T.C.

BİLECİK ŞEYH EDEBALİ ÜNİVERSİTESİ

MÜHENDİSLİK FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

MAKİNE ÖĞRENME YÖNTEMLERİNİ KULLANARAK MEME KANSERİ TAHMİNİ

ESRA ASLAN

PROJE-II ÇALIŞMASI

PROJE-II DANIŞMANI : Öğr. Gör. Yusuf Muştu

BİLECİK

23 Mayıs 2024

T.C.

BİLECİK ŞEYH EDEBALİ ÜNİVERSİTESİ

MÜHENDİSLİK FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

MAKİNE ÖĞRENME YÖNTEMLERİNİ KULLANARAK MEME KANSERİ TAHMİNİ

ESRA ASLAN

PROJE-II ÇALIŞMASI

PROJE-II DANIŞMANI : Öğr. Gör. Yusuf Muştu

BİLECİK

23 Mayıs 2024

ÖZET

Projenin Amacı

Bu projenin temel amacı, meme kanserini makine öğrenmesi yöntemleriyle tahmin etmektir. Meme kanseri, kadınlar arasında ölümlerin önde gelen nedenlerinden biridir ve erken teşhis edilmesi önemlidir. Bu projenin amacı, birçok farklı hastadan elde edilen rutin kan tahlili verilerinin hızlı bir şekilde işlenip sınıflandırılmasıyla meme kanserinin erken teşhisini sağlamaktır. Bu sayede meme kanseri tedavi edilebilir safhada yakalanabilir ve ölüm oranı azaltılabilir.

Projenin Kapsamı

Proje kapsamında, Kaggle web sitesinden alınan Meme Kanseri Coimbra veri seti kullanılmıştır. Bu veri seti, gerçek bir veri örneğini temsil eden 116 katılımcıdan elde edilen verileri içerir. Her bir katılımcının 9 farklı özelliği bulunmaktadır. Bu özellikler vücut kitle indeksi (BMI), yaş, glukoz, insülin, Resistin, homeostaz modeli değerlendirmesi (HOMA), Leptin, Adiponektin, monosit kemoatraktan protein-1 (MCP-) gibi önemli parametreleri içerir. Veri seti, katılımcıları hasta ve normal olmak üzere iki gruba ayırmaktadır.

Projenin kapsamı, sağlık pratisyenlerinin karar vermelerine ve tanı koymalarına yardımcı olmak için en etkili makine öğrenimi algoritmalarını kullanarak hasta ve normal bireyleri sınıflandırmayı içerir. Bu amaçla, destek vektör makinesi (SVM) ve yapay sinir ağları (YSA) gibi algoritmalar kullanılmıştır. Bu algoritmaların kullanımıyla, meme kanserini erken teşhis etmek için bir araç geliştirilmesi hedeflenmiştir.

Sonuçlar

Projenin sonuçları, kesinlik, doğruluk, hatırlama ve F1 puanı gibi metrikler kullanılarak değerlendirilmiştir. Yapılan deneysel çalışmalar, önerilen yöntemin etkin olduğunu ve

erken teşhis için doktorlar tarafından kullanılabileceğini göstermektedir. Raporun ilerleyen bölümlerinde, yazılımla ilgili çalışmaların detayları ayrıntılı olarak açıklanacaktır. Bu proje, makine öğrenimi tekniklerinin tıp alanında önemli bir rol oynayabileceğini göstermektedir ve meme kanseri gibi önemli sağlık sorunlarının çözümünde potansiyel bir araç olarak kullanılabileceğini vurgulamaktadır.

Anahtar Kelimeler: Makine Öğrenmesi, Meme Kanseri, Yazılım, Destek Vektör Makinesi, Sınıflandırma, Yapay Sinir Ağları, Derin Öğrenme

ABSTRACT

Project Objective

The main objective of this project is to predict breast cancer using machine learning techniques. Breast cancer is one of the leading causes of death among women, and early detection is crucial. The aim of this project is to enable early detection of breast cancer by quickly processing and classifying routine blood test data obtained from various patients. This would allow for the detection of breast cancer at a treatable stage and reduce mortality rates.

Scope of Project

The project utilized the Breast Cancer Coimbra dataset obtained from the Kaggle website. This dataset consists of data from 116 participants, representing a real-world example. Each participant has 9 different features, including important parameters such as body mass index (BMI), age, glucose, insulin, Resistin, homeostasis model assessment (HOMA), Leptin, Adiponectin, monocyte chemoattractant protein-1 (MCP-). The dataset divides participants into two groups: those classified as patients and those classified as normal.

The scope of the project involves using the most effective machine learning algorithms to assist healthcare practitioners in decision-making and diagnosis by classifying patients and normal individuals. For this purpose, support vector machines (SVM) and artificial neural networks (ANN) algorithms were employed. The goal was to develop a tool for early detection of breast cancer using these algorithms.

Results

The results of the project were evaluated using metrics such as precision, accuracy, recall, and F1 score. Experimental studies indicate that the proposed method is effective and can be used by doctors for early diagnosis. The subsequent sections of the report will

provide detailed explanations of the software-related studies. This project demonstrates the potential role of machine learning techniques in the medical field and highlights their use as a potential tool in addressing significant health issues such as breast cancer.

Keywords: Machine Learning, Breast Cancer, Software, Support Vector Machine, Classification, Artificial Neural Networks, Deep Learning

TEŞEKKÜR

Bu projenin başından sonuna kadar hazırlanmasında emeği bulunan ve beni bu konuya yönlendiren sevgili Elektrik Elektronik Mühendisi arkadaşım Tuğçe PEKMEZCİ'ye ve saygıdeğer danışmanım Sayın Öğr. Gör. Yusuf Muştu'ya tüm katkılarından ve hiç eksiltmediği desteğinden dolayı teşekkür ederim.

Esra ASLAN

23 Mayıs 2024

İÇİNDEKİLER

SİMGE LİSTESİ

- ***Vücut Kitle İndeksi (BMI)***: Bir kişinin vücut ağırlığını boyuna göre değerlendirmek için kullanılan bir ölçümdür. BMI, bir kişinin kilogram cinsinden ağırlığının, metre cinsinden boyunun karesine bölünmesiyle hesaplanır.

  Formülü şu şekildedir:

$$\text{BMI} = \frac{\text{Weight (kg)}}{\text{Height (m)}^2}$$

  BMI, genellikle kilo durumunu değerlendirmede kullanılır ve aşağıdaki kategorilere göre yorumlanır:

  - 18.5'in altında: Zayıf
  - 18.5 - 24.9 arası: Normal
  - 25 - 29.9 arası: Fazla kilolu
  - 30 - 34.9 arası: Obez (Tip I)
  - 35 - 39.9 arası: Aşırı obez (Tip II)
  - 40 ve üzeri: Morbid obezite (Tip III)

  BMI, genellikle sağlık profesyonelleri tarafından obezite riskini değerlendirmek ve sağlık koşullarının yönetimine yardımcı olmak için kullanılır. Ancak, BMI'nin sadece vücut ağırlığını ve boyunu dikkate aldığı ve kas kütlesini hesaba katmadığı unutulmamalıdır. Bu nedenle, bazı durumlarda BMI, bir kişinin gerçek sağlık durumunu tam olarak yansıtmayabilir.

- ***Glukoz***: Kan dolaşımında bulunan bir şeker türüdür. Vücut hücreleri için enerji kaynağıdır ve normal metabolik fonksiyonlar için önemlidir.

- ***İnsülin:*** Pankreastan salgılanan bir hormondur ve kan şekerini düzenlemeye yardımcı olur. Hücrelere glukozun alınmasını sağlar ve karaciğerde glukozun depolanmasını teşvik eder.

- ***Resistin*** : Yağ dokusundan salgılanan bir protein hormonudur. İnsülin direnci ile ilişkilendirilmiş olup, metabolik sendromun gelişiminde rol oynayabilir.

- ***Homeostaz Model Değerlendirmesi (HOMA)***: İnsülin direncinin bir ölçüsüdür. Glukoz ve insülin seviyelerine dayalı bir hesaplama yöntemidir ve tip 2 diyabet riskini değerlendirmede kullanılır.

- ***Leptin***: Yağ dokusundan salgılanan bir hormondur. İştahı kontrol etmeye, enerji dengesini düzenlemeye ve metabolizmayı etkilemeye yardımcı olur.

- ***Adiponektin***: Yağ dokusundan salgılanan bir protein hormonudur. İnsülin hassasiyetini artırabilir, anti-inflamatuar etkilere sahip olabilir ve enerji metabolizmasını düzenleyebilir.

- ***Monosit Kemoatraktan Protein-1 (MCP-1)***: İnflamasyon süreçlerinde rol oynayan bir sitokin (hücreler arası iletişim molekülü) olan MCP-1, özellikle tip 2 diyabet ve kardiyovasküler hastalıklarla ilişkilendirilmiştir.

- ***Eğitim Veri Seti***: Bir makine öğrenimi modelinin öğrenme sürecinde kullanılan veri setidir. Bu veri seti, modelin belirli bir problemi çözmek için örüntüleri öğrenmesine yardımcı olur.

- ***Test Veri Seti***:Bir makine öğrenimi modelinin performansını değerlendirmek için kullanılan bağımsız bir veri setidir. Model eğitildikten sonra, genellikle performansını değerlendirmek için ayrı bir test veri seti kullanılır.

- ***Min-max normalizasyonu***: Verileri belirli bir aralığa (genellikle [0, 1] veya [-1, 1]) dönüştürmek için kullanılan bir ölçekleme tekniğidir. Bu normalizasyon yöntemi, veri setindeki değerleri orijinal aralıklarından çıkararak minimum değeri sıfıra ve maksimum değeri bir (veya -1) olarak ölçekler.

$$x_{\text{norm}} = \frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}}$$

- ***Bins methodu***: Verileri belirli aralıklara bölen ve her aralığın frekansını hesaplayan bir tekniktir. Bu teknik, veri dağılımının görsel olarak temsil edilmesi ve analiz edilmesi için sıklıkla kullanılır.

- ***data..head()***: Pandas DataFrame'in ilk beş satırını görüntülemek için kullanılır.
- ***data.isnull().sum()***: Veri setindeki eksik değerlerin sayısını hesaplar. Eksik değerler, veri setindeki boş veya tanımlanmamış hücreleri ifade eder.
- ***data.duplicated()***: Veri setindeki yinelenen satırları bulur. Yinelenen satırlar, tamamen aynı veya belirli sütunlara göre aynı olan satırları ifade eder.
- ***data=data.drop_duplicates()***: DataFrame'den yinelenen satırları kaldırır.
- ***data.info()***: DataFrame hakkında temel bilgileri gösterir, bu bilgiler arasında satır sayısı, sütun sayısı, veri tipleri ve eksik değerlerin sayısı bulunur.
- ***value_counts()***: Bir Pandas Serisi veya DataFrame sütunundaki benzersiz değerlerin sayısını verir.
- ***plt.figure()***:Matplotlib kütüphanesinde yeni bir grafik figürü oluşturmak için kullanılan bir fonksiyondur. Bu fonksiyon çağrısı, grafik çizimleri için bir alan oluşturur ve bu alana çizimlerin eklenmesine izin verir.
- ***class_counts.plot()*:** Pandas Serisi üzerinde doğrudan çağrıldığında, bu yöntem verileri çizmek için Matplotlib kütüphanesini kullanır. Örneğin, sınıf sayıları gibi verileri çubuk grafik, çizgi grafik veya başka bir grafik türüyle görselleştirmek için kullanılabilir.
- ***plt.title()*:** Matplotlib kütüphanesindeki bir fonksiyondur ve bir grafik figürüne başlık eklemek için kullanılır. Başlık, genellikle grafiğin içeriğini özetleyen bir metin veya açıklama olarak kullanılır.
- ***plt.xlabel()*:** x eksenine bir etiket (label) eklemek için kullanılır. Bu fonksiyon, x eksenindeki verilerin neyi temsil ettiğini belirtmek için kullanılır.

ŞEKİL LİSTESİ

1 GİRİŞ

1.1 Meme Kanserine Genel Bakış

Dünya genelinde kanser, ölümlerin ikinci önde gelen sebebidir; 2018 yılında 9.6 milyon birey bu hastalıktan yaşamını yitirmiştir. Araştırmalar, dünya çapında her 6 ölümden 1'inin kansere bağlı olduğunu ortaya koymaktadır [1]. Kadınlarda en yaygın kanser türleri arasında meme, akciğer ve kolon kanserleri yer almakta olup, bu üçü kadınlarda tüm kanser vakalarının yarısını oluşturmaktadır [2]. 2018 yılında dünya genelinde bildirilen 18.1 milyon kanser vakasının %11.6'sını meme kanseri oluşturmakta olup, bu oran meme kanserini akciğer kanserinden sonra ikinci sıraya yerleştirmektedir [3]. Meme kanseri, dünya genelinde ve Türkiye'de kadınlar arasında en yaygın görülen kanser türlerinden biridir [4].

Meme kanserinin ilk belirtileri, özellikle süt kanalları ve bezlerdeki meme dokusunda küçük tümörler veya kitlelerdir. Bir kitlenin sınırları düzgün ve net ise iyi huylu olarak kabul edilir; öte yandan, sınırları düzensiz ve yapısı pürüzlü ise, büyük olasılıkla kötü huyludur ve kanser tehlikesi taşır [5].

