veri setinin filtrelenmesinden sonra, ifade edilen gen veri kümesi elde edildi. Ardında "durum" sütunu ile birleştirilerek makine öğrenimi sürecine hazır hale getirildi (Şekil 3.10). Daha sonra veri seti, tek algoritma tabanlı modellerin oluşturulması için eğitim ve test setine bölünmüş ve topluluk model eğitimi için bozulmadan kullanılmıştır.

```
> sonVeri <- cbind(durum, as.data.frame(t(exprs(filtrelenmis))))
> sonVeri[1:120 , 1:10]
               durum 204639_at 203440_at 222880_at 207078_at 222161_at 228647_at 236514_at 209027_s_at 222922_at
GSM494565 lung_cancer  2.957631  2.788278  2.150289  2.352086  2.067978  2.844277  2.428522    3.154043  2.055690
GSM494594 lung_cancer  2.972457  2.406614  2.214869  2.241157  1.891560  2.940918  2.369992    2.954483  2.126716
GSM494604 lung_cancer  2.834715  2.784670  2.789146  1.966601  2.356942  3.016275  2.432978    3.360617  2.445231
GSM494564 lung_cancer  3.040402  2.923946  2.401426  2.540333  1.977225  3.022716  2.374057    3.255373  2.243258
GSM494591 lung_cancer  2.931953  2.315622  2.151292  2.806937  1.854901  2.935586  2.351533    2.962618  2.278589
GSM494567 lung_cancer  3.008074  2.640007  2.133768  2.529794  2.042262  3.142208  2.761416    3.153055  2.159165
GSM494602 lung_cancer  2.943278  2.693695  2.654929  1.926759  1.947279  3.214059  2.825840    3.327242  2.600936
GSM494613 lung_cancer  2.553831  2.312410  2.275644  2.713228  1.945037  3.263331  2.408141    3.148430  2.160044
GSM494589 lung_cancer  2.858499  2.423124  2.104616  2.891559  2.080839  2.921809  2.708576    3.140487  2.095698
GSM494598 lung_cancer  3.026335  2.332125  2.431057  2.179643  1.896687  2.775187  2.562885    3.327199  2.583026
GSM494593 lung_cancer  2.829231  2.374220  2.557174  2.202414  1.939387  2.969546  2.878547    3.274368  2.335340
GSM494583 lung_cancer  2.895410  2.575033  2.298969  2.110188  2.055903  2.797676  2.567879    3.170169  2.211718
GSM494612 lung_cancer  2.530977  2.229857  2.279539  1.967510  1.910429  3.034328  2.338527    3.321300  2.386680
GSM494558 lung_cancer  2.985021  2.605370  2.561866  2.756033  2.205337  2.684256  2.266171    3.203950  2.109022
GSM494556 lung_cancer  2.774869  2.419566  2.357961  2.396303  1.858757  2.818056  2.614420    3.238944  2.483022
GSM494559 lung_cancer  2.910935  2.380355  2.256206  2.318416  2.052403  3.049562  2.955176    3.124701  2.011434
GSM494571 lung_cancer  2.925660  2.381350  2.328900  3.231427  1.947600  2.893632  2.190428    2.808595  2.125374
GSM494614 lung_cancer  2.758604  2.838301  2.379893  2.389858  1.871621  3.023612  2.727413    3.135184  2.235001
GSM494603 lung_cancer  3.064875  2.533095  2.497853  2.477547  2.020579  2.804003  2.469639    3.199214  2.347686
GSM494568 lung_cancer  3.043410  2.585782  2.571567  2.493208  2.069173  2.771264  2.355819    3.223537  2.186999
```

Şekil 3.10: Veri kümesinin bir kısmının görüntülenmesi.

3.3 MODELLEME

İstifleme (*Stacking*), bu çalışmada uygulanan topluluk yöntemidir. Aynı veri kümesinin farklı makine öğrenimi modellerinin tahminlerini, Torbalama (*Bagging*) ve Yükseltme (*Boosting*) gibi birleştirmekten oluşur (Zenko, ve diğ., 2001). İstifleme, iki seviyeli modellemeden oluşur; bunlardan ilki (genellikle alt-modeller olarak adlandırılır) tek bir algoritmaya dayalı modellerden ve son bir tahminin elde edilmesi için alt-modeller tarafından sağlanan sonuçların birleştirilmesinden sorumlu bir modelden oluşan ikinci bir seviyeden oluşur (Graczyk ve diğ., 2010).

İstifleme kullanılırken, topluluk modelinin farklı şekillerde yetenekli olmasını sağlamak için düşük korelasyonlu alt-modellere sahip olması tercih edilir (Dubuis ve diğ., 2011); böylece ikinci sınıflandırıcının iyi bir performansı gösterebilmek için alt-modeller tarafından sağlanan farklı tahminlerden nasıl daha iyi yararlanılacağını anlamasına izin verir.

Bu çalışmada farklı tutulan algoritmalar arasındaki korelasyonu ölçmek için her şeyden önce bir modelleme yapılmış ve o modelleme sırasında yüksek korelasyona sahip olan modeller (>0,7) kaldırılmıştır (Tablo 4.1). Bu tez çalışması kapsamında biri GLM ve diğeri Doğrusal SVM algoritması olmak üzere iki topluluk modeli oluşturulmuştur. Varyansı ölçebilmek için her algoritma üzerinden 50 kere model oluşturulmuştur ve her seferinde elde edilen doğruluk değeri kaydedilmiştir.

3.3.1 Kullanılan Araçlar

Analizler, R programlama dilinde (4.0.2 versiyonu) ve RStudio (1.3.1056 versiyonu) programı kullanılarak yapılmıştır. Tezde yazılan kodlar EKLER (Ek1, Ek2 ve Ek3) bölümünde sunulmaktadır.

3.3.2 Modelleme ve Performans Değerlendirme

Tez çalışması sırasında algoritmaların sonucu değerlendirebilmek için *holdout* yöntemi benimsenmiştir. *Holdout* veri setini eğitim ve test seti olarak adlandırılan iki kısma ayırmaktan oluşan bir yöntemdir. Eğitim seti, algoritmaya dayalı bir model oluşturmayı mümkün kılar ve test seti, modelin performansının tahmin edilmesini mümkün kılar. Böylece tez kapsamında veri seti; eğitim seti için %80 ve test seti için %20 olmak üzere 2 kısma ayrıldı. Eğitim seti daha sonra basit algoritma tabanlı modeller ve topluluk modeli oluşturmak için kullanıldı (Yadav ve Shukla, 2016; Raschka, 2020). Elde edilen modellerin performansları karışıklık matrisi kullanılarak doğruluk ölçütü ile değerlendirilmiştir.

3.3.3 Algoritmalar

Tez çalışmasında toplam 12 algoritma yani Doğrusal Destek Vektör Makinesi (*Linear SVM*), Radyal Destek Vektör Makinesi (*Radial SVM*), Doğrusal Ayırıcı Analizi (*Linear Discriminant Analysis*), k-En Yakın Komşular (*k-Nearest Neighbors*), CART, Saf Bayes (*Naive Bayes*), Çok Katmanlı Algılayıcı (*Multi-Layer Perceptron*), C5.0, Kısmi En Küçük Kareler (*Partial Least*

Squares), Öznitelik Çıkarımlı Sinir Ağları (*Neural Networks with Feature Extraction*), Rastgele Orman (*Random Forest*), Genelleştirilmiş doğrusal model (*Generalized linear model*) algoritmaları analizde kullanılmıştır. Analizde kullanılan paketler ve fonksiyonlarla ilgili bilgiler Tablo 3.1'de verilmiştir.

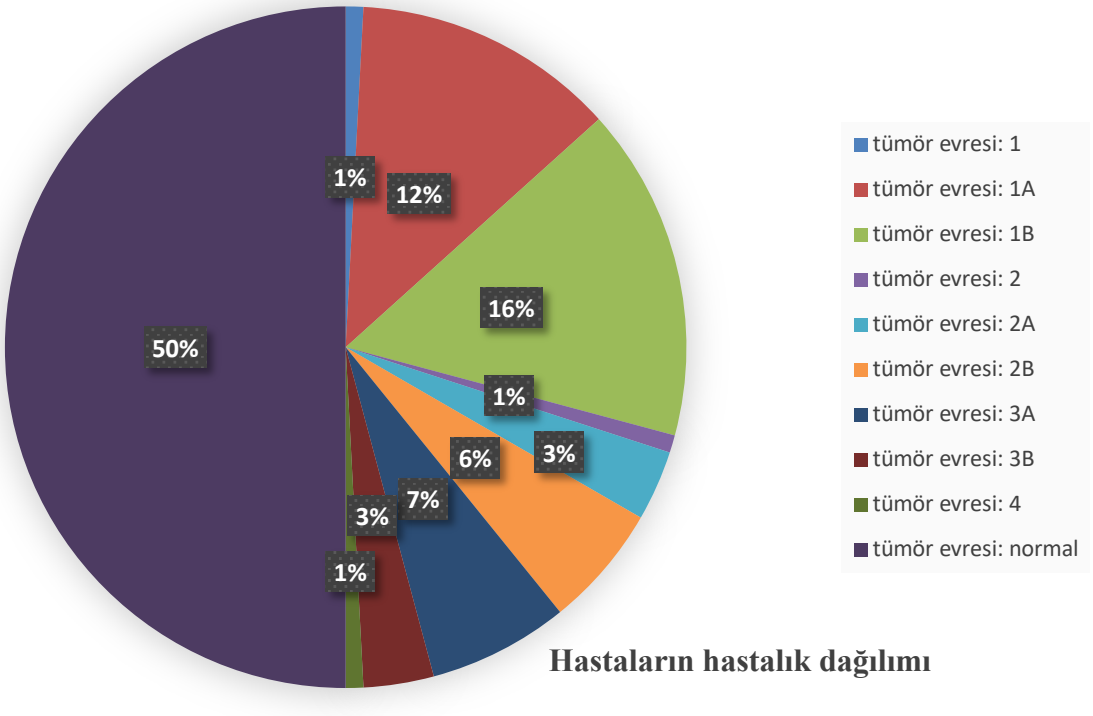
Tablo 3.1: Analizlerde kullanılan paket ve fonksiyon bilgileriÖ

Kullanılan Paket	Kullanılmış metot ve algoritmalar	Kullanım Amacı	Kaynak
Caret	Doğrusal Destek Vektör Makinesi	Model kurmak için kullanılmış algoritmaları	(Kuhn, 2015)
	Radyal Destek Vektör Makinesi		
	Linear Discriminant Analysis		
	K-En Yakın Komşular		
	CART		
	Saf Bayes		
	Çok Katmanlı Algılayıcı		
	C5.0		
	Kısmi En Küçük Kareler		
	Öznitelik Çıkarımlı Sinir Ağları		
	Genelleştirilmiş doğrusal model		
	Random Forest		
	ConfusionMatrix	Modellerin doğrululuğu ölçme	
CaretEnsemble	trainControl	Model doğrulama *(Model validation)*	(Deane-Mayer ve Knowles, 2019)
	caretStack	Topluluk modeli kurmak için kullanılan metodu	
GeoQuery	GetGeo	Veri seti elde edilmek için kullanılmış fonksiyonu	(Davis ve Meltzer, 2007)
	exprs	İfade edilmiş Gen veri kümesini çıkartmak için	
	GDS2eset	Veri seti eset set'e çevirkek	
GeneFilter	nsFilter	Filtreleme için kullanılmış fonksiyonu	(Gentleman ve diğ., 2021)

4 BULGULAR

Bu bölümde öncelikle veri seti ile ilgili bazı temel istatistikler ve ardından veri madenciliği analizi sonuçları verilmektedir.

Tez çalışmasında kullanılan veri seti, Tayvan'da sigara içmeyen kadınlardan toplanmıştır. Veri setinin hastalık evresi dağılımına bakıldığında, normal hastaların yanı sıra %16'sını temsil eden hastaların çoğunun faz 1B'de olduğu görülmektedir (Şekil 4.1). Bunun yanında faz 3A'da %7; Faz 2B'de %6; Faz 2A ve 3B'de %3; %1'i aşama 1, 2 ve 4'tedir.



Şekil 4.1: Hastaların tümör evresi dağlımı.

Şekil 4.2'de gösterildiği gibi veri seti %50 normal ve %50 kanser hastalarından oluşmaktadır, bu da onu mükemmel dengelenmiş bir veri seti yapmaktadır.

Şekil 4.2: Hastaların sınıf dağılımı.

Veri setini oluşturan 120 hastanın yaş dağılımı incelendiğinde, çoğunluğu 60 yaşında olan 37 ila 79 yaşındaki hastalardan oluştuğu görülmektedir (Şekil 4.3).

Şekil 4.3: Hastaların yaş dağılımı.

Veri seti ön işleme aşamasında sonra korelasyon analizi yapıldı ve yüksek korelasyonlu algoritmalar kaldırıldı (Tablo 4.1).

Doğrusal Destek Vektör Makinesi, Radyal Destek Vektör Makinesi, Doğrusal Ayırıcı Analizi, k-En Yakın Komşular, CART, Saf Bayes, Çok Katmanlı Algılayıcı, C5.0, Kısmi En Küçük Kareler, Öznitelik Çıkarımlı Sinir, Rastgele Orman, Genelleştirilmiş doğrusal model olmak üzere toplam 12 algoritma vardı. Tablo 4.1'de gösterildiği gibi, SVM Linear ve MLP; Doğrusal SVM ve pcaNNet; Radyal SVM ve Saf Bayes; PLS ve MLP; PLS ve pcaNNet modellerin arasında yüksek korelasyon olduğu için SVM Linear; Saf Bayes; MLP ve PLS algoritmalar analizin ikinci adımından çıkarıldı. Rastgele Orman ise **2.6.4** bölümünde bahsedildiği gibi bir topluluk öğrenme algoritma olduğu için de analizin ikinci adımında dahil edilmemiştir.

Korelasyon analizinin ardından kalan algoritmalar (Radyal SVM, LDA, KNN, rpart, C5.0, glm, pcaNNet), bir kısımda basit algoritma tabanlı modeller, diğer kısımda ise iki toplu öğrenme modeli oluşturmak için kullanılmıştır. Doğruluk ölçüsü (Fisher ve diğ., 2009; Canbek ve diğ., 2017; Chen ve diğ., 2020), bu iki model türünün performansını karşılaştırmak için kullanıldı. Veri setini test etmek için, veri setinin %80'i her iki modeli eğitmek ve %20'si test için kullanıldı.

Tablo 4.2'de gösterildiği gibi; her algoritma için 50 kez model oluşturulmuş ve her seferinde karışıklık matrisinden elde edilen doğruluk (*accuracy*) değeri kaydedilmiş ve sonunda varyans hesaplanmıştır. Sonuç olarak rastgele orman ve SVM doğrusal topluluk modelleri; GLM topluluk modeli, diğerlerine kıyasla en az varyansa sahiptir (Tablo 4.2).

Tablo 4.1: Korelasyon tablosu.

	svmLinear	svmRadial	lda	knn	rpart	naive_bayes	mlp	C5.0	pls	pcaNNet	rf	glm
svmLinear	1	0.354646	0.591064	-0.09118	0.410418	0.395137	0.726955	0.306211	0.466704	0.721077	0.171599	0.075809
svmRadial	0.354646	1	0.617331	0.220182	0.379404	0.76211	0.612615	0.252781	0.620917	0.373908	0.393371	-0.07078
lda	0.591064	0.617331	1	-0.04756	0.184597	0.441619	0.699358	0.365372	0.601308	0.496796	0.405474	0.02836
knn	-0.09118	0.220182	-0.04756	1	0.086097	0.095778	-0.08987	-0.04216	0.200477	0.071028	-0.0595	0.021557
rpart	0.410418	0.379404	0.184597	0.086097	1	0.635957	0.397333	0.651362	0.405776	0.541491	0.161701	-0.01934
naive_bayes	0.395137	0.76211	0.441619	0.095778	0.635957	1	0.678438	0.448278	0.665962	0.507124	0.574545	-0.32726
mlp	0.726955	0.612615	0.699358	-0.08987	0.397333	0.678438	1	0.365276	0.770616	0.682911	0.4118	-0.14426
C5.0	0.306211	0.252781	0.365372	-0.04216	0.651362	0.448278	0.365276	1	0.314127	0.394956	0.384442	0.291253
pls	0.466704	0.620917	0.601308	0.200477	0.405776	0.665962	0.770616	0.314127	1	0.729403	0.23603	-0.24221
pcaNNet	0.721077	0.373908	0.496796	0.071028	0.541491	0.507124	0.682911	0.394956	0.729403	1	0.150184	-0.13795
rf	0.171599	0.393371	0.405474	-0.0595	0.161701	0.574545	0.4118	0.384442	0.23603	0.150184	1	0.044289
glm	0.075809	-0.07078	0.02836	0.021557	-0.01934	-0.32726	-0.14426	0.291253	-0.24221	-0.13795	0.044289	1

Tablo 4.2: Seçilmiş algoritmaların doğruluk değerleri.

Aşama	SVM Radial	LDA	KNN	rpart	C5.0	glm	pcaNNet	SVM Linear Ensemble	GLM Ensemble
V1	0.958333	0.875	0.875	0.458333	0.958333	0.791667	0.958333333	0.916666667	1
V2	1	1	0.833333	0.958333	0.916667	0.666667	0.875	1	0.916666667
V3	0.916667	0.958333	0.875	0.875	0.958333	0.541667	0.958333333	1	0.916666667
V4	0.958333	1	1	0.875	0.958333	0.541667	0.916666667	1	0.916666667
V5	0.875	0.958333	0.875	0.916667	0.958333	0.5	1	0.958333333	0.916666667
V6	0.875	0.958333	0.958333	0.833333	0.916667	0.625	0.916666667	0.958333333	0.958333333
V7	0.916667	0.958333	0.916667	0.916667	0.916667	0.541667	1	1	0.916666667
V8	1	0.916667	0.916667	0.958333	0.958333	0.833333	1	0.875	0.916666667

Tablo 4.2 (devam): Seçilmiş algoritmaların doğruluk değerleri.