Meme kanseri riski birçok faktörle artmaktadır. Kadın olmak, yaş, ailede meme kanseri öyküsü bulunması, erken yaşta adet görme, tüm adet döngüsünün uzaması, geç menopoza girme, geç doğum yapma, çocuk sahibi olmama, uzun süreli oral kontraseptif kullanımı, önceki radyasyon maruziyeti, fiziksel aktivite eksikliği, emzirmeme ve kısa laktasyon süresi gibi östrojen maruziyet süresini azaltan bazı önemli risk faktörleri vardır. Ayrıca, yüksek yağlı diyet, çevredeki kimyasallara maruz kalma, sigara içme, alkol tüketimi, kürtaj ve çevre kirliliğinin de meme kanserinin gelişimine olumsuz etkileri bulunduğuna inanılmaktadır [6].

1.2 Meme Kanseri Tanı Yöntemlerine Genel Bakış

Bir kadının yaşam boyu meme kanserine yakalanma riski yüksek olduğu söylenir [7]. Bu nedenle, kanser tedavisini mümkün kılmak ve yaşam süresini uzatmak için erken tespit

çok önemlidir. Meme kanseri teşhisi, özellikle test verilerini yorumlarken, uzman insan bilgisi gerektirir [8]. Erken teşhis, ölüm riskini azaltır ve özellikle meme kanseri durumunda hastalara daha fazla tedavi seçeneği sunar [9].

Meme kanserini erken evrelerinde ya da semptomlar ortaya çıkmadan önce tespit edebilen tek yöntem röntgen mamografisidir [10]. Mamogramların yanlış sınıflandırılması durumlarında meme kanseri tahmininde hala iyileştirme gerekmektedir. Bu nedenle, erişilebilir ve uygun maliyetli yöntemlerden kaynaklanabilecek güvenilir belirleyiciler bulmak bir zorluk olarak kalmaktadır. Meme kanseri söz konusu olduğunda, rutin kan analizi mamografi ve manyetik rezonans görüntüleme (MRI) öncesinde kullanılabilecek ucuz ve kullanımı kolay bir prosedürdür çünkü erken tespit çok önemlidir [11].

1.3 Makine Öğrenimine Genel Bakış, Derin Öğrenme

Günümüz bilgisayarları insanlardan daha hızlı işlem yapabilmektedir, ancak karar verme yetenekleri insanlardan daha düşüktür. Bu nedenle bilgisayarların daha iyi analiz yapmasını ve karar vermesini sağlayan farklı makine öğrenme teknikleri geliştirilmiş ve geliştirilmektedir. Yapay zekâ teknolojilerinin yani derin öğrenme tekniklerinin gelişmesiyle birlikte hastalıkların teşhisinde önemli gelişmeler yaşanmıştır. Kümeleme, sınıflandırma yöntemleri, karar ağaçları, yapay sinir ağları gibi birçok teknik ile verilerden anlam çıkarımı ve tahmin yapılabilmektedir [12]. Meme kanserini tahmin etmek için derin öğrenme (DL) ve Makine öğrenmesi (ML) kavramları kullanılmaktadır. Meme kanserinin doğru teşhisini yapmak için bir dizi yöntem sunulmaktadır. Ayrıca, büyük veri ve makine öğrenimi kullanımı, tarama testlerinin doğruluğunu artırmak ve onları daha iyi yönlendirmek için yeni fırsatlar sunmaktadır.

2 KULLANILAN YAZILIMLAR VE YÖNTEMLER

2.1 Python Yazılım Dili

Python, genel amaçlı, yüksek seviyeli bir programlama dilidir. Girintilere güçlü bir vurgu yaparak, tasarım felsefesi kod okunabilirliğini önceliklendirir. Python, hafıza yönetimi ve dinamik tipleme kullanır. Nesne yönelimli, işlevsel ve yapılandırılmış (özellikle işlemsel) gibi çeşitli programlama paradigmalarıyla uyumludur. Geniş standart kütüphanesi nedeniyle sıkça "pil dahil" bir dil olarak adlandırılır. Python, Apache web sunucularında mod_wsgi ile barındırılan web siteleri için bir komut dosyası dili olarak kullanılabilir. Bu uygulamaları kolaylaştırmak için Web Sunucu Ağ Geçidi Arayüzü olarak bilinen bir standart API geliştirilmiştir. Tornado, Flask, Bottle, Zope, Django, Pylons, Pyramid, TurboGears ve web2py gibi web çerçeveleri, geliştiricilerin karmaşık uygulamalar oluşturmasına ve yönetmesine yardımcı olur. Demir ve Pyjs Ajax tabanlı uygulamalar, Python kullanılarak istemci tarafında geliştirilebilir. SQLAlchemy, ilişkisel veritabanı veri eşlemesi için kullanılabilir. Dropbox, bilgisayarlar arası bağlantıları programlamak için kullanılan Twisted çerçevesini kullanır. Python, ML tabanlı meme kanseri tahmini için tercih ettiğim programlama dilidir[29].

2.2 Veri Setinin Toplanması, Verilerin Hazırlanması ve Tanınması

Kaggle'dan alınan, meme kanseri hastaları ve sağlıklı kontrol hastalarına ait 9 klinik özellik ve bir sınıf değişkeni içeren büyük, temiz, gerçek bir veri seti Kaliforniya Üniversitesinden toplanmıştır. Veri seti excel dosyasında saklanmaktadır

Veri hazırlığı, veri setini modele uygun hale getirmek için yapılan ön işlemleri içerir. Eksik verilerin doldurulması, aykırı değerlerin ele alınması gibi adımlar bu sürecin bir parçasıdır. Veri tanıma ise, veri setinin yapısını anlamak ve ilişkileri görselleştirerek analiz etmek için yapılan işlemleri ifade eder. Görselleştirmeler, veri setinin içeriği hakkında daha derin bir anlayış sağlar.

Veri setini okumak ve analiz etmek için numpy, pandas, scipy ve seaborn gibi bazı faydalı kütüphaneler, import numpy as np ve import pandas as pd kodları ile içe aktarıldı.

2.2.1 Pandas Kütüphanesi

Pandas, Python programlama dilinde geliştirilen güçlü bir veri analizi ve manipülasyon kütüphanesidir. Adını "panel data" teriminden alır ve özellikle tablo benzeri verileri işlemek için tasarlanmıştır. Pandas, temel olarak iki ana veri yapısı olan Series ve DataFrame'i sunar. Series, tek boyutlu verileri, DataFrame ise çok boyutlu tablo benzeri verileri temsil eder. Bu yapılar, verilerin saklanması, işlenmesi ve analiz edilmesi için güçlü araçlar sunar.

2.2.2 Numpy Kütüphanesi

NumPy, Python'da kullanılan temel bir bilimsel hesaplama kütüphanesidir. Çok boyutlu diziler ve matrisler üzerinde hızlı işlemler yapmaya olanak tanır. Matematiksel fonksiyonlar, rastgele sayı üretimi ve lineer cebir işlemleri gibi birçok temel işlevi içerir. Veri analizi, mühendislik ve makine öğrenmesi gibi birçok alanda yaygın olarak kullanılır.

2.2.3 Scipy Kütüphanesi

Scipy, optimizasyon, interpolasyon, entegrasyon, Fourier dönüşümü, sinyal işleme, istatistik ve daha fazlasını içeren bir bilimsel ve teknik hesaplama kütüphanesidir.

2.2.4 Eksik ve tekrarlı değerler

Kan testi sonuçları, YSA analizine uygunluklarını artıracak yöntemler kullanılarak işlenir. Veri setindeki eksik, hatalı veya tutarsız veriler tespit edilip düzeltilmeli veya silinmelidir ***data.isnull().sum()*** veri setindeki eksik değerler (null) sayılır. Eksik değerler, veri setindeki bazı hücrelerin boş olması veya tanımlanmamış olması durumunda ortaya çı-

kar. Eksik değerler, veri analizi yaparken sorunlara yol açabilir. Eksik değerler nedeniyle veri setinin ortalama, standart sapma gibi istatistikleri yanlış hesaplanabilir veya eksik değerler içeren satırlar veya sütunlar analizden çıkarılabilir.

data.duplicated() veri setindeki yinelenen satırlar (duplicated) bulunur. Yinelenen satırlar, veri setindeki bazı satırların tamamen aynı olması veya belirli sütunlara göre aynı olması durumunda ortaya çıkar. Yinelenen satırlar, veri analizi yaparken sorunlara yol açabilir. Yinelenen satırlar nedeniyle veri setinin boyutu şişebilir veya yinelenen satırlar içeren sütunlar analizden çıkarılabilir.

data=data.drop_duplicates() komutu, DataFrame'den yinelenen satırları kaldırır.

data.info() komutu, DataFrame hakkında temel bilgiler verir. Bu bilgiler arasında satır sayısı, sütun sayısı, sütun isimleri, veri tipleri, bellek kullanımı ve eksik değerlerin sayısı bulunur.

2.3 Veri Görselleştirme

Görselleştirme veriyi analiz etmenin ve veriden belirli sonuçlar çıkarmanın (bilgi) en kolay yoludur. Yaptığımız bu görselleştirmeler, karmaşık bir şekilde görünen verileri kolayca anlamamıza yardımcı olur. Görselleştirmeler verideki ilişkileri, aykırı değerleri de fark etmemizi sağlar. Veri görselleştirme, veri biliminde keşifsel veri analizi (EDA) yapmamızı sağlar.

2.3.1 Matplotlib Kütüphanesi

Matplotlib, Python'da veri görselleştirmesi için kullanılan temel bir kütüphanedir. Grafik oluşturmak için kullanılır ve çeşitli grafik türlerini destekler. Basit ve esnek bir API'ye sahiptir, bu da kullanıcıların çeşitli grafikler oluşturmasını sağlar. Matplotlib, çizgi grafikleri, histogramlar, scatter plotlar ve bar grafikleri gibi birçok grafik türünü destekler. Veri analizi, veri görselleştirmesi ve raporlama gibi çeşitli uygulamalarda kullanılır.

2.3.2 Seaborn Kütüphanesi

Seaborn, Python'da veri görselleştirmesi için kullanılan bir kütüphanedir. Matplotlib'e dayalı olarak geliştirilmiştir ve daha çekici ve bilgilendirici grafikler oluşturmayı sağlar. Seaborn, basit bir API'ye sahiptir ve istatistiksel veri analizi için önceden tanımlanmış renk paletleri ve tema seçenekleri sunar. Bu özellikleri sayesinde, kullanıcılar çeşitli grafik türlerini kolayca oluşturabilir ve veri setlerinin görsel analizini yapabilirler. Seaborn, çizgi grafikleri, scatter plotlar, histogramlar, kutu grafikleri ve ısı haritaları gibi çeşitli grafik türlerini destekler.

2.3.3 Histogram Grafik

Histogramlar, verinin değerlerinin frekansını ve yoğunluğunu gösterir ve bir değişkenin dağılımı hakkında hızlı bir şekilde fikir edinmemizi sağlar. Bu dağılımlar genellikle normal dağılım olarak bilinir ve **Gaussian dağılımı** ya da **çan eğrisi** olarak da adlandırılır. Bir veri setiyle karşılaştığımızda, değişkenlerin olasılıksal dağılımlarını belirlemeye çalışırız. Bu, genellikle verinin normal dağılıp dağılmadığına bakarak yapılır. Normal dağılımı tanımlayan iki parametre vardır: ortalama ve standart sapma. Normal bir dağılımın özellikleri ortalama, mod ve medyan ile tanımlanır. Eğer bu değerler birbirine yakınsa veya eşitse, normal dağılımdan bahsedebiliriz. İki tür çarpık dağılım vardır. Tepe değeri > medyan > ortalama ise sola çarpık **(Negatively Skewed Distribution**) bir dağılım, tepe değeri < medyan < ortalama ise sağa çarpık **(Positively Skewed Distribution)** bir dağılım söz konusudur.

2.3.4 Çubuk Grafiği

Çubuk Grafiği, kategoriler arasındaki ayrımı yapmak ve sayısal karşılaştırmaları göstermek için yatay veya dikey sütunlar kullanır. Bir eksen, karşılaştırılan kategorileri temsil ederken, diğer eksen bir değer ölçeğini gösterir. Çubuk grafikleri, bir süre aralığındaki sürekli ve kesintisiz gelişmeleri göstermeyişleriyle histogramlardan ayrılır.

2.3.5 Pasta Grafiği

Pasta Grafikleri, bir daireyi orantılı bölümlere ayırarak, kategoriler arasındaki oranları ve yüzdeleri göstermeye yardımcı olur. Her bir yayın uzunluğu, her bir kategorinin oranını temsil ederken, tam daire %100'e eşit olan tüm verilerin toplamını temsil eder. Pasta grafikleri, okuyucunun verilerin orantılı dağılımı hakkında hızlı bir fikir edinmesine yardımcı olur.

2.4 Makine Öğrenmesi

Makine öğrenimi, bilgisayarların halihazırda var olan verilerden öğrenerek büyük, karmaşık veri kümelerindeki örüntüleri hızlı bir şekilde tanımlamasına yardımcı olmak için çeşitli istatistiksel, olasılıksal ve optimizasyon yaklaşımlarını kullanan bir yapay zeka alt alanıdır. ML, kapasitesi sayesinde kanserin teşhis ve tedavisinde de yaygın olarak kullanılmaktadır [20]. Algoritmaların veriler üzerinde eğitilme şekillerine bağlı olarak çeşitli makine öğrenimi kategorileri mevcuttur. Başlıca türleri şunlardır;

- **Denetimli Öğrenme:** Etiketli veriler veya bilinen bir çıktıya sahip veriler algoritmalara beslenir. Algoritmalar, girdi verilerini çıktı verilerine dönüştürmeyi öğrenir ve daha sonra yeni verileri tahmin etmek için bu bilgiyi kullanır. Örneğin, köpek ve kedi fotoğraflarını tanımlamayı öğrenebilir ve bu bilgiyi yeni görüntülerin sınıflandırılması için kullanabilir.