V9	0.916667	0.916667	1	0.958333	1	0.583333	1	0.958333333	0.916666667
V10	0.875	1	1	0.916667	0.958333	0.583333	1	1	0.958333333
V11	0.958333	0.833333	0.833333	0.916667	1	0.625	1	0.958333333	0.916666667
V12	0.916667	0.958333	0.833333	0.875	0.916667	0.541667	0.916666667	1	1
V13	0.958333	0.916667	0.916667	0.833333	0.958333	0.333333	0.791666667	0.958333333	0.958333333
V14	1	0.958333	0.958333	0.833333	0.916667	0.708333	0.875	0.833333333	0.958333333
V15	0.833333	0.958333	1	0.916667	0.875	0.666667	1	1	0.958333333
V16	1	0.916667	0.958333	0.916667	0.875	0.5	1	0.958333333	0.958333333
V17	1	0.875	0.958333	0.958333	0.916667	0.708333	0.916666667	0.958333333	1
V18	0.916667	0.958333	0.875	0.916667	1	0.75	0.916666667	0.958333333	0.958333333
V19	0.791667	0.916667	0.916667	0.916667	1	0.5	0.958333333	0.916666667	0.916666667
V20	1	0.916667	0.875	0.916667	0.833333	0.583333	0.875	0.875	0.875
V21	0.958333	0.958333	1	0.916667	0.875	0.541667	0.958333333	0.958333333	0.916666667
V22	0.958333	1	1	0.875	0.916667	0.333333	1	0.958333333	0.958333333
V23	0.958333	0.916667	0.916667	0.916667	1	0.416667	0.958333333	0.916666667	0.958333333
V24	0.958333	1	0.916667	0.916667	0.958333	0.708333	0.916666667	0.916666667	0.958333333
V25	0.958333	0.958333	0.958333	0.875	0.958333	0.583333	1	0.916666667	0.916666667
V26	1	0.916667	0.958333	0.875	0.958333	0.583333	0.958333333	0.958333333	0.958333333
V27	0.916667	0.958333	0.958333	0.916667	0.958333	0.625	0.833333333	1	1
V28	1	0.875	0.875	0.875	0.958333	0.541667	0.958333333	0.958333333	0.958333333
V29	1	0.916667	0.75	0.916667	0.958333	0.541667	0.916666667	0.916666667	0.916666667
V30	0.916667	0.875	0.916667	0.875	0.916667	0.458333	1	0.958333333	0.958333333
V31	0.916667	0.916667	1	0.916667	0.791667	0.375	0.958333333	0.916666667	0.916666667
V32	0.916667	0.833333	0.958333	0.875	0.958333	0.541667	0.958333333	0.958333333	0.916666667
V33	0.916667	0.916667	0.916667	0.958333	0.958333	0.708333	0.916666667	0.958333333	1
V34	0.958333	0.958333	0.958333	0.916667	0.916667	0.541667	0.875	1	0.875

Tablo 4.2 (devam): Seçilmiş algoritmaların doğruluk değerleri.

V35	0.916667	0.875	0.875	0.875	1	0.708333	0.916666667	0.958333333	0.875
V36	0.916667	0.958333	0.916667	0.875	0.916667	0.666667	0.875	0.916666667	1
V37	0.958333	1	0.958333	1	0.958333	0.458333	0.958333333	0.833333333	0.958333333
V38	0.958333	0.916667	1	0.916667	0.958333	0.583333	0.916666667	0.875	0.875
V39	0.958333	0.958333	1	0.875	0.833333	0.5	0.916666667	0.916666667	0.958333333
V40	0.958333	0.958333	0.875	0.875	0.916667	0.541667	1	0.916666667	0.916666667
V41	1	0.958333	0.916667	0.833333	0.916667	0.666667	1	1	0.958333333
V42	0.916667	0.916667	0.958333	0.916667	1	0.375	0.916666667	0.958333333	0.958333333
V43	0.958333	0.916667	0.833333	0.333333	0.916667	0.666667	0.958333333	0.916666667	0.958333333
V44	0.833333	0.958333	0.916667	0.958333	0.958333	0.375	0.916666667	0.916666667	0.958333333
V45	1	0.958333	0.958333	0.875	0.875	0.458333	0.916666667	0.958333333	0.875
V46	0.916667	0.875	0.958333	0.958333	0.958333	0.583333	0.833333333	0.916666667	0.958333333
V47	0.958333	0.916667	1	0.958333	1	0.708333	1	0.916666667	0.958333333
V48	0.916667	1	0.916667	0.833333	0.958333	0.708333	0.958333333	0.916666667	1
V49	0.833333	1	1	0.875	0.916667	0.5	1	0.833333333	0.916666667
V50	0.958333	0.833333	0.875	0.791667	0.958333	0.416667	1	1	0.958333333
Varyans	0.002495	0.002066	0.003333	0.011905	0.002143	0.013977	0.002746599	0.002047194	0.001267715

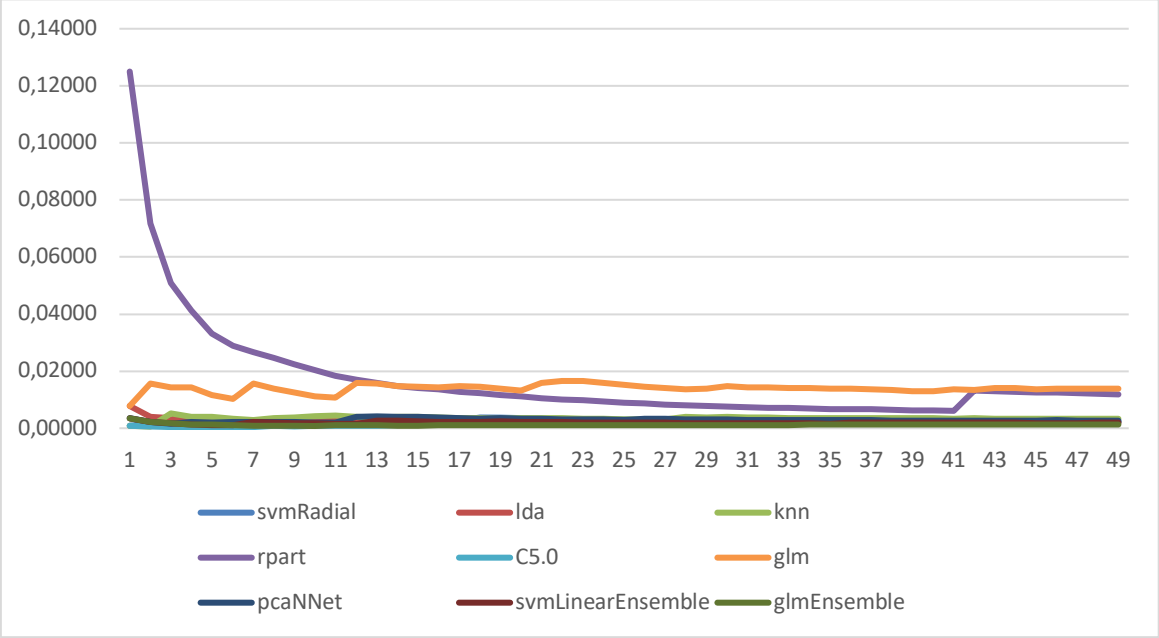
Tablo 4.3, Şekil 4.4 ve Şekil 4.5 eğrisinden her modelin varyans aşamasının analizine göre başlangıçta Radyal SVM ve K en Yakın Komşu modellerinin en küçük varyansı tutarken 15'inci seferden itibaren GLM'ye dayalı topluluk modeli tarafından aşıldı. GLM topluluk modeli 0,001267715'lik bir varyansa ulaşana kadar diğer tüm modellere karşı daha küçük bir varyansı gösterdi.

Tablo 4.3: Modellerin varyansının tablosu.

	svmRadial	lda	knn	rpart	C5.0	glm	pcaNNet	svmLinear Ensemble	glmEnsemble
V1	0.00087	0.00781	0.00087	0.12500	0.00087	0.00781	0.00347	0.00347	0.00347
V2	0.00174	0.00405	0.00058	0.07176	0.00058	0.01563	0.00231	0.00231	0.00231
V3	0.00116	0.00347	0.00521	0.05093	0.00043	0.01432	0.00159	0.00174	0.00174
V4	0.00226	0.00260	0.00399	0.04132	0.00035	0.01441	0.00226	0.00139	0.00139
V5	0.00255	0.00208	0.00394	0.03310	0.00046	0.01157	0.00191	0.00116	0.00122
V6	0.00215	0.00174	0.00331	0.02894	0.00050	0.01033	0.00215	0.00107	0.00107
V7	0.00248	0.00171	0.00285	0.02675	0.00047	0.01559	0.00220	0.00220	0.00096
V8	0.00222	0.00164	0.00347	0.02474	0.00077	0.01389	0.00217	0.00193	0.00087
V9	0.00233	0.00172	0.00378	0.02230	0.00069	0.01252	0.00210	0.00185	0.00085
V10	0.00218	0.00287	0.00417	0.02030	0.00085	0.01127	0.00202	0.00167	0.00079
V11	0.00200	0.00263	0.00437	0.01845	0.00089	0.01077	0.00204	0.00162	0.00110
V12	0.00189	0.00247	0.00401	0.01703	0.00082	0.01596	0.00410	0.00149	0.00105
V13	0.00207	0.00230	0.00386	0.01581	0.00085	0.01568	0.00417	0.00267	0.00099
V14	0.00265	0.00215	0.00408	0.01485	0.00116	0.01485	0.00408	0.00260	0.00094
V15	0.00277	0.00205	0.00391	0.01400	0.00138	0.01455	0.00399	0.00242	0.00090
V16	0.00285	0.00220	0.00374	0.01356	0.00133	0.01435	0.00380	0.00227	0.00103
V17	0.00271	0.00209	0.00365	0.01284	0.00146	0.01467	0.00364	0.00214	0.00098
V18	0.00368	0.00200	0.00345	0.01220	0.00155	0.01453	0.00344	0.00212	0.00097
V19	0.00373	0.00192	0.00338	0.01162	0.00210	0.01379	0.00352	0.00236	0.00117
V20	0.00357	0.00184	0.00352	0.01109	0.00219	0.01330	0.00336	0.00224	0.00114
V21	0.00343	0.00193	0.00363	0.01057	0.00210	0.01598	0.00334	0.00213	0.00110
V22	0.00329	0.00187	0.00347	0.01014	0.00219	0.01657	0.00319	0.00210	0.00106
V23	0.00317	0.00193	0.00332	0.00973	0.00211	0.01650	0.00309	0.00206	0.00103
V24	0.00306	0.00186	0.00322	0.00933	0.00204	0.01582	0.00308	0.00203	0.00101
V25	0.00308	0.00182	0.00313	0.00897	0.00197	0.01518	0.00296	0.00195	0.00098
V26	0.00298	0.00176	0.00305	0.00866	0.00190	0.01465	0.00334	0.00196	0.00107

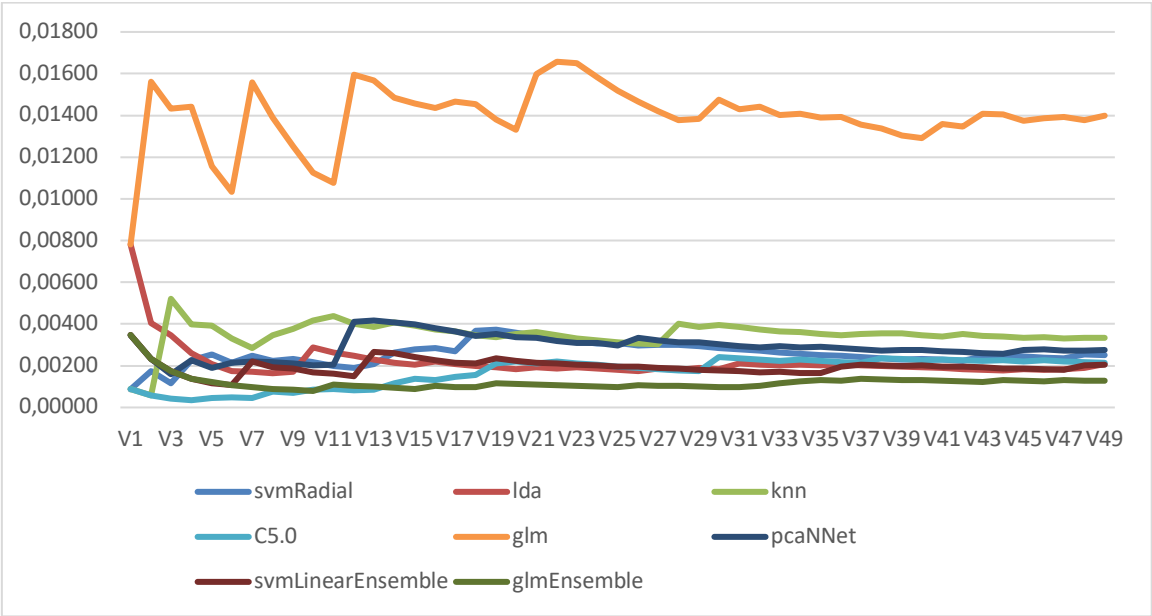
Tablo 4.3 (devam): Modellerin varyansının tablosu.

V27	0.00300	0.00186	0.00304	0.00834	0.00184	0.01419	0.00322	0.00189	0.00104
V28	0.00301	0.00181	0.00401	0.00807	0.00179	0.01376	0.00313	0.00187	0.00103
V29	0.00293	0.00189	0.00387	0.00780	0.00175	0.01383	0.00313	0.00181	0.00100
V30	0.00286	0.00184	0.00395	0.00757	0.00242	0.01475	0.00303	0.00178	0.00099
V31	0.00278	0.00211	0.00386	0.00733	0.00235	0.01431	0.00294	0.00173	0.00098
V32	0.00272	0.00206	0.00374	0.00725	0.00229	0.01441	0.00287	0.00168	0.00105
V33	0.00264	0.00201	0.00366	0.00705	0.00224	0.01401	0.00293	0.00170	0.00116
V34	0.00258	0.00205	0.00362	0.00685	0.00228	0.01409	0.00287	0.00165	0.00125
V35	0.00252	0.00201	0.00352	0.00666	0.00223	0.01389	0.00291	0.00164	0.00132
V36	0.00246	0.00208	0.00346	0.00680	0.00218	0.01393	0.00284	0.00197	0.00129
V37	0.00241	0.00203	0.00351	0.00663	0.00213	0.01355	0.00278	0.00206	0.00137
V38	0.00235	0.00199	0.00356	0.00647	0.00236	0.01336	0.00272	0.00202	0.00134
V39	0.00230	0.00195	0.00354	0.00631	0.00231	0.01305	0.00274	0.00199	0.00133
V40	0.00232	0.00192	0.00345	0.00624	0.00227	0.01292	0.00276	0.00202	0.00130
V41	0.00228	0.00188	0.00339	0.00610	0.00231	0.01360	0.00271	0.00197	0.00128
V42	0.00223	0.00184	0.00352	0.01321	0.00226	0.01347	0.00265	0.00195	0.00125
V43	0.00245	0.00181	0.00344	0.01305	0.00222	0.01408	0.00260	0.00192	0.00123
V44	0.00248	0.00178	0.00338	0.01275	0.00226	0.01405	0.00256	0.00188	0.00130
V45	0.00244	0.00183	0.00333	0.01260	0.00222	0.01374	0.00276	0.00186	0.00128
V46	0.00239	0.00179	0.00337	0.01245	0.00225	0.01385	0.00277	0.00183	0.00126
V47	0.00235	0.00184	0.00331	0.01224	0.00222	0.01394	0.00272	0.00181	0.00130
V48	0.00254	0.00189	0.00334	0.01198	0.00218	0.01376	0.00274	0.00202	0.00129
V49	0.00250	0.00207	0.00333	0.01191	0.00214	0.01398	0.00275	0.00205	0.00127



Şekil 4.4: Modellerin varyansının dağılımı.

Şekil 4.4, Şekil 4.5 ve Tablo 4.3'te bahsedilen 49 test sırasında varyansın gelişimini göstermektedir. Şekil 4.4'deki diğer algoritmaların varyans ilerlemesi daha açık göstermek için rpart algoritmasız Şekil 4.5'te gösterilmiştir.



Şekil 4.5: Rpart modelinin varyans dağılımı.

5 TARTIŞMA VE SONUÇ

Makine öğrenimi, analitik model oluşturmayı otomatikleştirmeyi amaçlayan bir veri analizi yöntemidir. Bu tez kapsamında topluluk öğrenme yöntemi kullanarak makine öğrenme modellerinin varyansının azaltılması amaçlanmaktadır.

Mikrodizi veri setleri genellikle az örnek, çok fazla öznitelikten oluşmaktadır. Bu tez kapsamında kullanılan veri seti, NCBI GEO veri tabanından çekilmiş 120 örnek ve 54675 öznitelikten oluşmaktadır. Chen ve diğ, (2009) çalışmalarında az örneklem ve fazla özniteliği olan veri setlerine topluluk öğrenme uygulayıp daha iyi sonuca vardıkları görülmektedir. Modeli oluşturmak için kullanılan veri kümesi en önemli öznitelikleri içeriyorsa, topluluk yöntemlerinin yüksek varyanstan kaçınmaya yardımcı olduğu gösterilmiştir (Fishel ve diğ., 2007). Bu tez çalışmasında da örnek sayısı az, öznitelik sayısı fazla olan veri setine topluluk öğrenmesi uygulandığında; Chen ve diğ. (2009) benzer şekilde performansın arttığı ve Fishel ve diğ. (2007) benzer şekilde varyansın azaldığı gözlemlenmiştir.

Makine öğrenme modeli oluşturmak için hazır hale getirildikten sonra Doğrusal ve Radyal Destek Vektör Makinesi, Doğrusal Ayırıcı Analizi, k-En Yakın Komşu, CART, Saf Bayes, Çok Katmanlı Algılayıcı, C5.0, Kısmi En Küçük Kareler, Öznitelik Çıkarımlı Sinir Ağları, Rasgele Orman, Genelleştirilmiş doğrusal model (*GLM*) olmak üzere toplam 12 algoritma ile korelasyon analizi yapıldı ve ondan sonra 4 algoritma kalan algoritmalarla arasında yüksek korelasyon olduğu için deneyden kaldırıldı. İstifleme topluluğu öğrenme kavramı, alt sınıflandırıcıları tarafından sağlanan sonuçları kullanarak nihai tahmini yapacak bir algoritma kullanmaktır; daha optimize edilmiş bir modeli elde etmek için tezimizde korelasyon analizi yapılmıştır, bu nedenle, istifleme algoritması tarafından yapılan tahmin doğruluğunu optimize etmeyi amaçlayan istifleme modelimizde ilişkili algoritmaları kaldırılmıştır. Model korelasyon analizi yerine (ya da korelasyon ile birlikte) kullanılabilecek yöntemlerden bir tanesi model parametreleştirmedir; Huang ve Boutros (2016) çalışmalarında veri setlerine korelasyon analizi yerine modelleri parametreleştirerek oluşturdukları modeller performanslı olduğunu göstermiştir.

Korelasyon analizinden sonra makine öğrenme modelleri oluşturulmuştur. Modelleme kısmı 2 bölüme ayrılmıştır: Kalan 7 algoritmaya dayalı 7 basit modelin oluşturulması ve ardından 2 farklı istifleme topluluk modelin (yani doğrusal *SVM* tabanlı ve *GLM* tabanlı topluluk öğrenimi)

oluşturulması. *Holdout* yöntemi kullanılarak, her model veri setinin %80'i ile eğitildi ve test için %20'si kullanıldı. Oluşturulan her model (basit veya topluluk) 50 kez çalıştırıldı ve doğruluk değeri her seferinde kaydedildi (Tablo 4.2) ve varyans yeniden hesaplanıp kaydedildi (Tablo 4.3). Elde edilen varyans tablosundan (Tablo 4.3) 14'üncü kereden sonuna kadar 0,00127 değerini aldığında, GLM tabanlı *Stacking* topluluk modelinin ardından Doğrusal *SVM* tabanlı *Stacking* topluluk modelinin basit algoritma tabanlı modellere göre daha düşük varyansı puanladığı görülmektedir. Literatür araştırmaları sırasında, aynı konuyu ele alan çok fazla çalışmaya rastlanmamıştır fakat son zamanlarda yapılan çalışmalar, topluluk öğrenme yöntemlerinin gerçekten güçlü olduğunu göstermiştir. Torbalama, artırma ve istifleme olan en popüler topluluk öğrenme türü birçok çalışmada hem varyans ve önyargı *(bias)* azaltmış hem de doğruluk artmıştır (Chen ve diğ., 2009; Guile ve Wang, 2010; Zhang ve diğ., 2010; Khoshgoftaar ve diğ., 2013). Ayrıca, karar ağaçlarına dayalı bir topluluk yöntemi olarak kabul edilen rastgele orman, çoğunlukla iyi hiperparametrize edildiğinde daha verimli olduğu gösterilmiştir (Xiao ve diğ., 2014; Huang ve Boutros, 2016; Wang ve diğ., 2020). Bu tez kapsamında Rastgele algoritma modeli denenmiştir ve tek algoritma tabanlı modellere karşı düşük bir varyans göstermiştir. Bir *NSCLC* görüntü sınıflandırma deneyi sırasında, rastgele orman ve AdaBoost topluluk öğrenme modelleri, Çoklu destek vektörü makine özyinelemeli öznitelik çıkarımları, L1 cezalı lojistik regresyon, Saf Bayes, seyrek kodlama uzamsal piramit eşleştirme (*sparse coding spatial pyramid matching*), yerellik kısıtlamalı doğrusal kodlama (*locality constrained linear coding*) ve en yakın sınıf ortalaması sınıflandırıcına (*nearest class mean classifier*) karşı en iyi sınıflandırma sonuçlarını göstermiştir (Wang ve diğ., 2014) . Rastgele orman algoritma, bir karar ağaç topluluk modeli olduğu için tez çalışmalarımızda çıkarılmıştır.