- **Denetimsiz Öğrenme:** Etiketsiz veriler veya bilinen bir çıktıya sahip olmayan veriler algoritmalara verilir. Algoritmalar, verilerdeki gizli özellikleri, kümeleri ve diğer kalıpları tanımlamak için eğitilir. Örneğin, müşterilerin demografik özelliklerini bilmeden, algoritma, onları satın alma alışkanlıklarına göre kategorize etmeyi öğrenebilir.

- **Takviyeli Öğrenme:** Girdi toplamak yerine, algoritmalar çevreleriyle etkileşime girer ve geri bildirim alarak öğrenir. Algoritmalar, bir görevi veya hedefi gerçekleştirmek için girdiye yanıt olarak faaliyetlerini en üst düzeye çıkarma yeteneği

kazanır. Örneğin, bir takviyeli öğrenme sistemi, bir video oyununu oynamak için farklı hamleler deneyerek ve puan kazanarak veya can kaybederek eğitilebilir[21].

2.4.1 Min-Max Normalizyonu

Makine öğrenimi algoritmalarının performansını artırmaya yardımcı olmak için özelliklerin orijinal birimlerini ve anlamını kaybetmemeye dikkat ederek her bir özelliği 0 ile 1 arasında bir aralığa dönüştürmek için DataFrame'in tüm sütunlarını min-max normalizasyonu ile normalleştirilir. Bu yöntem, verilerin dağılımını değiştirmez, ancak aykırı değerlerin etkisini azaltır.

Min-max normalizasyonu formülü aşağıdaki gibidir:

$$x_{\text{norm}} = \frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}}$$

Bu formülde, $x$orijinal değeri, x_{norm} normalleştirilmiş değeri, x_{min} sütundaki en küçük değer ve x_{max} sütundaki en büyük değeri temsil eder.

Bu formül, her bir özelliğin 0 ile 1 arasında bir değere dönüştürülmesine yardımcı olur, böylece algoritmaların performansını artırırken orijinal veri birimlerini ve anlamlarını korur.

2.4.2 Değişkenleri Ayırma Yöntemi

Veri analizi ve modelleme sürecinde, genellikle bir DataFrame'den hedef değişkenleri (bağımlı değişkenler) ve giriş değişkenleri (bağımsız değişkenler) ayırmamız gerekir. Hedef değişken, tahmin etmeye çalıştığımız değişkendir, genellikle sınıflandırma için kullanılır. Örneğin, hastaların sağlıklı veya hasta olup olmadığını tahmin etmek istiyorsak, bu değişken hedef değişken olacaktır. Giriş değişkenleri ise hedef değişkenini etkileyen veya açıklayan özelliklerdir.

```
# Hedef degiskeni secme
y = data['Classification']
```

```
# Giris degiskenlerini secme
x = data.drop(['Classification'], axis=1)
```

Burada, `data` DataFrame'inden 'Classification' adlı sütunu y değişkenine atıyoruz. Bu sütun, her gözlemin hangi sınıfa ait olduğunu belirtir. Ardından, 'Classification' sütununu çıkararak geri kalan tüm sütunları x değişkenine atıyoruz. Bu sütunlar, gözlemlerin özelliklerini veya niteliklerini belirtir.

```
# Sinif dagilimini gorsellestirme
class_counts = y.value_counts()
plt.figure(figsize=(4, 3))
class_counts.plot(kind='bar', color='skyblue')
plt.title('Sinif dagilimi')
plt.xlabel('Sinif')
plt.ylabel('Sayi')
plt.xticks(rotation=0)
```

Burada, `value_counts()` yöntemiyle her sınıfın veri kümesinde kaç kez göründüğünü hesaplıyoruz. Ardından, bir çubuk grafik oluşturuyoruz. Grafik başlığı, eksen etiketleri ve x eksenindeki etiketlerle birlikte grafik boyutunu ve renklerini ayarlıyoruz. Bu, sınıfların dağılımını görselleştirmek için kullanılır ve sunum dosyanıza eklemek için uygun bir grafik elde etmenizi sağlar.

2.4.3 Eğitim ve Test Veri Seti

Veri seti genellikle eğitim ve test veri setlerine bölünür. Eğitim veri seti, makine öğrenimi modelini eğitmek için kullanılırken, test veri seti modelin performansını ölçmek için kullanılır. Veri setinin bölünme oranı, test veri setinin toplam veri setine oranını belirtir.

```
from sklearn.model_selection import train_test_split
```

Burada, `train_test_split` adlı bir fonksiyon içe aktarılır. Bu fonksiyon, veri kümesini eğitim ve test alt kümelerine böler ve ardından bu alt kümelerin boyutlarını ve veri bölünme oranını görselleştirmek için bir pasta grafiği oluşturur.

```
xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size=0.1,
    stratify=y)
```

Bu satırda, `train_test_split` fonksiyonu ile veri kümesi eğitim ve test alt kümelerine bölünür. `xtrain` ve `xtest`, bağımsız değişkenlerin eğitim ve test veri setlerini, `ytrain` ve `ytest` ise bağımlı değişkenlerin eğitim ve test veri setlerini temsil eder. `test_size=0.1` parametresi, test alt kümesinin veri kümesinin %10'u kadar olacağını belirtir.

```
print("Egitim veri setinin boyutu:", xtrain.shape)
print("Test veri setinin boyutu:", xtest.shape)
```

Bu satırlar, eğitim ve test veri setlerinin boyutlarını yazdırır.

plt.pie() ile de eğitim ve test veri setlerinin oranlarını gösteren bir pasta grafiği oluşturur.

2.5 Destek Vektör Makinesi (SVM) Modeli

SVM, doğrusal veya doğrusal olmayan sınıflandırma, regresyon ve hatta aykırı değer tespiti görevleri için kullanılan güçlü bir makine öğrenimi algoritmasıdır. SVM, veri setindeki sınıfları ayırmak için en iyi ayrım çizgisini veya hiper düzlemi bulmaya çalışır. Bu ayrım çizgisinin, sınıflar arasındaki marjı maksimize etmesi beklenir. Marj, ayrım çizgisine en yakın noktaların (destek vektörleri) mesafesidir. SVM, doğrusal veya doğrusal olmayan ayrım çizgileri bulabilir. Bu model, yeni veriler üzerinde tahmin yapmak için kullanılabilir.

```
from sklearn.svm import SVC
svm_model = SVC(kernel='poly', gamma=8)
svm_model.fit(x_train, y_train)
```

`SVC` sınıfı `from sklearn.svm import SVC` komutuyla içe aktarılır. `svm_model` adlı bir nesne oluşturulur ve `kernel='poly', gamma=8` parametreleriyle bir polinom çekirdeği kullanarak bir SVM modeli oluşturulur. Polinom çekirdeği, doğrusal olmayan ayrım çizgileri bulmak için kullanılır. `gamma=8` parametresi, polinom çekirdeğinin

derecesini belirtir. Model, eğitim veri setleri (`x_train` ve `y_train`) ile eğitilir. Bu komut, eğitim veri setlerindeki bağımsız değişkenler ile bağımlı değişken arasındaki ilişkiyi öğrenmeye çalışır. Bu model, yeni veriler üzerinde tahmin yapmak için kullanılabilir.

2.5.1 SVM Modelinin Performansının Değerlendirilmesi

Doğruluk puanı, karmaşıklık matrisi ve sınıflandırma raporu gibi bazı metrikler kullanılarak SVM ile bir sınıflandırma modeli oluşturulur ve değerlendirilir. Bu metrikleri hesaplamak için `sklearn.metrics` modülünden bazı fonksiyonlar içe aktarılır. Eğitim veri kümesindeki özelliklerle modelin tahminleri üretilir ve doğruluğu hesaplanır. Tahminler arasında her sınıf için doğru ve yanlış tahminlerin sayısını gösteren karmaşıklık matrisi oluşturulur. Aynı işlemler test veri kümesindeki özelliklerle de tekrarlanır. Test veri kümesindeki her sınıf için hassasiyet, geri çağırma, F1-puanı ve destek gibi bazı metrikleri içeren bir sınıflandırma raporu oluşturulur ve ekrana yazdırılır. Bu metrikler, modelin performansını değerlendirmek için kullanılır.

2.5.2 Isı Haritası

Isı haritası, karmaşıklık matrisini görselleştirmek için kullanılmıştır. Renkler sınıflandırma sonuçlarının görsel bir temsilini sağlar. Matristeki her hücre, gerçek sınıfın (satırın) tahmin edilen sınıfla (sütun) eşleştiği örneklerin sayısını temsil eder. Isı haritasındaki renk tonları, yanlış tahminlerin ve doğru tahminlerin yoğunluğunu görselleştirir. Koyu renkler doğru tahminleri ve açık renkleri temsil eder.

2.5.3 Rastgele Örnekleme Yöntemi

Rastgele örnekleme, bir popülasyonun tüm üyeleri arasından rastgele seçilen örneklerle çalışma yöntemidir. Basit rastgele örnekleme, bu yöntemin en yaygın kullanılan türlerinden biridir. Her bir öğenin seçilme olasılığı aynıdır.

Veri çerçevesinin sınıflandırma modeli benzer prosedürlerle oluşturulur ve performansı

değerlendirilir.

`Classification` sütunu 0 ve 1 olan veri noktalarını seçer ve `df_class_0` ve `df_class_1` adlı yeni bir veri çerçevelerine atar. `df_class_1` veri çerçevesinden 250 adet veri noktası seçer ve `replace=True` parametresiyle aynı veri noktasının birden fazla seçilmesine izin verir. Böylece, `df_class_1` veri çerçevesinin boyutunu artırır ve `df_class_1_over` adlı yeni bir veri çerçevesine atar. Aynı işlemler `df_class_0` veri çerçevesi içinde tekrarlanır.

`df_class_0_over` ve `df_class_1_over` adlı iki veri çerçevesini `pd.concat` fonksiyonuyla birleştirir ve `axis=0` parametresiyle satır bazında birleştirir. Böylece, her iki sınıfın da eşit sayıda (250) veri noktasına sahip olduğu dengeli bir veri çerçevesi oluşturur ve `df_test_over` adlı yeni bir veri çerçevesine atar.

`df_test_over` veri çerçevesindeki veriler `x1` ve `y1` adlı iki değişken eğitim ve test veri kümelerine belirlenen oranlara göre ayırılır. Bu değişkenler, sırasıyla, eğitim veri kümesinin özellikleri, test veri kümesinin özellikleri, eğitim veri kümesinin etiketleri ve test veri kümesinin etiketleridir.

`random_state=0` parametresini, rastgeleliği kontrol etmek için kullanılır. Aynı parametre değeri ile çalıştırıldığında, fonksiyon aynı şekilde verileri böler.

`shuffle=True` parametresi, verilerin bölünmeden önce karıştırılacağını belirtir. Böylece, verilerin dağılımı eşitlenir.

`stratify=y1` parametresi, eğitim ve test veri kümelerindeki etiketlerin oranının `y1` değişkenindeki orana yakın olmasını sağlar. Böylece, sınıf dengesizliği önlenir.

2.6 Derin Öğrenme Modeli Oluşturulması ve Değerlendirilmesi

Derin öğrenme, karmaşık veri örüntülerini öğrenmek için derin ve karmaşık model yapılarını kullanan bir makine öğrenimi alt dalıdır. Bu algoritmalar, birçok katmana sahip sinir ağlarını kullanarak veri setlerinden doğrudan öğrenme yeteneğine sahiptir.

Derin öğrenme ile bir sınıflandırma modeli oluşturulur, eğitilir ve derlenir. Modelin eğitimini belirleyen parametreler arasında optimizer, loss ve metrics bulunur. Optimizer parametresi, modelin ağırlıklarını güncellemek için kullanılacak algoritmayı belirtir. Örneğin, *"adam"* optimizer'ı, adaptif öğrenme hızına sahip popüler bir algoritmadır. Ayrıca, *ModelCheckpoint* adlı bir geri arama fonksiyonu içe aktarılır. Bu fonksiyon, eğitim sırasında modelin en iyi performans gösteren ağırlıklarını bir dosyaya kaydeder.

num_epochs adlı bir değişken, modelin kaç kez eğitim verisini tamamlaması gerektiğini belirtir. Eğitim sırasında kaydedilen metrikler *history* adlı bir değişkende saklanır.

Modelin eğitimini başlatmak için girdi verisi (*X1_s_train*), çıktı verisi (*y1_s_train*), eğitim döngüsü sayısı (*epochs*), her döngüde kaç adım yapılacağı (*steps_per_epoch*), doğrulama verisi (*validation_data*) ve geri arama listesi (*callbacks*) gibi parametreler kullanılır.

Derin öğrenme modelinin performansını ölçmek için kayıp değeri ve metrik değeri kullanılır. Kayıp değeri, modelin ne kadar yanıldığını gösterir ve modelin tahminleri ile gerçek etiketler arasındaki farkı ölçer. Metrik değerler, modelin başarısını değerlendirmek için kullanılır. *Metrics* parametresi, modelin performansını değerlendirmek için kullanılacak metrikleri belirtir. Örneğin, *accuracy*, *precision*, *recall* ve *auc* gibi metrikler, modelin doğruluğunu, hassasiyetini, geri çağırma oranını ve alan altında eğri (area under curve) değerini hesaplar.