Bu çalışma istifleme topluluk öğrenme yöntemine dayanmaktadır ve tek algoritmaya dayalı algoritmalardan daha düşük varyansa sahip oldukları görünmüştür. Elde edilen bu sonucun dışında topluluk model türü olan istifleme modellerinin en bilinen öznitelikleri yüksek performans elde etmeyi sağlamaktadır. Sonuç olarak örnek sayısının az öznitelik sayısının fazla olduğu mikrodizi veri setlerinde istifleme topluluk öğrenme yönteminin kullanılması önerilmektedir. Bu çalışmada kullanılan yöntemin veri bilimi veya biyoenformatik alanlarında çalışan araştırmacıların ilgisini çekmesi umulmaktadır. Bu tez çerçevesinde yapılan çalışmalar farklı şekillerde genişletilebilir. Biyoenformatik açısından; bu tezde kullanılan yöntemi mikrodizi veri kümeleri yerine diğer veri kümeleri (omik veya Yeni Nesil Dizileme veri

kümeleri gibi) ile de denemek, kullanılan modellerin varyansını azaltmaya yardımcı olabilir. Veri madenciliği (veya ilgili alanlar) açısından ise, bu tezde kullanılan yöntemi farklı alanlardaki veri kümelerine uygulanarak ya da daha farklı hibrit modeller uygulayarak, düşük varyanslı modellerin ortaya çıkmasını sağlayabilir ve aynı zamanda makine öğrenme alanında topluluk modellerinin kullanımının yaygınlaşmasına katkıda bulunabilir.

6 KAYNAKLAR

Abdulla, M. and Khasawneh, M.T. (2020) 'G-Forest: An ensemble method for cost-sensitive feature selection in gene expression microarrays', *Artificial Intelligence in Medicine*, 108, p. 101941.

Abeel, T. *et al.* (2010) 'Robust biomarker identification for cancer diagnosis with ensemble feature selection methods', *Bioinformatics*, 26(3), pp. 392–398. doi:10.1093/bioinformatics/btp630.

Al-Jarrah, O.Y. *et al.* (2014) 'Machine-Learning-Based Feature Selection Techniques for Large-Scale Network Intrusion Detection', in *2014 IEEE 34th International Conference on Distributed Computing Systems Workshops (ICDCSW)*. *2014 IEEE 34th International Conference on Distributed Computing Systems Workshops (ICDCSW)*, pp. 177–181. doi:10.1109/ICDCSW.2014.14.

Angenendt, P. (2005) 'Progress in protein and antibody microarray technology', *Drug Discovery Today*, 10(7), pp. 503–511. doi:10.1016/S1359-6446(05)03392-1.

'Artificial neural network' (2021) *Wikipedia*. Available at: https://en.wikipedia.org/w/index.php?title=Artificial_neural_network&oldid=1026722813 (Accessed: 15 June 2021).

Ashfaq, R.A.R. *et al.* (2017) 'Fuzziness based semi-supervised learning approach for intrusion detection system', *Information Sciences*, 378, pp. 484–497. doi:10.1016/j.ins.2016.04.019.

Azimi-Pour, M., Eskandari-Naddaf, H. and Pakzad, A. (2020) 'Linear and non-linear SVM prediction for fresh properties and compressive strength of high volume fly ash self-compacting concrete', *Construction and Building Materials*, 230, p. 117021. doi:10.1016/j.conbuildmat.2019.117021.

Babichev, S.A. *et al.* (2016) 'Computational analysis of microarray gene expression profiles of lung cancer', *Biopolymers and cell*, (32, № 1), pp. 70–79.

Bayes, T. (1763) 'LII. An essay towards solving a problem in the doctrine of chances. By the late Rev. Mr. Bayes, FRS communicated by Mr. Price, in a letter to John Canton, AMFR S', *Philosophical transactions of the Royal Society of London*, (53), pp. 370–418.

Becker, R.A., Chambers, J.M. and Wilks, A.R. (1988) *The New s Language: A Programming Environment for Data Analysis and Graphics*. Pacific Grove, Calif: Chapman & Hall.

Bolón-Canedo, V. and Alonso-Betanzos, A. (2019) 'Ensembles for feature selection: A review and future trends', *Information Fusion*, 52, pp. 1–12. doi:10.1016/j.inffus.2018.11.008.

Boser, B.E., Guyon, I.M. and Vapnik, V.N. (1992) 'A training algorithm for optimal margin classifiers', in *Proceedings of the fifth annual workshop on Computational learning theory*. New York, NY, USA: Association for Computing Machinery (COLT '92), pp. 144–152. doi:10.1145/130385.130401.

Breiman, L. *et al.* (1984) *Classification and regression trees*. CRC press.

Breiman, L. (1999) 'Random forests', *UC Berkeley TR567* [Preprint]. Available at: http://machinelearning202.pbworks.com/w/file/fetch/60606349/breiman_randomforests.pdf.

Brownlee, J. (2018) 'A Gentle Introduction to k-fold Cross-Validation', *Machine Learning Mastery*, 22 May. Available at: https://machinelearningmastery.com/k-fold-cross-validation/ (Accessed: 3 June 2021).

Brownlee, J. (2019) 'A Gentle Introduction to Model Selection for Machine Learning', *Machine Learning Mastery*, 1 December. Available at: https://machinelearningmastery.com/a-gentle-introduction-to-model-selection-for-machine-learning/ (Accessed: 2 June 2021).

Bühlmann, P. (2012) 'Bagging, Boosting and Ensemble Methods', in Gentle, J.E., Härdle, W.K., and Mori, Y. (eds) *Handbook of Computational Statistics: Concepts and Methods*. Berlin, Heidelberg: Springer (Springer Handbooks of Computational Statistics), pp. 985–1022. doi:10.1007/978-3-642-21551-3_33.

Bujlow, T., Riaz, T. and Pedersen, J.M. (2012) 'A method for classification of network traffic based on C5.0 Machine Learning Algorithm', in *2012 International Conference on Computing, Networking and Communications (ICNC)*. *2012 International Conference on Computing, Networking and Communications (ICNC)*, pp. 237–241. doi:10.1109/ICCNC.2012.6167418.

Calza, S. and Pawitan, Y. (2010) 'Normalization of Gene-Expression Microarray Data', in Fenyö, D. (ed.) *Computational Biology*. Totowa, NJ: Humana Press (Methods in Molecular Biology), pp. 37–52. doi:10.1007/978-1-60761-842-3_3.

Canbek, G. *et al.* (2017) 'Binary classification performance measures/metrics: A comprehensive visualized roadmap to gain new insights', in *2017 International Conference on Computer Science and Engineering (UBMK)*. *2017 International Conference on Computer Science and Engineering (UBMK)*, pp. 821–826. doi:10.1109/UBMK.2017.8093539.

Chen, P. *et al.* (2020) 'Robustness of Accuracy Metric and its Inspirations in Learning with Noisy Labels', *arXiv:2012.04193 [cs]* [Preprint]. Available at: http://arxiv.org/abs/2012.04193 (Accessed: 2 May 2021).

Chen, W. *et al.* (2009) 'Gene Expression Data Classification Using Artificial Neural Network Ensembles Based on Samples Filtering', in *2009 International Conference on Artificial Intelligence and Computational Intelligence*. *2009 International Conference on Artificial Intelligence and Computational Intelligence*, pp. 626–628. doi:10.1109/AICI.2009.441.

Chollet, F. (2018) *Deep learning with Python*. Manning New York.

Cunningham, P., Cord, M. and Delany, S.J. (2008) 'Supervised Learning', in Cord, M. and Cunningham, P. (eds) *Machine Learning Techniques for Multimedia: Case Studies on Organization and Retrieval*. Berlin, Heidelberg: Springer (Cognitive Technologies), pp. 21–49. doi:10.1007/978-3-540-75171-7_2.

Davis, S. and Meltzer, P.S. (2007) 'GEOquery: a bridge between the Gene Expression Omnibus (GEO) and BioConductor', *Bioinformatics (Oxford, England)*, 23(14), pp. 1846–1847. doi:10.1093/bioinformatics/btm254.

Deane-Mayer, Z.A. and Knowles, J.E. (2019) *caretEnsemble: Ensembles of Caret Models*. Available at: https://CRAN.R-project.org/package=caretEnsemble (Accessed: 27 March 2021).

Dey, A. (2016) 'Machine learning algorithms: a review', *International Journal of Computer Science and Information Technologies*, 7(3), pp. 1174–1179.

Dietterich, T.G. and Kong, E.B. (1995) *Machine learning bias, statistical bias, and statistical variance of decision tree algorithms*. Citeseer.

Dobson, A.J. and Barnett, A.G. (2018) *An Introduction to Generalized Linear Models*. CRC Press.

Dubuis, A. *et al.* (2011) 'Predicting spatial patterns of plant species richness: a comparison of direct macroecological and species stacking modelling approaches', *Diversity and Distributions*, 17(6), pp. 1122–1131. doi:https://doi.org/10.1111/j.1472-4642.2011.00792.x.

Dzierżak, R. (2019) 'Comparison of the influence of standardization and normalization of data on the effectiveness of spongy tissue texture classification', *Informatyka, Automatyka, Pomiary w Gospodarce i Ochronie Środowiska*, T. 9, nr 3. doi:10.35784/IAPGOS.62.

El Naqa, I. and Murphy, M.J. (2015) 'What Is Machine Learning?', in El Naqa, I., Li, R., and Murphy, M.J. (eds) *Machine Learning in Radiation Oncology: Theory and Applications*. Cham: Springer International Publishing, pp. 3–11. doi:10.1007/978-3-319-18305-3_1.

El-Bendary, N. *et al.* (2015) 'Using machine learning techniques for evaluating tomato ripeness', *Expert Systems with Applications*, 42(4), pp. 1892–1905. doi:10.1016/j.eswa.2014.09.057.

Eldardiry, H., Neville, J. and Rossi, R.A. (2020) 'Ensemble Learning for Relational Data.', *Journal of Machine Learning Research*, 21(49), pp. 1–37.

Faraggi, D. and Simon, R. (1995) 'A neural network model for survival data', *Statistics in Medicine*, 14(1), pp. 73–82. doi:10.1002/sim.4780140108.

Fishel, I., Kaufman, A. and Ruppin, E. (2007) 'Meta-analysis of gene expression data: a predictor-based approach', *Bioinformatics*, 23(13), pp. 1599–1606. doi:10.1093/bioinformatics/btm149.

Fisher, C.W., Lauria, E.J.M. and Matheus, C.C. (2009) 'An Accuracy Metric: Percentages, Randomness, and Probabilities', *Journal of Data and Information Quality*, 1(3), p. 16:1-16:21. doi:10.1145/1659225.1659229.

Fujita, A. *et al.* (2006) 'Evaluating different methods of microarray data normalization', *BMC Bioinformatics*, 7(1), p. 469. doi:10.1186/1471-2105-7-469.

Gardner, M.W. and Dorling, S.R. (1998) 'Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences', *Atmospheric Environment*, 32(14), pp. 2627–2636. doi:10.1016/S1352-2310(97)00447-0.

'Generalized linear model' (2021) *Wikipedia*. Available at: https://en.wikipedia.org/w/index.php?title=Generalized_linear_model&oldid=1018579650 (Accessed: 15 June 2021).

Gentleman, R. *et al.* (2021) *genefilter: genefilter: methods for filtering genes from high-throughput experiments*. Bioconductor version: Release (3.12). doi:10.18129/B9.bioc.genefilter.

Gola, J. *et al.* (2018) 'Advanced microstructure classification by data mining methods', *Computational Materials Science*, 148, pp. 324–335. doi:10.1016/j.commatsci.2018.03.004.

Gotway, C.A. and Stroup, W.W. (1997) 'A Generalized Linear Model Approach to Spatial Data Analysis and Prediction', *Journal of Agricultural, Biological, and Environmental Statistics*, 2(2), pp. 157–178. doi:10.2307/1400401.

Govindarajan, R. *et al.* (2012) 'Microarray and its applications', *Journal of Pharmacy & Bioallied Sciences*, 4(Suppl 2), pp. S310–S312. doi:10.4103/0975-7406.100283.

Graczyk, M. *et al.* (2010) 'Comparison of Bagging, Boosting and Stacking Ensembles Applied to Real Estate Appraisal', in *Intelligent Information and Database Systems*. *Asian Conference on Intelligent Information and Database Systems*, Springer, Berlin, Heidelberg, pp. 340–350. doi:10.1007/978-3-642-12101-2_35.

Guan, D. *et al.* (2014) 'A Review of Ensemble Learning Based Feature Selection', *IETE Technical Review*, 31(3), pp. 190–198. doi:10.1080/02564602.2014.906859.

Guile, G.R. and Wang, W. (2010) 'Factors affecting boosting ensemble performance on DNA microarray data', in *The 2010 International Joint Conference on Neural Networks (IJCNN)*. *The 2010 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7. doi:10.1109/IJCNN.2010.5596600.

Han, J., Pei, J. and Kamber, M. (2011) *Data Mining: Concepts and Techniques*. Elsevier.

Heller, M.J. (2002) 'DNA Microarray Technology: Devices, Systems, and Applications', *Annual Review of Biomedical Engineering*, 4(1), pp. 129–153. doi:10.1146/annurev.bioeng.4.020702.153438.

Hochreiter, S., Clevert, D.-A. and Obermayer, K. (2006) 'A new summarization method for affymetrix probe level data', *Bioinformatics*, 22(8), pp. 943–949. doi:10.1093/bioinformatics/btl033.

Huang, B.F.F. and Boutros, P.C. (2016) 'The parameter sensitivity of random forests', *BMC Bioinformatics*, 17(1), p. 331. doi:10.1186/s12859-016-1228-x.

Huang, H. *et al.* (2015) 'Maximum F1-Score Discriminative Training Criterion for Automatic Mispronunciation Detection', *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(4), pp. 787–797. doi:10.1109/TASLP.2015.2409733.

Irizarry, R.A., Wu, Z. and Jaffee, H.A. (2006) 'Comparison of Affymetrix GeneChip expression measures', *Bioinformatics*, 22(7), pp. 789–794. doi:10.1093/bioinformatics/btk046.

Jemal, A. *et al.* (2010) 'Cancer statistics, 2010', *CA: a cancer journal for clinicians*, 60(5), pp. 277–300. doi:10.3322/caac.20073.

Jiang, T., Gradus, J.L. and Rosellini, A.J. (2020) 'Supervised Machine Learning: A Brief Primer', *Behavior Therapy*, 51(5), pp. 675–687. doi:10.1016/j.beth.2020.05.002.

Jolliffe, I.T. and Cadima, J. (2016) 'Principal component analysis: a review and recent developments', *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065), p. 20150202. doi:10.1098/rsta.2015.0202.

Kearns, M. *et al.* (1997) 'An Experimental and Theoretical Comparison of Model Selection Methods', *Machine Learning*, 27(1), pp. 7–50. doi:10.1023/A:1007344726582.

Khoshgoftaar, T.M. *et al.* (2013) 'A Review of Ensemble Classification for DNA Microarrays Data', in *2013 IEEE 25th International Conference on Tools with Artificial Intelligence*. *2013 IEEE 25th International Conference on Tools with Artificial Intelligence*, pp. 381–389. doi:10.1109/ICTAI.2013.64.

Kim, K.-J. and Cho, S.-B. (2006) 'Ensemble classifiers based on correlation analysis for DNA microarray classification', *Neurocomputing*, 70(1), pp. 187–199. doi:10.1016/j.neucom.2006.03.002.

Kotsiantis, S.B., Zaharakis, I.D. and Pintelas, P.E. (2006) 'Machine learning: a review of classification and combining techniques', *Artificial Intelligence Review*, 26(3), pp. 159–190. doi:10.1007/s10462-007-9052-3.

Kuhn, M. (2015) 'caret: Classification and Regression Training', *Astrophysics Source Code Library*, p. ascl:1505.003.

Larose, D.T. and Larose, C.D. (2014) *Discovering Knowledge in Data: An Introduction to Data Mining*. John Wiley & Sons.

Li, T., Zhu, S. and Ogihara, M. (2006) 'Using discriminant analysis for multi-class classification: an experimental investigation', *Knowledge and Information Systems*, 10(4), pp. 453–472. doi:10.1007/s10115-006-0013-y.

Lin, E. *et al.* (2016) 'Peptide microarray patterning for controlling and monitoring cell growth', *Acta Biomaterialia*, 34, pp. 53–59. doi:10.1016/j.actbio.2016.01.028.

Liu, G., Yi, Z. and Yang, S. (2007) 'A hierarchical intrusion detection model based on the PCA neural networks', *Neurocomputing*, 70(7), pp. 1561–1568. doi:10.1016/j.neucom.2006.10.146.

Liu, H., Liu, L. and Zhang, H. (2010) 'Ensemble gene selection by grouping for microarray data classification', *Journal of Biomedical Informatics*, 43(1), pp. 81–87. doi:10.1016/j.jbi.2009.08.010.

Lu, T.-P. *et al.* (2010) 'Identification of a novel biomarker, SEMA5A, for non-small cell lung carcinoma in nonsmoking women', *Cancer Epidemiology, Biomarkers & Prevention: A Publication of the American Association for Cancer Research, Cosponsored by the American Society of Preventive Oncology*, 19(10), pp. 2590–2597. doi:10.1158/1055-9965.EPI-10-0332.

Lu, T.-P. *et al.* (2015) 'Identification of regulatory SNPs associated with genetic modifications in lung adenocarcinoma', *BMC research notes*, 8, p. 92. doi:10.1186/s13104-015-1053-8.

Ma, P.C. *et al.* (2005) 'Functional expression and mutations of c-Met and its therapeutic inhibition with SU11274 and small interfering RNA in non–small cell lung cancer', *Cancer research*, 65(4), pp. 1479–1488.

Maimon, O. and Rokach, L. (eds) (2010) *Data Mining and Knowledge Discovery Handbook*. 2nd edn. Springer US. doi:10.1007/978-0-387-09823-4.

Mehta, P. *et al.* (2019) 'A high-bias, low-variance introduction to machine learning for physicists', *Physics reports*, 810, pp. 1–124.