2.6.1 Tensorflow Kütüphanesi

Makine öğrenmesi için Google tarafından geliştirilen uçtan uca bir açık kaynak platformudur. Makine öğrenmesi uygulamaları için kapsamlı, esnek araçlar ve kütüphaneler ile topluluk kaynakları içeren bir ekosisteme sahiptir. Çok sayıda soyutlama seviyesi bulunduğu için çözülmek istenilen probleme uygun olanı seçme imkânı sunar.

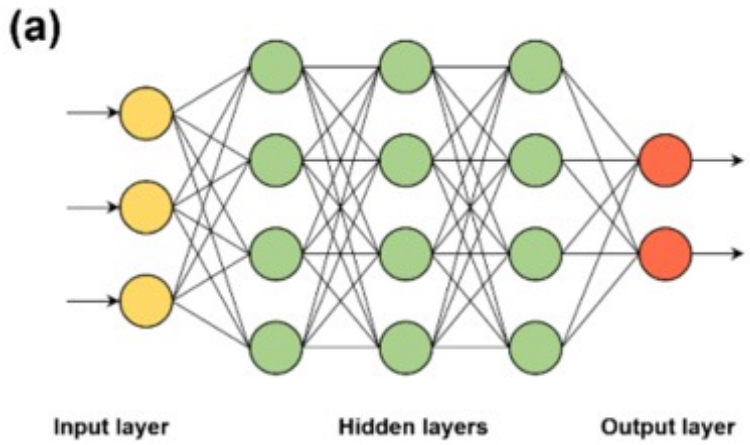
2.6.2 Keras Kütüphanesi

Neredeyse her tür derin öğrenme modelini tanımlamak ve eğitmek için uygun bir yol sağlayan Python için bir derin öğrenme kütüphanesidir. Keras, Tensorflow , Theano ve CNTK

üzerinde çalışabilen Python ile yazılmış bir üst düzey sinir ağları API'sıdır. İçerdiği çok fazla işlevsel fonksiyon sayesinde Keras kolayca bir derin öğrenme modeli oluşturmamızı ve onu eğitmemizi sağlar.

2.6.3 Derin Sinir Ağları (DSN'ler)

İkiden fazla nöron katmanına sahip bir YSA'ya DNN denir. Her katmandan geçen veriler toplama, çarpma ve fonksiyon uygulamaları dahil olmak üzere çeşitli matematiksel işlemlere tabi tutulabilir. Her bir nöronun bir sonraki katmanı ne kadar etkilediğini gösteren ağırlık sayıları katmanları birbirine bağlar. Öğrenme süreci boyunca, ağın belirli bir girdi için amaçlanan çıktıyı sağlamasını sağlamak için ağırlıklar değiştirilir [27].



Şekil 1: DNN modelinin tipik yapısı

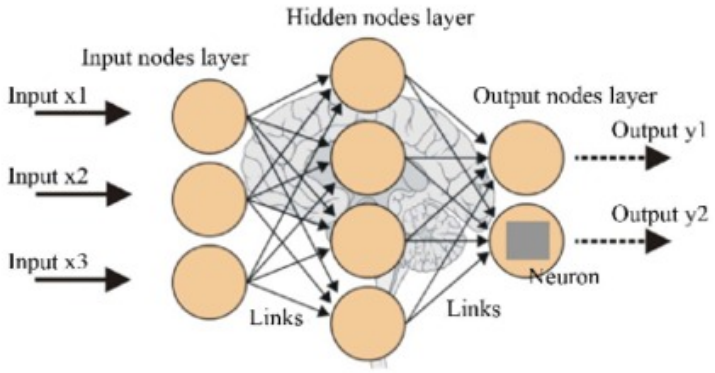
DNN'ler, karmaşık desenleri ve ilişkileri tanımlamak için metin, ses, görüntü ve ses dahil olmak üzere büyük hacimli verileri analiz edebilir. Ayrıca geleneksel algoritmalar için zor veya imkansız olan yüz tanıma, dil çevirisi, altyazı oluşturma, oyun oynama ve müzik besteleme gibi faaliyetleri de gerçekleştirebilirler. Transformatörler, tekrarlayan sinir ağları ve evrişimli sinir ağları DNN'lerin birkaç türüdür.

2.6.4 Çizgi Grafik

Çizgi grafikler, devam eden ya da belirli zaman aralığındaki "sayısal" değişimleri göstermek için kullanılır. Çizgi grafikler gruplandırıldığında, veriler arasındaki eğilimleri ve ilişkileri gösterir. Bu grafikler aynı zamanda olayların nasıl geliştiği ve süregeldiği konusunda "büyük resmin" anlaşılmasına da yardımcı olur.

2.7 Yapay Sinir Ağları

İnsan beyninin yapısından ve işleyişinden ilham alan bir tür makine öğrenimi modeli YSA'dır. Biyolojik bir sinir ağının mimarisi YSA'nınkine çok benzer [22]. Veri işleme ve gönderme yeteneğine sahip matematiksel yapılar olan bir dizi yapay nörondan oluşur.



Şekil 2: Bir YSA'nın temel yapısı[23]

Bir YSA üç katmandan oluşur: bir çıktı katmanı, bir gizli katman ve bir girdi katmanı. En üst katman olan giriş katmanı, dış kaynaktan gelen ham verilerin alındığı yerdir. Son katman olan çıktı katmanı ise ağın çıktısını üretir. Özellik çıkarma ve ara hesaplamaları gerçekleştiren ara katmanlar gizli katmanlar olarak adlandırılır. Bir YSA, ne kadar çok gizli katmana sahip olursa verilerden o kadar karmaşık ve soyut örüntüler öğrenebilir.

Her katmandaki nöronlar arasındaki bağlantıların ağırlığı değişir. Bu ağırlıklar, hedef değerlere yeterince yaklaşana kadar bağımsız olarak yinelemeli olarak değiştirilir. Ağırlıklar tam olarak kalibre edildiğinde, sistem eğitilmiş olarak kabul edilebilir. Test süreci bu

noktadan sonra başlayabilir [24]. Bir YSA için boyut azaltma, kümeleme, regresyon ve sınıflandırma dahil olmak üzere çeşitli kullanım alanları vardır. Daha karmaşık ve güçlü modeller geliştirmek için DL, evrişimli sinir ağları, tekrarlayan sinir ağları vb. gibi diğer makine öğrenimi yaklaşımlarıyla da entegre edilebilir.

3 MATERYAL OLUŞTURMA

3.1 DATASET

Projede Kaggle web sitesinden alınan Meme Kanseri Coimbra veri seti kullanılmıştır[32]. Veri seti, 64 meme kanseri hastası ve 52 sağlıklı kontrol hastası için 9 klinik değişken içeren 116 tam kayıttan oluşmaktadır. Yaş, BMI, yedi kan faktörü ve meme kanseri veya sağlıklı kontrol sınıfını gösteren ikili bir değişken bu özellikleri oluşturmaktadır. Sağlıklı kontroller 1 ile, hastalar ise 2 ile gösterilmektedir; bunlar aşağıdaki veri çerçevesinde gösterilmektedir.

Index	Age	BMI	Glucose	Insulin	HOMA	Leptin	Adiponectin	Resistin	MCP.1	classification
0	48	23.5	70	2.707	0.467409	8.8071	9.7024	7.99585	417.114	1
1	83	20.6905	92	3.115	0.706897	8.8438	5.42929	4.06405	468.786	1
2	82	23.1247	91	4.498	1.00965	17.9393	22.432	9.27715	554.697	1
3	68	21.3675	77	3.226	0.612725	9.8827	7.16956	12.766	928.22	1
4	86	21.1111	92	3.549	0.805386	6.6994	4.81924	10.5763	773.92	1
5	49	22.8545	92	3.226	0.732087	6.8317	13.6798	10.3176	530.41	1
6	89	22.7	77	4.69	0.890787	6.964	5.58986	12.9361	1256.08	1
7	76	23.8	118	6.47	1.8832	4.311	13.2513	5.1042	280.694	1
8	73	22	97	3.35	0.801543	4.47	10.3587	6.28445	136.855	1
9	75	23	83	4.952	1.01384	17.127	11.579	7.0913	318.302	1
10	34	21.47	78	3.469	0.667436	14.57	13.11	6.92	354.6	1

Şekil 3: Data Frame

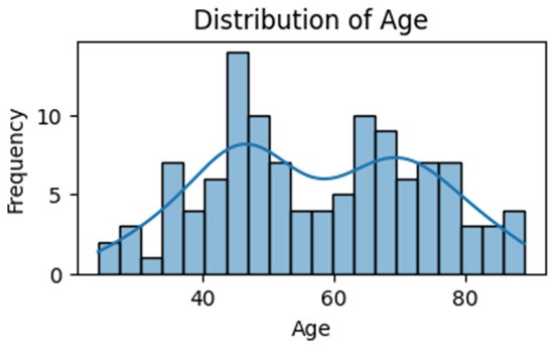
4 Çıktılar ve Yorumlar

Şekil 4'ü incelediğimizde antropometrik veriler ve kan testi verileri kullanılarak oluşturulan veri setinin pandas DataFrame nesnesine dönüştürülmüş ilk beş satırını görüyoruz. Bu tabloda her satır bir hastayı, her sütun ise bir biyokimyasal parametreyi veya sınıflandırmayı temsil etmektedir. Sınıflandırma, hastanın meme kanseri olup olmadığını gösterir. 1 olarak etiketlenen hastalar meme kanseri olan hastaları, 2 olarak etiketlenenler ise meme kanseri olmayan hastaları ifade etmektedir.

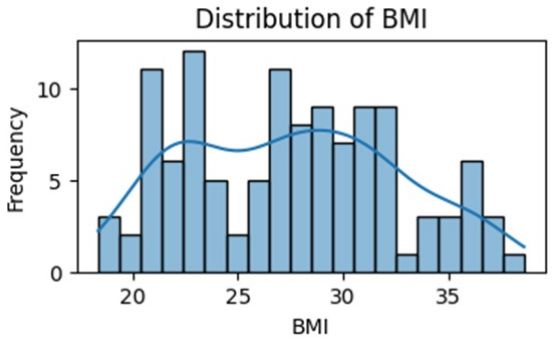
	Age	BMI	Glucose	Insulin	HOMA	Leptin	Adiponectin	Resistin	MCP.1	Classification
0	48	23.500000	70	2.707	0.467409	8.8071	9.702400	7.99585	417.114	1
1	83	20.690495	92	3.115	0.706897	8.8438	5.429285	4.06405	468.786	1
2	82	23.124670	91	4.498	1.009651	17.9393	22.432040	9.27715	554.697	1
3	68	21.367521	77	3.226	0.612725	9.8827	7.169560	12.76600	928.220	1
4	86	21.111111	92	3.549	0.805386	6.6994	4.819240	10.57635	773.920	1

Şekil 4: DataFrame'in ilk beş satırı

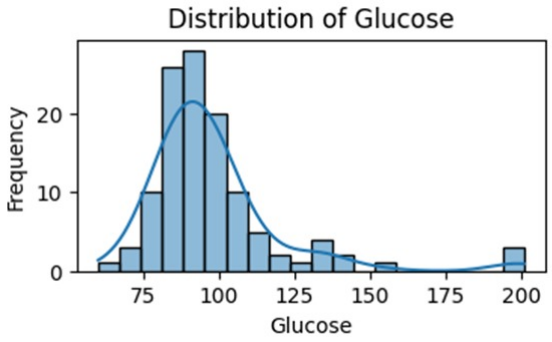
Şekil 5'den itaberen veri dağılımını görselleştirmek için her sütun için bir histogram çizilmiştir. Şekil 5'de gördüğümüz yaş dağılımı grafiği bize mevcut veri setimizde 40-60 ve 60-80 yaş aralığındaki kişilerin yoğunluğunun daha fazla olduğunu göstermektedir. Buna bağlı olarak dalgalanmanın da bu aralıklarda zirve yaptığı gözlemlenmiştir. Görüldüğü üzere, veri dağılım grafiklerini görselleştirerek, büyük veri setlerindeki değişkenlerin dağılımının yorumlanmasını daha kolay algılayabilir ve bundan çeşitli anlamlar ve yorumlar çıkarabiliriz.



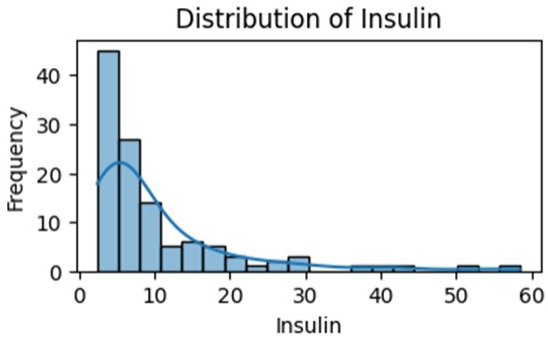
Şekil 5: Veri dağılım grafiği (Yaş Dağılımı)



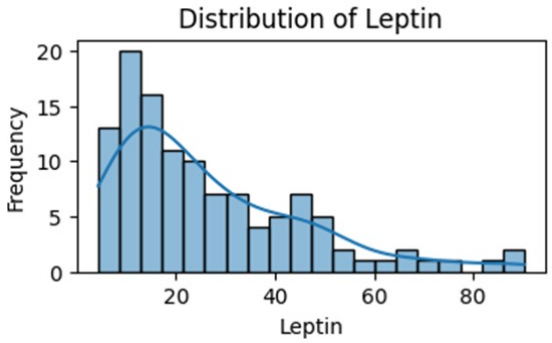
Şekil 6: Veri dağılım grafiği (BMI dağılımı)



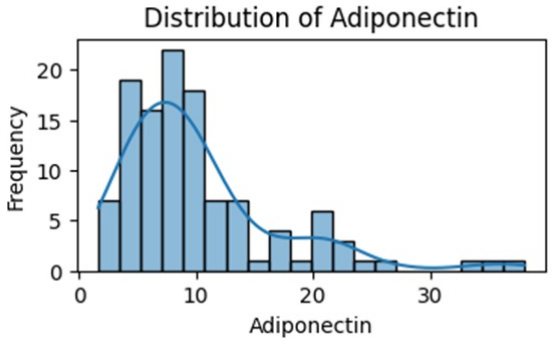
Şekil 7: Veri dağılım grafiği (Glikoz Dağılımı)



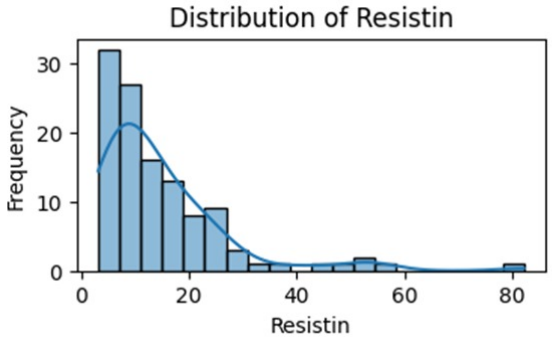
Şekil 8: Veri dağılım grafiği (İnsülin Dağılımı)



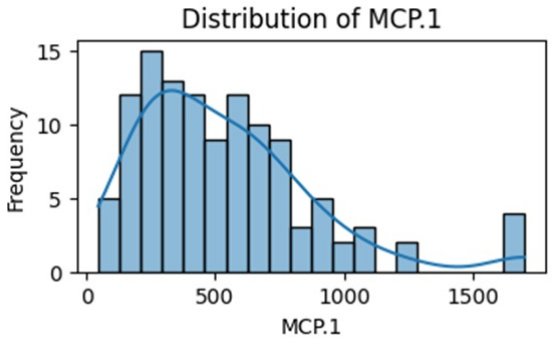
Şekil 9: Veri dağılım grafiği (HOMA dağılımı)



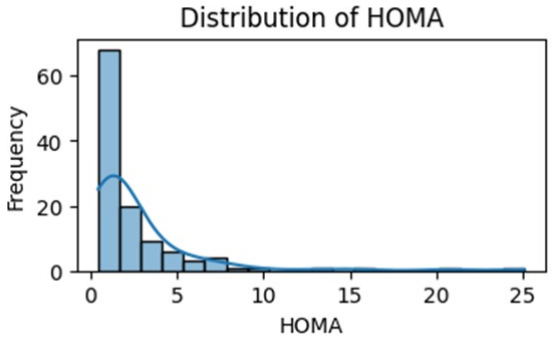
Şekil 10: Veri dağılım grafiği (Leptin dağılımı)



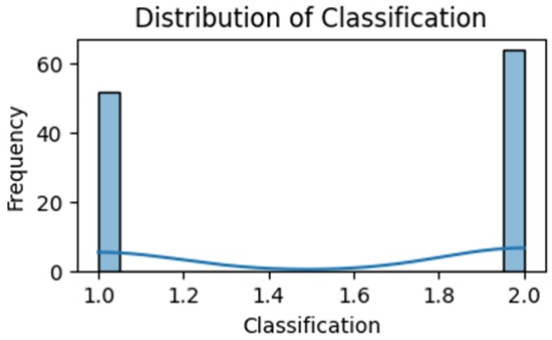
Şekil 11: Veri dağılım grafiği (Adiponektin dağılımı)



Şekil 12: Veri dağılım grafiği (Resistin dağılımı)



Şekil 13: Veri dağılım grafiği (MCP.1'in dağılımı)



Şekil 14: Veri dağılım grafiği (Sınıflandırma Dağılımı)

Şekil 15'deki çıktı, bir DataFrame'deki eksik değerleri kontrol etmek için kullanılan bir işlemi göstermektedir. DataFrame'deki her sütundaki eksik değerlerin sayısını hesaplamak için kullanılır. Bu çıktı, her sütun için eksik değerlerin sayısını gösterir. Bu veri setinde eksik değer olmadığı görülmektedir. Eğer eksik değer olsaydı, ilgili sütunda sıfır olmayan bir sayı görürdük.

```
Age             0
BMI             0
Glucose         0
Insulin         0
HOMA            0
Leptin          0
Adiponectin     0
Resistin        0
MCP.1           0
Classification  0
dtype: int64
```

Şekil 15: DataFrame'deki her sütun için eksik değer sayısı

Şekil 16 her bir satır için yineleme durumunu göstermektedir. Bu veri setinde yinelenen bir satırın olmadığı görülmektedir. Eğer yinelenen bir satır olsaydı, ilgili satırda True değerini görürdük.

```
0      False
1      False
2      False
3      False
4      False
       ...
111    False
112    False
113    False
114    False
115    False
Length: 116, dtype: bool
```

Şekil 16: DataFrame'deki her satır için yineleme durumu

Şekil 17, DataFrame'in 116 satır ve 115 sütundan oluştuğunu, her sütunda eksik değer bulunmadığını ve bellek kullanımının yaklaşık 10,0 kilobayt olduğunu göstermektedir.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 116 entries, 0 to 115
Data columns (total 10 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Age             116 non-null    int64
 1   BMI             116 non-null    float64
 2   Glucose         116 non-null    int64
 3   Insulin         116 non-null    float64
 4   HOMA            116 non-null    float64
 5   Leptin          116 non-null    float64
 6   Adiponectin     116 non-null    float64
 7   Resistin        116 non-null    float64
 8   MCP.1           116 non-null    float64
 9   Classification  116 non-null    int64
dtypes: float64(7), int64(3)
memory usage: 10.0 KB
```

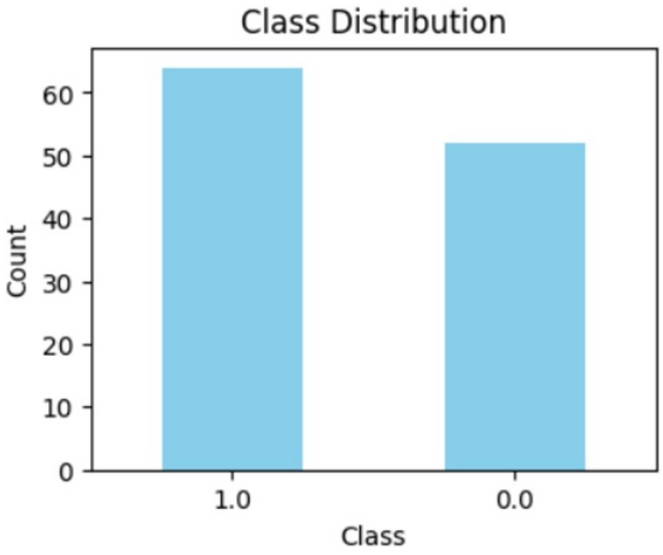
Şekil 17: DataFrame hakkında temel bilgiler

Şekil 18, veri kümesindeki her bir özelliğin 0 ile 1 arasında bir aralığa dönüştürülmüş ilk beş satırını göstermektedir.

	Age	BMI	Glucose	Insulin	HOMA	Leptin	Adiponectin	Resistin	MCP.1	Classification
0	0.369231	0.253850	0.070922	0.004908	0.000000	0.052299	0.221152	0.060665	0.224659	0.0
1	0.907692	0.114826	0.226950	0.012190	0.009742	0.052726	0.103707	0.010826	0.255926	0.0
2	0.892308	0.235278	0.219858	0.036874	0.022058	0.158526	0.571021	0.076906	0.307912	0.0
3	0.676923	0.148328	0.120567	0.014171	0.005911	0.064811	0.151538	0.121131	0.533934	0.0
4	0.953846	0.135640	0.226950	0.019936	0.013748	0.027782	0.086940	0.093375	0.440565	0.0

Şekil 18: DataFrame'in ilk beş satırı

Şekil 19'daki 'Sınıf' sütununda 0 ve 1 olmak üzere iki farklı sınıf değeri bulunduğundan, grafikte iki çubuk görülmektedir. Her çubuk, ilgili sınıf değerinin yüksekliğine bağlı olarak meydana gelme sayısını temsil eder. Bu tablo, sınıf değerlerinin dağılımını hızlı bir şekilde görselleştirir ve sınıflar arasındaki sayısal farklılıkları vurgular. Sınıf dengesizliği durumunda, yani sınıflar arasında büyük bir fark varsa, bu grafik bu dengesizliği açıkça gösterecektir.

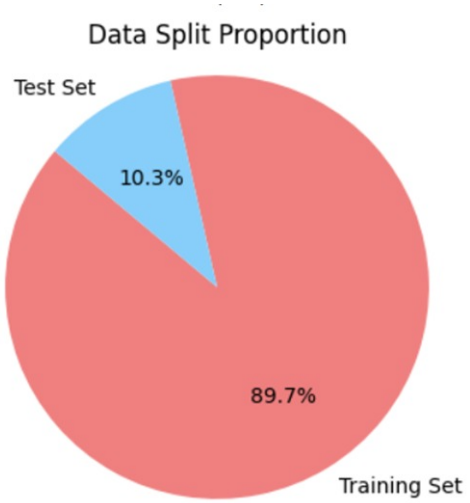


Şekil 19: Veri kümesindeki sınıf dağılımını gösteren çubuk grafik

Şekil 20'de (104, 9) çıktısı xtrain veri setinin 104 satır ve 9 sütundan oluştuğunu göstermektedir. Eğitim ve test alt kümelerinin oranlarını gösteren Şekil 21'deki pasta grafiğine baktığımızda, test kümesinin %10,3 ve eğitim kümesinin %89,7 olduğunu görüyoruz.

```
Training Set Size: (104, 9)
Test Set Size: (12, 9)
Training Label Set Size: (104,)
Test Label Set Size: (12,)
```

Şekil 20: Eğitim ve test alt kümelerinin boyutları



Şekil 21: Eğitim ve test alt kümelerinin oranlarını gösteren pasta grafiği

Şekil 22'da modelin bir polinom kernel kullanacağı ve polinom kernelin derecesinin 8 olduğu görülmektedir.