Mendoza, D.P. *et al.* (2020) 'Role of imaging biomarkers in mutation-driven non-small cell lung cancer', *World Journal of Clinical Oncology*, 11(7), pp. 412–427. doi:10.5306/wjco.v11.i7.412.

Mohammed, M. *et al.* (2018) 'Using stacking ensemble for microarray-based cancer classification', in *2018 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)*. IEEE, pp. 1–8.

Molla, M. *et al.* (2004) 'Using Machine Learning to Design and Interpret Gene-Expression Microarrays', *AI Magazine*, 25(1), pp. 23–23. doi:10.1609/aimag.v25i1.1745.

Molnar, C. (no date) *4.3 GLM, GAM and more | Interpretable Machine Learning*. Available at: https://christophm.github.io/interpretable-ml-book/extend-lm.html (Accessed: 15 June 2021).

Nelder, J.A. and Wedderburn, R.W.M. (1972) 'Generalized Linear Models', *Journal of the Royal Statistical Society. Series A (General)*, 135(3), pp. 370–384. doi:10.2307/2344614.

Nian, R., Liu, J. and Huang, B. (2020) 'A review On reinforcement learning: Introduction and applications in industrial process control', *Computers & Chemical Engineering*, 139, p. 106886. doi:10.1016/j.compchemeng.2020.106886.

Opitz, D. and Maclin, R. (1999) 'Popular Ensemble Methods: An Empirical Study', *Journal of Artificial Intelligence Research*, 11, pp. 169–198. doi:10.1613/jair.614.

Özkan, Y. and Erol, Ç.S. (2015) *Biyoenformatik DNA mikrodizi: veri madenciliği*. Papatya Yayıncılık Eğitim.

Panca, V. and Rustam, Z. (2017) 'Application of machine learning on brain cancer multiclass classification', *AIP Conference Proceedings*, 1862(1), p. 030133. doi:10.1063/1.4991237.

Park, T. *et al.* (2003) 'Evaluation of normalization methods for microarray data', *BMC Bioinformatics*, 4(1), p. 33. doi:10.1186/1471-2105-4-33.

Pirooznia, M. *et al.* (2008) 'A comparative study of different machine learning methods on microarray gene expression data', *BMC Genomics*, 9(1), p. S13. doi:10.1186/1471-2164-9-S1-S13.

Quinlan, J.R. (1996) 'Learning decision tree classifiers', *ACM Computing Surveys (CSUR)*, 28(1), pp. 71–72.

Ranaie, M. *et al.* (2018) 'Evaluating the statistical performance of less applied algorithms in classification of worldview-3 imagery data in an urbanized landscape', *Advances in Space Research*, 61(6), pp. 1558–1572. doi:10.1016/j.asr.2018.01.004.

Raschka, S. (2020) 'Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning', *arXiv:1811.12808 [cs, stat]* [Preprint]. Available at: http://arxiv.org/abs/1811.12808 (Accessed: 26 March 2021).

Ringnér, M. (2008) 'What is principal component analysis?', *Nature Biotechnology*, 26(3), pp. 303–304. doi:10.1038/nbt0308-303.

Rocca, J. (2021) *Ensemble methods: bagging, boosting and stacking*, *Medium*. Available at: https://towardsdatascience.com/ensemble-methods-bagging-boosting-and-stacking-c9214a10a205 (Accessed: 22 May 2021).

Russo, G., Zegar, C. and Giordano, A. (2003) 'Advantages and limitations of microarray technology in human cancer', *Oncogene*, 22(42), pp. 6497–6507. doi:10.1038/sj.onc.1206865.

Safiyari, A. and Javidan, R. (2017) 'Predicting lung cancer survivability using ensemble learning methods', in *2017 Intelligent Systems Conference (IntelliSys)*. *2017 Intelligent Systems Conference (IntelliSys)*, pp. 684–688. doi:10.1109/IntelliSys.2017.8324368.

Shannon, C.E. (1948) 'A mathematical theory of communication', *The Bell system technical journal*, 27(3), pp. 379–423.

Sharma, N. (2019) *Importance of Distance Metrics in Machine Learning Modelling*, *Medium*. Available at: https://towardsdatascience.com/importance-of-distance-metrics-in-machine-learning-modelling-e51395ffe60d (Accessed: 1 June 2021).

Sontakke, S., Lohokare, J. and Dani, R. (2017) 'Diagnosis of liver diseases using machine learning', in *2017 International Conference on Emerging Trends Innovation in ICT (ICEI)*. *2017 International Conference on Emerging Trends Innovation in ICT (ICEI)*, pp. 129–133. doi:10.1109/ETIICT.2017.7977023.

Sun, Y. *et al.* (2012) 'Role of insulin-like growth factor-1 signaling pathway in cisplatin-resistant lung cancer cells', *International Journal of Radiation Oncology, Biology, Physics*, 82(3), pp. e563-572. doi:10.1016/j.ijrobp.2011.06.1999.

Tavish Srivastava (2019) 'Evaluation Metrics Machine Learning', *Analytics Vidhya*, 6 August. Available at: https://www.analyticsvidhya.com/blog/2019/08/11-important-model-evaluation-error-metrics/ (Accessed: 3 June 2021).

Templin, M.F. *et al.* (2002) 'Protein microarray technology', *Drug Discovery Today*, 7(15), pp. 815–822. doi:10.1016/S1359-6446(00)01910-2.

Turgut, S., Dağtekin, M. and Ensari, T. (2018) 'Microarray breast cancer data classification using machine learning methods', in *2018 Electric Electronics, Computer Science, Biomedical Engineerings' Meeting (EBBT)*. *2018 Electric Electronics, Computer Science, Biomedical Engineerings' Meeting (EBBT)*, pp. 1–3. doi:10.1109/EBBT.2018.8391468.

ujjwalkarn (2016) 'A Quick Introduction to Neural Networks', *the data science blog*, 9 August. Available at: https://ujjwalkarn.me/2016/08/09/quick-intro-neural-networks/ (Accessed: 15 June 2021).

'URL-1' (2006). Available at: https://www.affymetrix.com/about_affymetrix/outreach/educator/downloads/genechip_teachersguide.pdf.

Vaibhaw, Sarraf, J. and Pattnaik, P.K. (2020) 'Chapter 2 - Brain–computer interfaces and their applications', in Balas, V.E., Solanki, V.K., and Kumar, R. (eds) *An Industrial IoT Approach for Pharmaceutical Industry Growth*. Academic Press, pp. 31–54. doi:10.1016/B978-0-12-821326-1.00002-4.

Vapnik, V.N. (1998) 'Adaptive and learning systems for signal processing communications, and control', *Statistical learning theory* [Preprint].

Veloso de Melo, V. and Banzhaf, W. (2018) 'Automatic feature engineering for regression models with machine learning: An evolutionary computation and statistics hybrid', *Information Sciences*, 430–431, pp. 287–313. doi:10.1016/j.ins.2017.11.041.

Verma, A. and Mehta, S. (2017) 'A comparative study of ensemble learning methods for classification in bioinformatics', in *2017 7th International Conference on Cloud Computing, Data Science Engineering - Confluence*. *2017 7th International Conference on Cloud Computing, Data Science Engineering - Confluence*, pp. 155–158. doi:10.1109/CONFLUENCE.2017.7943141.

Wang, H. *et al.* (2014) 'Novel image markers for non-small cell lung cancer classification and survival prediction', *BMC Bioinformatics*, 15(1), p. 310. doi:10.1186/1471-2105-15-310.

Wang, Q. *et al.* (2020) 'Random Forest with Self-Paced Bootstrap Learning in Lung Cancer Prognosis', *ACM Transactions on Multimedia Computing, Communications, and Applications*, 16(1s), p. 34:1-34:12. doi:10.1145/3345314.

Wang, Yuyan *et al.* (2019) 'Stacking-based ensemble learning of decision trees for interpretable prostate cancer detection', *Applied Soft Computing*, 77, pp. 188–204. doi:10.1016/j.asoc.2019.01.015.

Wikipedia (2021) ‘Data pre-processing’, *Wikipedia*. Available at: https://en.wikipedia.org/w/index.php?title=Data_pre-processing&oldid=1009186498 (Accessed: 23 March 2021).

Wold, H. (1975) ‘11 - Path Models with Latent Variables: The NIPALS Approach**NIPALS = Nonlinear Iterative PArtial Least Squares.’, in Blalock, H.M. et al. (eds) *Quantitative Sociology*. Academic Press (International Perspectives on Mathematical and Statistical Modeling), pp. 307–357. doi:10.1016/B978-0-12-103950-9.50017-4.

Wold, S. *et al.* (1984) ‘The collinearity problem in linear regression. The partial least squares (PLS) approach to generalized inverses’, *SIAM Journal on Scientific and Statistical Computing*, 5(3), pp. 735–743.

Xiao, G. *et al.* (2014) ‘Adaptive Prediction Model in Prospective Molecular Signature–Based Clinical Studies’, *Clinical Cancer Research*, 20(3), pp. 531–539.

Yadav, A. (2018) *SUPPORT VECTOR MACHINES(SVM)*, *Medium*. Available at: https://towardsdatascience.com/support-vector-machines-svm-c9ef22815589 (Accessed: 27 July 2021).

Yadav, S. and Shukla, S. (2016) ‘Analysis of k-Fold Cross-Validation over Hold-Out Validation on Colossal Datasets for Quality Classification’, in *2016 IEEE 6th International Conference on Advanced Computing (IACC)*. *2016 IEEE 6th International Conference on Advanced Computing (IACC)*, pp. 78–83. doi:10.1109/IACC.2016.25.

Yaman, E. and Subasi, A. (2019) ‘Comparison of Bagging and Boosting Ensemble Machine Learning Methods for Automated EMG Signal Classification’, *BioMed Research International*, 2019, p. e9152506. doi:10.1155/2019/9152506.