```
SVC
SVC(gamma=8, kernel='poly')
```

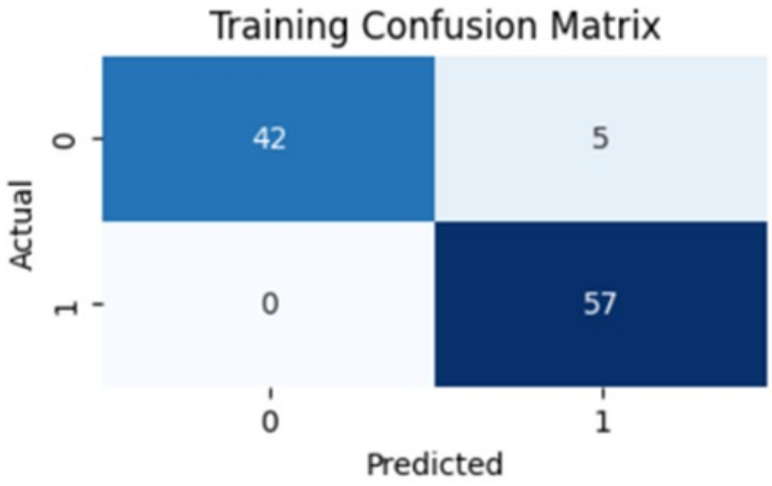
Şekil 22: SVM Model

Şekil 23'deki çıktıdaki değerlere odaklandığımızda, sol üst köşedeki (42) değeri "Sağlıklı" sınıfının doğru tahmin edildiği örnek sayısını temsil etmektedir. Sağ üst köşedeki (5) değeri ise "Sağlıklı" sınıfının "Hasta" olarak yanlış tahmin edildiği vaka sayısını göstermektedir. Sol alt köşedeki (0) değeri, "Hasta" sınıfının "Sağlıklı" olarak yanlış tahmin edildiği vaka sayısını göstermektedir. Sağ alt köşedeki değer (57) "Hasta" sınıfının doğru tahmin edildiği örnek sayısını gösterir.

Nokta değerleri Şekil 24'de gösterilmektedir. '0.0': ' precision ': 1.0000, 'recall': 0.893617, 'f1 skoru': 0.943820, 'destek': 47" bölümü, hesaplanan duyarlılık, "Sağlıklı" sınıfı için F1 puanı, geri çağırma ve "Sağlıklı" sınıfı için tahminlerin ne kadar doğru olduğunu, kaç gerçek pozitifin doğru tahmin edildiğini ve modelin bu sınıf için ne kadar iyi performans

gösterdiğini gösterir. " ’1.0’: ’precision’: 0,919355, ’geri çağırma’: 1.0000 , ’f1 puanı’: 0.957985, ’destek’: 57" "Hasta" sınıfı için hesaplanan hassasiyet, geri çağırma ve F1 puanını temsil eder. Bu değerler "Hasta" sınıfının tahminlerinin ne kadar doğru olduğunu, kaç doğru pozitifin doğru tahmin edildiğini ve modelin bu sınıf için ne kadar iyi performans gösterdiğini belirtir.

Şekil 24’de modelin eğitim veri kümesindeki doğruluğu 95,1923076923077 olarak hesaplanmıştır. Şekil 26’da, modelin test veri kümesindeki doğruluğu 58.3333333336 olarak hesaplanmıştır.



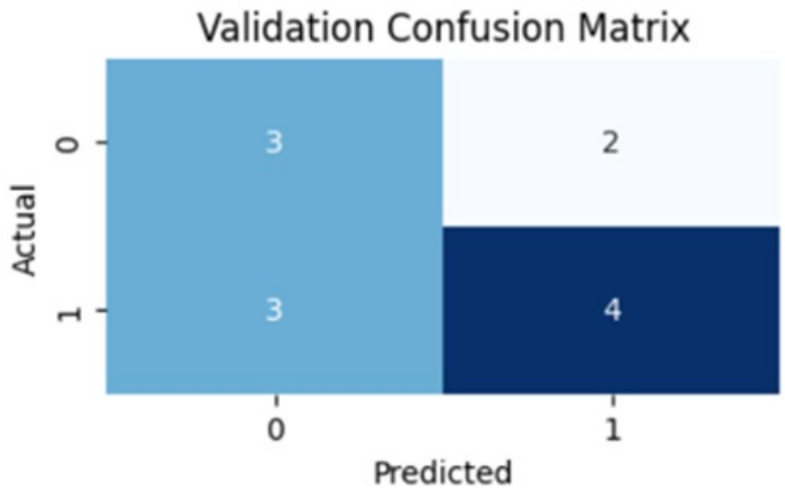
Şekil 23: Eğitim veri setindeki karmaşıklık matrisi

Training Classification Report:

	precision	recall	f1-score	support
0.0	1.000000	0.893617	0.943820	47.000000
1.0	0.919355	1.000000	0.957983	57.000000
accuracy	0.951923	0.951923	0.951923	0.951923
macro avg	0.959677	0.946809	0.950902	104.000000
weighted avg	0.955800	0.951923	0.951583	104.000000

Training Accuracy: 95.1923076923077

Şekil 24: Eğitim veri setindeki sınıflandırma raporu ve modelin eğitim veri setindeki doğruluğu



Şekil 25: Test veri kümesi üzerinde karmaşıklık matrisi

Validation Classification Report:

	precision	recall	f1-score	support
0.0	50.000000	60.000000	54.545455	500.000000
1.0	66.666667	57.142857	61.538462	700.000000
accuracy	58.333333	58.333333	58.333333	58.333333
macro avg	58.333333	58.571429	58.041958	1200.000000
weighted avg	59.722222	58.333333	58.624709	1200.000000

Validation Accuracy: 58.333333333333336

Şekil 26: Test veri kümesi üzerinde sınıflandırma raporu ve modelin test veri kümesi üzerindeki doğruluğu

Şekil 27'deki çıktı, 'Sınıf' sütununda bulunan farklı sınıf değerlerinin sayısını göstermektedir. İlk sütun sınıf değerlerini temsil ederken, ikinci sütun her bir sınıf değerinin kaç kez tekrarlandığını gösterir. Bu durumda, sınıf değeri 0'dır (sağlıklı) ve toplamda 52 kez tekrarlanmıştır. Sınıf değeri 1 (meme kanseri) toplam 64 kez tekrarlanmıştır. Bu bilgi sınıf dağılımını anlamak için kullanılır. Örneğin bu veri setinde 0 sınıfına ait örnek sayısı 52, 1 sınıfına ait örnek sayısı ise 64'tür. Bu bilgi, sınıflar arasındaki denge veya dengesizliğin göstergesidir. Sınıflar arasında çok fazla fark olmadığı için modelin dengeli olabileceğini söyleyebiliriz. Eğer büyük bir fark varsa, sınıf dengesizliği ile ilgili önlemler alınması

gerekebilir. Şekil 28'deki bu çıktı aynı zamanda sınıflandırma modelini değerlendirmek için de kullanılacaktır. Bu bilgi, modelin hangi sınıfı daha iyi tahmin ettiğini veya hangi sınıfın en hatalı tahminlere sahip olduğunu değerlendirmek için kullanılabilir.

```
1.0    64
0.0    52
Name: Classification, dtype: int64
```

Şekil 27: Meme kanseri olan (Hasta) ve meme kanseri olmayan (Sağlıklı) kişilerin sayısını gösteren veri çerçevesi

Şekil 28'teki çıktıda rastgele örnekleme yöntemi ile veri çerçevesi dengeli bir şekilde arttırılarak sınıflandırma modeli için oluşturulan df_test_over isimli veri çerçevesi görülmektedir.

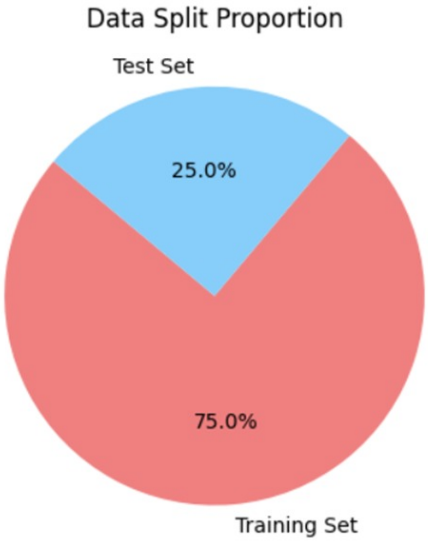
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 500 entries, 21 to 114
Data columns (total 10 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Age             500 non-null    float64
 1   BMI             500 non-null    float64
 2   Glucose         500 non-null    float64
 3   Insulin         500 non-null    float64
 4   HOMA            500 non-null    float64
 5   Leptin          500 non-null    float64
 6   Adiponectin     500 non-null    float64
 7   Resistin        500 non-null    float64
 8   MCP.1           500 non-null    float64
 9   Classification  500 non-null    float64
dtypes: float64(10)
memory usage: 43.0 KB
```

Şekil 28: df_test_over veri çerçevesinin temel bilgileri

Şekil 29'da (125, 9) çıktısı x1test veri setinin 125 satır ve 9 sütundan oluştuğunu göstermektedir. Eğitim ve test alt kümelerinin oranlarını gösteren Şekil 30'daki pasta grafiğine baktığımızda, test kümesinin %25, eğitim kümesinin ise %75 olduğunu görüyoruz.

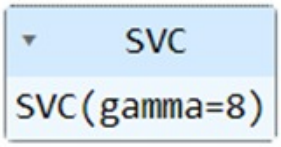
```
Training data shape is :(375, 9).
Training label shape is :(375,).
Testing data shape is :(125, 9).
Testing label shape is :(125,).
```

Şekil 29: x1 ve y1 değişkenlerinin eğitim ve test veri setlerinin boyutları



Şekil 30: Veri bölünmesini gösteren bir pasta grafik

Şekil 31, verilerin doğrusal olmayan bir şekilde ayrılmasını sağlayan rbf çekirdeğinin svc_s_modelinde kullanıldığını göstermektedir.



Şekil 31: SVM Model

Şekil 32'deki çıktıdaki değerlere odaklandığımızda, sol üst köşedeki (179) değeri "Sağlıklı" sınıfının doğru tahmin edildiği örnek sayısını temsil etmektedir. Sağ üst köşedeki (8) değeri ise "Sağlıklı" sınıfının "Hasta" olarak yanlış tahmin edildiği vaka sayısını göstermektedir. Sol alt köşedeki (2) değeri, "Hasta" sınıfının "Sağlıklı" olarak yanlış tahmin edildiği vaka sayısını göstermektedir. Sağ alt köşedeki değer (186) ise "Hasta" sınıfının doğru tahmin edildiği örnek sayısını göstermektedir.

Şekil 34'deki çıktıdaki değerlere odaklandığımızda sol üst köşedeki değer (60) modelin "Sağlıklı" sınıfını doğru tahmin ettiği örnek sayısını temsil etmektedir. Sağ üst köşedeki (3) değeri ise "Sağlıklı" sınıfının "Hasta" olarak yanlış tahmin edildiği örnek sayısını göstermektedir. Sol alt köşedeki (2) değeri, "Hasta" sınıfının "Sağlıklı" olarak yanlış tahmin edildiği vaka sayısını göstermektedir. Sağ alt köşedeki değer (60) ise "Hasta" sınıfının doğru tahmin edildiği örnek sayısını göstermektedir. Bu çıktı, modelin yüksek doğruluğunun bir sonucu olarak gözlemlenmektedir.

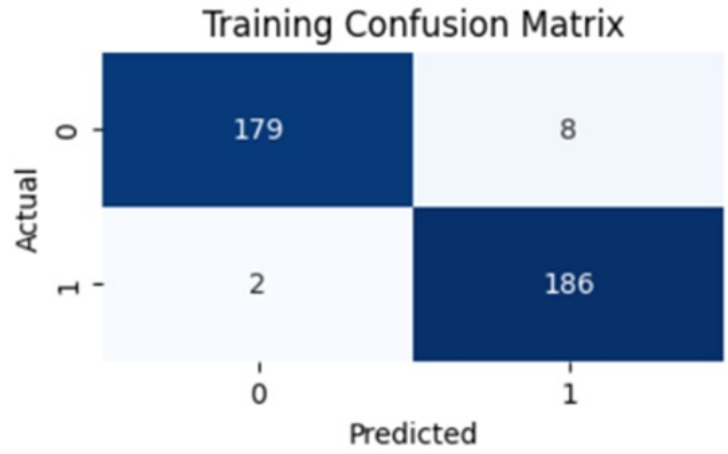
Nokta değerleri Şekil 35'de gösterilmektedir. '0.0': 'precision': 0.967742, 'recall': 0.952381, 'f1 skoru': 0.96, 'destek': 63" bölümü, "Sağlıklı" sınıfı için hesaplanan duyarlılık, F1 puanı, geri çağırma ve "Sağlıklı" sınıfı için tahminlerin ne kadar doğru olduğunu, kaç gerçek pozitifin doğru tahmin edildiğini ve modelin bu sınıf için ne kadar iyi performans gösterdiğini gösterir.

" '1.0': 'precision': 0,952381, 'geri çağırma': 0.967742, 'f1 puanı': 0.96, 'destek': 62" "Hasta" sınıfı için hesaplanan hassasiyet, geri çağırma ve F1 puanını temsil eder. Bu değerler, "Hasta" sınıfının tahminlerinin ne kadar doğru olduğunu, kaç doğru pozitifin doğru tahmin edildiğini ve modelin bu sınıf için ne kadar iyi performans gösterdiğini gösterir. "Doğruluk": 0.960000", modelin test veri setindeki doğruluk oranını temsil eder. Bu değer, tüm sınıflar için doğru tahminlerin yüzdesini temsil eder.

" 'Makro ortalama': 'precision': 0.960061, 'recall':0.960061, 'f1 score': 0.96, 'support': 125", tüm sınıfların temsili için ortalama hassasiyet, geri çağırma ve F1 puanlarıdır. " 'ağırlıklı ortalama': 'precision': 0.960123, 'geri çağırma': 0.960000, 'f1 puanı': 0.96, 'destek': 125" tüm sınıfların hassasiyet, geri çağırma ve F1 puanlarının ağırlıklı ortalamasını temsil eder.

Sınıflandırma sonuçları "classification_report" raporlarında, sınıflandırma modeli için veri çerçevesinin rastgele örnekleme yöntemiyle dengeli bir şekilde artırılmasının modelin performansı üzerindeki etkisi açıkça görülmektedir. Parametre ayarlamalarının olumlu etkisi Şekil 24 ve Şekil 33 karşılaştırılarak görülmüştür. Şekil 24'de modelin eğitim veri kümesindeki doğruluğu 95,1923076923077 olarak hesaplanırken, oluşturulan yeni modelde eğitim veri kümesinin doğruluğu 97,333333333334 olarak hesaplanmıştır. Aynı şekilde Şekil 2'de modelin test veri kümesindeki doğruluğu 58.3333333336 olarak hesaplanmış, oluşturulan yeni modelde ise test veri kümesinin doğruluğu 96.0 olarak hesaplanmıştır.

Ortaya çıkan değerlendirme değerleri yüzde 90'ın üzerindedir. Bu değerlerin yüzde 90'ın üzerinde olması modelimizin iyi çalıştığını göstermektedir.



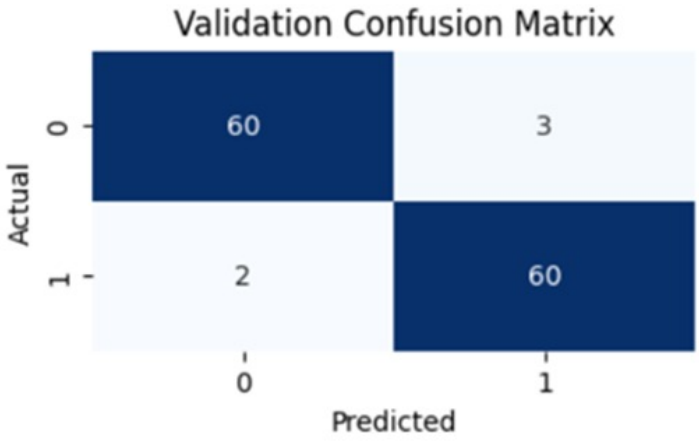
Şekil 32: Eğitim veri setindeki karmaşıklık matrisi

Training Classification Report:

	precision	recall	f1-score	support
0.0	0.988950	0.957219	0.972826	187.000000
1.0	0.958763	0.989362	0.973822	188.000000
accuracy	0.973333	0.973333	0.973333	0.973333
macro avg	0.973857	0.973290	0.973324	375.000000
weighted avg	0.973816	0.973333	0.973325	375.000000

Training Accuracy: 97.33333333333334

Şekil 33: Eğitim veri kümesindeki sınıflandırma raporu ve modelin eğitim veri kümesindeki doğruluğu



Şekil 34: Test veri kümesi üzerinde karmaşıklık matrisi

Validation Classification Report:

	precision	recall	f1-score	support
0.0	0.967742	0.952381	0.96	63.00
1.0	0.952381	0.967742	0.96	62.00
accuracy	0.960000	0.960000	0.96	0.96
macro avg	0.960061	0.960061	0.96	125.00
weighted avg	0.960123	0.960000	0.96	125.00

Validation Accuracy: 96.0

Şekil 35: Test veri kümesi üzerinde sınıflandırma raporu ve modelin test veri kümesi üzerindeki doğruluğu

Şekil 36'daki çıktı modelin bir özetini temsil eder ve aşağıdaki bilgileri içerir:

Model adı: "sıralı"

Her katmanın adı, türü ve çıktısı.

Her katman için parametre sayısı.

Toplam parametre sayısı.

Eğitilebilir parametre sayısı.

Eğitilemeyen parametrelerin sayısı.

Bu çıktı, modelin genel yapısını ve her katmandaki parametre sayısını gösterir. Örneğin, ilk katmanın adı "dense (Dense)" ve çıktı modeli "(None, 256)" şeklindedir. Yoğun katman, her bir giriş nöronunun her bir çıkış nöronuna bağlandığı tam bağlantılı bir katmandır. Bu katman aynı zamanda modelin giriş katmanıdır. Bu katmanda toplam 2560 pa-

rametre bulunmakla birlikte bunların hiçbiri eğitilememektedir. İkinci katman "dense_1" olarak adlandırılır ve modelin gizli katmanlarından biridir.

Son katman "dense_2" olarak adlandırılır ve 1 çıkış birimine sahiptir. Bu katman aynı zamanda modelin çıkış katmanıdır. Modelin çıktısı olan değer, veri noktasının 0 veya 1 sınıfına ait olma olasılığını gösterir. Tablo 5.1'deki bu çıktı modelin yapısal bilgilerini ve parametre sayısını göstermektedir. Bu bilgiler modelin özetini ve genel karmaşıklığını anlamak için kullanılır.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 256)               2560

 dense_1 (Dense)             (None, 512)               131584

 dense_2 (Dense)             (None, 1)                 513

=================================================================
Total params: 134657 (526.00 KB)
Trainable params: 134657 (526.00 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Şekil 36: DL modelinin yapısını gösteren bir tablo

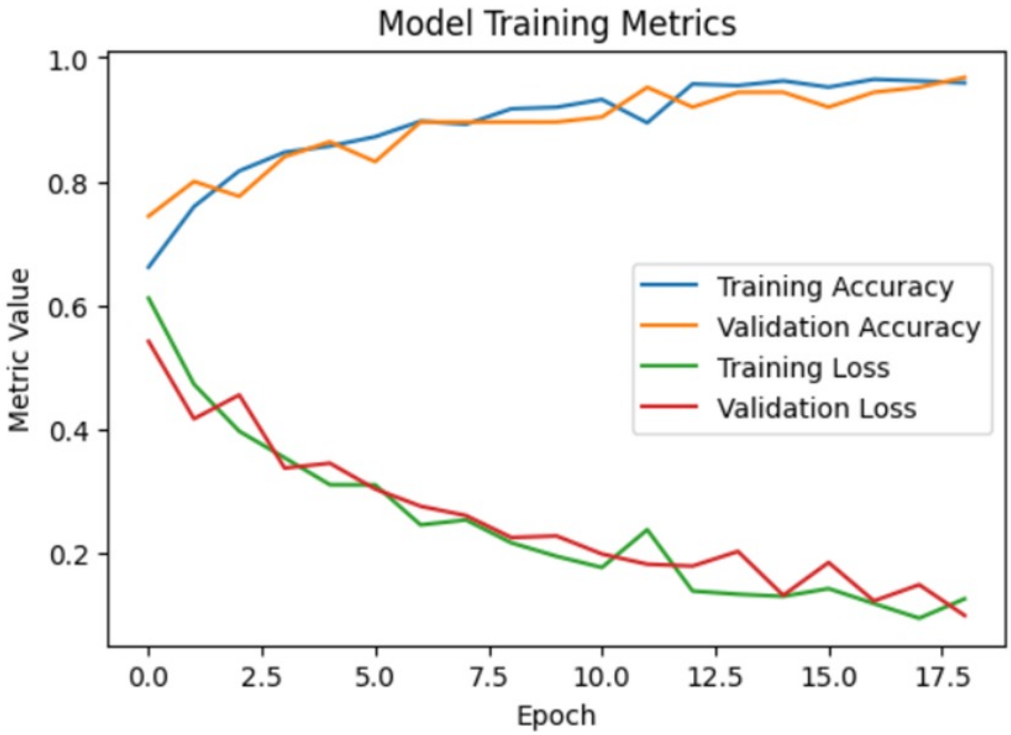
37'deki çıktı, eğitim süreci boyunca her bir epok için kayıp ve doğruluk ölçümlerini ve doğrulama seti için kayıp ve doğruluk ölçümlerini göstermektedir. Her bir dönemde elde edilen kayıp ve doğruluk değerleri modelin eğitim performansını yansıtmaktadır. Örneğin, "Epoch 17/20" satırı 17. epoch'un eğitim aşamasını temsil etmektedir. Bu dönemin sonunda eğitim setinde 0,1185 kayıp ve 0,9649 doğruluk elde edilmiştir. Aynı şekilde doğrulama kümesi için de 0,1228 kayıp ve 0,9440 doğruluk elde edilmiştir. Bu çıktı, eğitim sürecinin ilerleyişini izlemek ve modelin performansını değerlendirmek için kullanılmıştır. Kayıp değerleri ne kadar düşük ve doğruluk değerleri ne kadar yüksek olursa modelin performansının o kadar iyi olduğu söylenebilir.

```
50/50 [==============================] - 0s 4ms/step - loss: 0.1336 - accuracy: 0.9549 -
precision: 0.9588 - recall: 0.9490 - auc: 0.9905 - val_loss: 0.2027 - val_accuracy: 0.9440 - val_precision:
0.9104 - val_recall: 0.9839 - val_auc: 0.9849
Epoch 15/20
45/50 [============================>...] - ETA: 0s - loss: 0.1288 - accuracy: 0.9638 - precision:
0.9531 - recall: 0.9786 - auc: 0.9900
50/50 [==============================] - 0s 6ms/step - loss: 0.1302 - accuracy: 0.9624 -
precision: 0.9479 - recall: 0.9804 - auc: 0.9896 - val_loss: 0.1318 - val_accuracy: 0.9440 - val_precision:
0.9231 - val_recall: 0.9677 - val_auc: 0.9910
Epoch 16/20
41/50 [=========================>.....] - ETA: 0s - loss: 0.1359 - accuracy: 0.9572 - precision:
0.9588 - recall: 0.9588 - auc: 0.9896
Epoch 16: val_accuracy did not improve from 0.95200
50/50 [==============================] - 0s 6ms/step - loss: 0.1427 - accuracy: 0.9523 -
precision: 0.9455 - recall: 0.9598 - auc: 0.9881 - val_loss: 0.1848 - val_accuracy: 0.9200 - val_precision:
0.9483 - val_recall: 0.8871 - val_auc: 0.9869
Epoch 17/20
45/50 [============================>...] - ETA: 0s - loss: 0.1263 - accuracy: 0.9611 - precision:
0.9514 - recall: 0.9724 - auc: 0.9908
50/50 [==============================] - 0s 7ms/step - loss: 0.1185 - accuracy: 0.9649 -
precision: 0.9550 - recall: 0.9745 - auc: 0.9921 - val_loss: 0.1228 - val_accuracy: 0.9440 - val_precision:
0.9508 - val_recall: 0.9355 - val_auc: 0.9951
Epoch 18/20
44/50 [=========================>....] - ETA: 0s - loss: 0.1037 - accuracy: 0.9573 - precision:
0.9560 - recall: 0.9613 - auc: 0.9957
50/50 [==============================] - 0s 7ms/step - loss: 0.0948 - accuracy: 0.9624 -
precision: 0.9615 - recall: 0.9662 - auc: 0.9966 - val_loss: 0.1489 - val_accuracy: 0.9520 - val_precision:
0.9242 - val_recall: 0.9839 - val_auc: 0.9913
Epoch 19/20
36/50 [======================>.........] - ETA: 0s - loss: 0.1259 - accuracy: 0.9583 - precision: 0.9448
- recall: 0.9716 - auc: 0.9900
50/50 [==============================] - 0s 6ms/step - loss: 0.1261 - accuracy: 0.9592 -
precision: 0.9444 - recall: 0.9745 - auc: 0.9898 - val_loss: 0.0993 - val_accuracy: 0.9680 - val_precision:
0.9677 - val_recall: 0.9677 - val_auc: 0.9959
```

Şekil 37: Dönem için Kayıp ve Doğruluk Ölçütleri

Şekil 38, eğitim kaybı ve doğrulama kaybı ölçümlerinin epok sayısına göre nasıl değiştiğini gösteren bir grafiktir. Bu grafik, modelin eğitim sürecindeki performansını değerlendirmek için önemlidir. İdeal olarak, daha az eğitim kaybı ve daha az doğrulama kaybı beklenir. Şekil 39'daki grafik modelimizin iyi performans gösterdiğini doğrulamaktadır.

Şekil 38 ayrıca eğitim doğruluğu ve doğrulama doğruluğu metriklerinin eğitim sürecindeki epok sayısına bağlı olarak nasıl değiştiğini gösteren bir grafik çizmektedir. Bu grafik modelin doğruluk performansını değerlendirmek için kullanılır. İdeal olarak, eğitim doğruluğu artacak ve doğrulama doğruluğu yükselecektir. Şekil 40'daki grafik modelimizin iyi performans gösterdiğini doğrulamaktadır.



Şekil 38: Eğitim Metrikleri, Doğrulama Kaybı ve Eğitim, Doğrulama Doğruluğu Grafikleri

Şekil 40'daki çıktı, yüklenen modelin eğitim veri kümesi üzerindeki kayıp ve doğruluk puanlarını temsil etmektedir. "Kayıp 0.09164224565029144" ifadesi, modelin eğitim veri seti üzerinde elde ettiği kayıp puanını ifade eder. Daha düşük bir kayıp puanı, modelin tahminlerinin gerçek etiketlere daha yakın olduğunu gösterir. Modelimiz düşük bir kayıp puanına sahiptir. "Accuracy: 0.9599999785423279" ifadesi, modelin eğitim veri seti üzerinde elde ettiği doğruluk puanını temsil eder. Doğruluk puanı, modelin doğru tahminlerinin yüzdesini temsil eder. Bu durumda modelin eğitim veri setinde %95,99 doğruluk oranına sahip olduğu anlaşılmaktadır. Modelimiz yüksek bir doğruluk puanına sahiptir.

```
12/12 [==============================] - 0s 3ms/step - loss: 0.0916 - accuracy: 0.9600 -
precision: 0.9779 - recall: 0.9415 - auc: 0.9976
[0.09164224565029144,
 0.9599999785423279,
 0.9779005646705627,
 0.9414893388748169,
 0.997582197189331]
```

Şekil 39: Eğitim Değerlendirme Kaybı ve Değerlendirme Doğruluğu

Şekil 40'daki çıktı, yüklenen modelin test veri kümesi üzerindeki kayıp ve doğruluk puanlarını temsil etmektedir. "Loss 0.0993197113275528" ifadesi, modelin test veri seti üzerinde elde ettiği kayıp puanını ifade etmektedir. Daha düşük bir kayıp puanı, modelin tahminlerinin gerçek etiketlere daha yakın olduğunu gösterir. Modelimiz düşük bir kayıp puanına sahiptir. "Accuracy: 0.9679999947547913" ifadesi, modelin test veri seti üzerinde elde ettiği doğruluk puanını temsil eder. Doğruluk puanı, modelin doğru tahminlerinin yüzdesini temsil eder. Bu durumda modelin test veri setinde %96,79 doğruluk oranına sahip olduğu anlaşılmaktadır. Modelimiz yüksek bir doğruluk puanına sahiptir.