Ying, X. (2019) ‘An Overview of Overfitting and its Solutions’, *Journal of Physics: Conference Series*, 1168, p. 022022. doi:10.1088/1742-6596/1168/2/022022.

Zenko, B., Todorovski, L. and Dzeroski, S. (2001) ‘A comparison of stacking with meta decision trees to bagging, boosting, and stacking with other methods’, in *Proceedings 2001 IEEE International Conference on Data Mining*. *Proceedings 2001 IEEE International Conference on Data Mining*, pp. 669–670. doi:10.1109/ICDM.2001.989601.

Zhang, C. *et al.* (2018) ‘A Study on Overfitting in Deep Reinforcement Learning’, *arXiv:1804.06893 [cs, stat]* [Preprint]. Available at: http://arxiv.org/abs/1804.06893 (Accessed: 16 May 2021).

Zhang, H. and Su, J. (2004) ‘Naive bayesian classifiers for ranking’, in *European conference on machine learning*. Springer, pp. 501–512.

Zhang, Z. *et al.* (2010) ‘A Robust Ensemble Classification Method Analysis’, in Arabnia, H.R. (ed.) *Advances in Computational Biology*. New York, NY: Springer (Advances in Experimental Medicine and Biology), pp. 149–155. doi:10.1007/978-1-4419-5913-3_17.

Zhou, Z.-H. (2009) ‘Ensemble learning.’, *Encyclopedia of biometrics*, 1, pp. 270–273.

Zou, J., Han, Y. and So, S.-S. (2009) ‘Overview of Artificial Neural Networks’, in Livingstone, D.J. (ed.) *Artificial Neural Networks: Methods and Applications*. Totowa, NJ: Humana Press (Methods in Molecular Biology™), pp. 14–22. doi:10.1007/978-1-60327-101-1_2.

EKLER

Ek 1: Veri seti çekme ve mikrodizi önişlemi uygulama R kodları

```
# Analize başlamadan önce çalışma alanındaki her nesneyi temizlemek
# ve kütüphaneler yüklemek
rm(list=ls(all=TRUE))
library(GEOquery)
library(caret)
library(caretEnsemble)

# veri seti NCBİ'den indirmek
okunan <- getGEO("GSE19804",GSEMatrix=TRUE, getGPL=FALSE)
eset <- okunan[[1]]
annotation(eset) <- "hgu133plus2.db"

#Veri seti filtrelemek
library(genefilter)
filtrelenmis <- nsFilter(eset, var.cutoff=0.95,
var.filter=TRUE)$eset
dim(eset)
dim(filtrelenmis)

# Veri siniflari etiketleme
colnames(pData(filtrelenmis))
durum <- factor(pData(filtrelenmis)$characteristics_ch1)

levels(durum)
levels(durum)[levels(durum)=="tissue: paired normal adjacent"] <-
"Lung_cancer"
levels(durum)[levels(durum)=="tissue: lung cancer"] <- "control"
table(durum)


# Veri seti ön işlemi bitirdikten sonra .csv dosya olarak kaydetmek
filtrelenmisVeri = as.matrix(t(exprs(filtrelenmis)))
sonVeri <- data.frame(scale(filtrelenmisVeri, center = TRUE, scale =
TRUE))
dim(sonVeri)

write.table(DataFrame(durum, sonVeri),

file="C:/Users/Adnaane/Dropbox/thesis/tez/tez_resources/tez_kod/son_
veri.csv",sep = ',', row.names = F)
```

Ek 2: Basit modellerin oluşturması ve değerlendirmesi R kodları

```
# Analize başlamadan önce çalışma alanındaki her nesneyi temizlemek
# ve kütüphaneler yüklemek
Library('caret')

# Kaydedilmiş .csv dosyayı yüklemek ve veri madenciliği süreci için
hazır hale getirmek

data_set <-
read.csv("C:/Users/Adnaane/Dropbox/thesis/tez/tez_resources/tez_kod/
son_veri.csv")

durum <-as.factor(data_set$durum)

sonVeri <- data_set[, -which(names(data_set) == "durum")]

# kullanılan algoritma listesi

algorithmList <- c('svmLinear', 'svmRadial', 'lda', 'knn', 'rpart',
'naive_bayes', 'mlp', 'C5.0', 'pls', 'rf', 'glm', 'pcaNNet')

acc_frame <- DataFrame()

var_frame <- DataFrame()

# tek model algoritmaları oluşturup kestirim yapıp sonuçları kaydetmek

for(algorithm in algorithmList){

  str_acc = c();

  for(sefer in 1:50){

    print('==============================')

    print(sefer)

    model_ <- NULL

    sinir <- floor(.8*length(durum))

    ind <- sample.int(n=length(durum),size=sinir,replace=F)

    veriEgitim <- sonVeri[ind,]

    sinifEgitim <- durum[ind]

    veriTest  <- sonVeri[-ind,]

    sinifTest <- durum[-ind]

    model_ <- train(y=sinifEgitim, x=veriEgitim, method=algorithm)

    predicted <- predict(model_,veriTest)

    confMatrix <- table(predicted,sinifTest)
```

```
    accuracy <- sum(diag(confMatrix))/sum(confMatrix)
    str_acc = c(str_acc , accuracy)
    print(str_acc)
    acc_frame[algorithm, sefer] <- accuracy
  }
  var_frame[algorithm, 'variance'] <- var(str_acc)
}
#Elde edilen doğruluk tablosu kaydetmek
write.table(acc_frame,
file="C:/Users/Adnaane/Dropbox/thesis/tez/tez_resources/tez_kod/_veri.csv",sep = ',', row.names = T)
#Varyansı hesaplamak
var_frame <- DataFrame()
for(algorithm in algorithmList){
  str_acc = c()
  for(sefer in 1:50){
   str_acc<-c(str_acc, acc_frame[algorithm, sefer])
   var_frame[algorithm, sefer]<- var(str_acc)
  }
}
#Elde edilen varyans tablosu kaydetmek
write.table(var_frame,
file="C:/Users/Adnaane/Dropbox/thesis/tez/tez_resources/tez_kod/_var.csv",sep = ',', row.names = T)
```

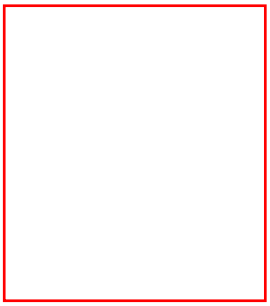
Ek 3: Ensemble modellerin oluşturması ve değerlendirmesi R kodları

```
# Topluluk modelleri oluşturmak için gereken kütüphaneleri yüklemek
library('caretEnsemble')

library('caret')

# Kaydedilmiş on işlemiş .csv dosyayı yüklemek ve veri madenciliği
süreci için hazır hale getirmek

data_set                                                          <-
read.csv("C:/Users/Adnaane/Dropbox/thesis/tez/tez_resources/tez_kod/
son_veri.csv")

# Veri seti modelleme aşaması için hazir hale getirmek

durum <-as.factor(data_set$durum)

sonVeri <- data_set[, -which(names(data_set) == "durum")]

dim(data_set)

dim(sonVeri)

table(durum)

# zayif model kontrol ve stakModel kontrolu

control <- trainControl(savePredictions="final", classProbs=TRUE)

# Staking modeli için birinci aşamada kullanılan model listesi

algorithmList  <-  c('svmRadial',  'lda',  'knn',  'rpart',  'C5.0',
'pcaNNet', 'glm')

acc_frame <- data.frame()

var_frame <- data.frame()

str_acc <- c()

for(i in 1:50){

  sinir <- floor(.8*length(durum))

  ind <- sample.int(n=length(durum),size=sinir,replace=F)

  sinifEgitim  <- durum[ind]

  veriEgitim <- cbind(sinifEgitim, sonVeri[ind,])

  veriTest  <- sonVeri[-ind,]

  sinifTest <- durum[-ind]

  models <- caretList(sinifEgitim~.,data=veriEgitim,

                      trControl=control,
```

```
                        methodList=algorithmList)
  stack.glm <- caretStack(models, method="glm", metric="Accuracy")
  predicted <- predict(stack.glm,veriTest)
  confMatrix <- table(predicted,sinifTest)
  accuracy <- sum(diag(confMatrix))/sum(confMatrix)
  print('============================')
  str_acc <- c(str_acc , accuracy)
  var_frame['glm_ensemble', i] <- accuracy
  var_frame['glm_ensemble_var', i] <- var(str_acc)
  print('============================')
}
# Elde edilen varyans tabloyu .csv olarak kaydetmek
write.table(var_frame,
file="C:/Users/Adnaane/Dropbox/thesis/tez/tez_resources/tez_kod/glm_
var_full.csv",sep = ',', row.names = T)
```

ÖZGEÇMİŞ

Kişisel Bilgiler	
Adı Soyadı	
Doğum Yeri	
Doğum Tarihi	
Uyruğu	
Telefon	
E-Posta Adresi	
Web Adresi	



Eğitim Bilgileri	
Lisans	
Üniversite	IAI-TOGO
Fakülte	Genie Logiciel (Bilgisayar Bilimleri)
Bölümü	Genie Logiciel (Bilgisayar Bilimleri)
Mezuniyet Yılı	16.11.2016

Yüksek Lisans	
Üniversite	Istanbul Üniversitesi
Enstitü Adı	Fen Bilimleri Enstitüsü
Anabilim Dalı	Enformatik
Programı	Enformatik

Makale ve Bildiriler

Makale

[1] Bawa, T. A. , Özkan, Y., & Erol, Ç., (2021). Reanalysis of Non-Small-Cell Lung Cancer Microarray Gene Expression Data. Proceedings, vol.74, no.1, 22.

Bildiri

[2] Bawa, T. A. , Özkan, Y., & Erol, Ç., (2020). Reanalysis of Non-Small Cell Lung Cancer Microarray Gene Expression Data . 7th International Management Information Systems Conference (En iyi 3. Bildiri seçilmiştir).