{'precision': 0.9677419066429138, 'remember': 0.9677419066429138,'auc': 0.995903730-392456 } kısmı tahminlerin ne kadar doğru olduğunu, kaç gerçek pozitifin doğru tahmin edildiğini ve modelin o sınıf için ne kadar iyi performans gösterdiğini gösterir. Elde edilen değerlendirme değerleri yüzde 90'ın üzerindedir. Bu değerlerin yüzde 90'ın üzerinde olması modelimizin iyi çalıştığını göstermektedir.

```
4/4 [==============================] - 0s 8ms/step - loss: 0.0993 - accuracy: 0.9680 - precision:
0.9677 - recall: 0.9677 - auc: 0.9959
[0.0993197113275528,
 0.9679999947547913,
 0.9677419066429138,
 0.9677419066429138,
 0.995903730392456]
```

Şekil 40: Test Değerlendirme Kaybı ve Değerlendirme Doğruluğu

5 SONUÇLAR VE ÖNERİLER

5.1 Sonuçlar

Veri Temizleme: Veri seti üzerinde gereksiz sütunların kaldırılması ve eksik verilerin doldurulması işlemleri gerçekleştirilmiştir. Bu, modelin daha doğru sonuçlar vermesi için gereklidir.

Veri Normalizasyonu: Veriler, Min-Max normalizasyonu kullanılarak ölçeklendirilmiştir. Bu işlem, modelin performansını artırmak için önemli bir adımdır.

Anormallik Algılama: Verilerdeki aşırı uç değerler tespit edilip temizlenmiştir. Bu, modelin genel performansını ve doğruluğunu artırmak için kritik öneme sahiptir.

Korelasyon Analizi: Pearson korelasyon katsayıları kullanılarak özellikler arasındaki ilişkiler değerlendirilmiştir. Bu analiz, hangi özelliklerin model için daha önemli olduğunu belirlemeye yardımcı olmuştur.

Özellik Seçimi: Özellik seçimi sürecinde Yinelemeli Korelasyon Önem Dereceleri (PER) ve Minimum Önem Dereceleri (MIF) yöntemleri kullanılarak en anlamlı 10 özellik belirlenmiştir. Bu yöntemler, modelin daha az karmaşık ve daha açıklayıcı olmasını sağlar. Özellik seçimi, modelin sadece en önemli ve etkili özelliklerle çalışarak gereksiz hesaplama yükünü azaltır ve performansını artırır.

5.2 Model Performans Değerlendirmesi

Destek Vektör Makineleri (SVM): SVM modeli,%96.0 doğruluk oranı ile yüksek performans göstermiştir. SVM'nin doğruluk, kesinlik, geri çağırma ve F1 skoru gibi metriklerde iyi sonuçlar verdiği gözlemlenmiştir.

Yapay Sinir Ağları (ANN): ANN modeli de benzer şekilde yüksek doğruluk oranlarına sahiptir. Modelin %95.99 doğruluk oranı ve düşük kayıp puanları, etkin bir performans sergilediğini göstermektedir.

Çapraz Doğrulama: Modeller, çapraz doğrulama teknikleri kullanılarak test edilmiştir. Bu, modelin genelleme yeteneğini artırmak için önemli bir adımdır.

5.3 Sonuçların Genel Değerlendirmesi

Projenin sonuçları, makine öğrenmesi algoritmalarının meme kanseri tahmininde etkili olduğunu göstermektedir. Hem SVM hem de ANN modelleri, yüksek doğruluk oranları ve düşük hata oranları ile başarılı bir performans sergilemiştir. Bu modeller, erken teşhis için doktorlar tarafından kullanılabilir potansiyeldedir .

5.4 Öneriler

Veri Setinin Genişletilmesi: Kullanılan veri seti, 116 katılımcıdan elde edilen verilerden oluşmaktadır. Daha geniş bir veri seti kullanarak modelin genelleme yeteneği artırılabilir. Özellikle farklı demografik gruplardan veri toplanması, modelin doğruluğunu ve güvenilirliğini artıracaktır.

Farklı Algoritmaların Kullanılması: SVM ve ANN dışında farklı makine öğrenmesi algoritmalarının da denenmesi önerilir. Örneğin, karar ağaçları, rastgele ormanlar ve k-en yakın komşu algoritmaları gibi yöntemler de meme kanseri tahmini için değerlendirilebilir. Bu, en etkili algoritmanın belirlenmesine yardımcı olacaktır.

Model Optimizasyonu: Kullanılan algoritmaların hiperparametre optimizasyonu yapılarak daha iyi sonuçlar elde edilebilir. Bu amaçla, grid search veya random search gibi

yöntemler kullanılabilir. Hiperparametre optimizasyonu, modelin performansını maksimize etmek için kritik öneme sahiptir.

Model Performansının Sürekli İzlenmesi: Modelin performansını sürekli izlemek ve gerektiğinde güncellemeler yapmak önemlidir. Veri setinde değişiklikler olduğunda modeli yeniden eğitmek ve test etmek gereklidir. Bu, modelin güncel ve doğru kalmasını sağlar.

Klinik Uygulamalar: Geliştirilen modelin klinik ortamlarda kullanılabilirliği test edilmelidir. Gerçek dünyadaki uygulamalarda modelin performansını değerlendirmek, güvenilirliğini ve etkinliğini artıracaktır. Klinik testler, modelin sağlık pratisyenleri tarafından benimsenmesi için gereklidir.

Daha Fazla Araştırma ve Deneme: Özellik seçimi ve hiperparametre optimizasyonu için daha fazla araştırma yapın ve çeşitli yöntemleri deneyin. Farklı yöntemler ve teknikler, modelin performansını farklı şekillerde etkileyebilir. Bu nedenle, çeşitli yaklaşımlar deneyerek en iyi sonucu veren yöntemleri belirlemek önemlidir.

Projenin kapsamlı değerlendirmesi, makine öğrenimi tekniklerinin tıp alanında önemli bir rol oynayabileceğini ve meme kanseri gibi önemli sağlık sorunlarının çözümünde potansiyel bir araç olarak kullanılabileceğini göstermektedir .

6 EKLER

```
import numpy as np
import pandas as pd
from scipy import stats
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, confusion_matrix,
    classification_report
import tensorflow as tf
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Dense, Dropout
from tensorflow.keras.callbacks import ModelCheckpoint

# Veri seti okunur
data = pd.read_csv('./dataR2.csv')

# Veri dagilim grafikleri
for column in data.columns:
    plt.figure(figsize=(4, 2))
    sns.histplot(data[column], bins=20, kde=True)
    plt.title(f'Distribution of {column}')
    plt.xlabel(column)
    plt.ylabel('Frequency')
    plt.show()

# Eksik degerler kontrol edilir
data.isnull().sum()

# Yinelemeler kaldirilir
data = data.drop_duplicates()

# Veri seti normallestirilir
for column in data.columns:
    data[column] = (data[column] - data[column].min()) / (data[column].
```

```
    max() - data[column].min())
35
36 # Hedef degisken ayarlanir
37 y = data['Classification']
38 X = data.drop(['Classification'], axis=1)
39
40 # Veri seti egitim ve test setlerine ayrilir
41 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size
    =0.1, stratify=y)
42
43 # SVC modeli kurulur ve egitilir
44 svm_model = SVC(kernel='poly', gamma=8)
45 svm_model.fit(X_train, y_train)
46
47 # Modelin performansi degerlendirilir
48 predictions_train = svm_model.predict(X_train)
49 accuracy_train = accuracy_score(y_train, predictions_train)
50 confusion_matrix_train = confusion_matrix(y_train, predictions_train)
51 classification_report_train = classification_report(y_train,
    predictions_train)
52
53 predictions_test = svm_model.predict(X_test)
54 accuracy_test = accuracy_score(y_test, predictions_test)
55 confusion_matrix_test = confusion_matrix(y_test, predictions_test)
56 classification_report_test = classification_report(y_test,
    predictions_test)
57
58 # Over-sampling yapilir
59 df_class_0 = data[data['Classification'] == 0]
60 df_class_1 = data[data['Classification'] == 1]
61
62 df_class_1_over = df_class_1.sample(250, replace=True)
63 df_class_0_over = df_class_0.sample(250, replace=True)
64 df_test_over = pd.concat([df_class_0_over, df_class_1_over], axis=0)
65
66 y_over = df_test_over['Classification']
67 X_over = df_test_over.drop(['Classification'], axis=1)
```

```
68
69 # Over-sampling sonrasi veri seti egitim ve test setlerine ayrilir
70 X_train_over, X_test_over, y_train_over, y_test_over = train_test_split
       (X_over, y_over, test_size=0.25, random_state=0, shuffle=True,
       stratify=y_over)
71
72 # SVC modeli over-sampling verisi ile kurulur ve egitilir
73 svc_s_model = SVC(kernel='rbf', gamma=8)
74 svc_s_model.fit(X_train_over, y_train_over)
75
76 # Modelin performansi degerlendirilir
77 predictions_train_over = svc_s_model.predict(X_train_over)
78 accuracy_train_over = accuracy_score(y_train_over,
       predictions_train_over)
79 confusion_matrix_train_over = confusion_matrix(y_train_over,
       predictions_train_over)
80 classification_report_train_over = classification_report(y_train_over,
       predictions_train_over)
81
82 predictions_test_over = svc_s_model.predict(X_test_over)
83 accuracy_test_over = accuracy_score(y_test_over, predictions_test_over)
84 confusion_matrix_test_over = confusion_matrix(y_test_over,
       predictions_test_over)
85 classification_report_test_over = classification_report(y_test_over,
       predictions_test_over)
86
87 # Yapay sinir agi modeli kurulur ve egitilir.
88 dl_model = Sequential()
89 dl_model.add(Dense(256, activation='relu', input_shape=([9])))
90 dl_model.add(Dense(512, activation='relu'))
91 dl_model.add(Dense(1, activation='sigmoid'))
92 dl_model.compile(optimizer='adam', loss='binary_crossentropy', metrics
       =['accuracy', 'Precision', 'Recall', 'AUC'])
93
94 # Modelin performansi degerlendirilir
95 history = dl_model.fit(X_train_over, y_train_over, epochs=20,
       steps_per_epoch=50, validation_data=(X_test_over, y_test_over),
```

```
    callbacks=[checkpoint])

# Model performansi grafik olarak gosterilir.
plt.figure(figsize=(6, 4))
plt.plot(history.history['accuracy'], label='Training Accuracy')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
plt.plot(history.history['loss'], label='Training Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.title('Model Training Metrics')
plt.xlabel('Epoch')
plt.ylabel('Metric Value')
plt.legend()
plt.show()

# Modelin performansi tekrar degerlendirilir
dl_model.evaluate(X_train_over, y_train_over)
dl_model.evaluate(X_test_over, y_test_over)
```

KAYNAKLAR

[1] World Health Organization. (2021). Cancer. Retrieved November 11, 2022, from https://www.who.int/en/news-room/factsheets/detail/cancer.

[2] Bray, F., Ferlay, J., Soerjomataram, I., Siegel, R. L., Torre, L. A., & Jemal, A. (2018). Global cancer statistics 2018: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. CA: a cancer journal for clinicians, 68(6), 394-424.

[3] Ferlay, J., Colombet, M., Soerjomataram, I., Mathers, C., Parkin, D. M., Piñeros, M., et al. (2019). Estimating the global cancer incidence and mortality in 2018: GLOBOCAN sources and methods. International Journal of Cancer. Wiley-Liss Inc. https://doi.org/10.1002/ijc.31937.

[4] Ozmen, V., Boylu, S., Ok, E., Canturk, N. Z., Celik, V., Kapkac, M., et al. (2015). Factors affecting breast cancer treatment delay in Turkey: a study from Turkish Federation of Breast Diseases Societies. European Journal of Public Health, 25, 9–14. doi:10.1093/eurpub/cku086.

[5] Papageorgiou, E. I., Subramanian, J., Karmegam, A., & Papandrianos, N. (2015). A risk management model for familial breast cancer: A new application using Fuzzy Cognitive Map method. Computer Methods and Programs in Biomedicine, 122(2), 123–135.

[6] Drageset, S., Lindstrøm, T. C., Giske, T., & Underlid, K. (2011). Being in suspense: Women's experiences awaiting breast cancer surgery. Journal of Advanced Nursing, 1-11.

[7] Alpteker, H., & Avcı, A. (2010). Kırsal alandaki kadınların meme kanseri bilgisi ve kendi kendine meme muayenesi uygulama durumlarının belirlenmesi. Meme Sağlığı Dergisi, (2):74-79.

[8] Sevli, O. (2019). Göğüs Kanseri Teşhisinde Farklı Makine Öğrenmesi Tekniklerinin Performans Karşılaştırması. Avrupa Bilim ve Teknoloji Dergisi, (16), 176-185.

[9] Eyupoglu, C. (2018). Breast cancer classification using k-nearest neighbors algorithm. The Online Journal of Science and Technology, 8(3), 29-34.

[10] Juanpere, S., Perez, E., Huc, O., Motos, N., Pont, J., & Pedraza, S. (2011). Imaging of breast implants—a pictorial review. Insights Into Imaging, 2, 653–670. doi: 10.1007/s13244-011-0122-3.

[11] Ahmad, F., & Yusoff, N. (2013). Classifying breast cancer types based on fine needle aspiration biopsy data using random forest classifier. 13th International Conference on in Intelligent Systems Design and Applications (ISDA), IEEE, 121-5.

[12] Kolay, N., & Erdoğmuş, P. (2017). The classification of breast cancer with Machine Learning Techniques. In Electric.

[13] Deniz, A., Kiziloz, H. E., Dokeroglu, T., & Cosar, A. (2017). Robust multiobjective evolutionary feature subset selection algorithm for binary classification using machine learning techniques. Neurocomputing, 241, 128–146. https://doi.org/10.1016/J.NEUCOM.2017.02.033.

[14] Gillies, R. J., Kinahan, P. E., & Hricak, H. (2016). Radiomics: Images Are More than Pictures, They Are Data. Radiology, 278(2), 563–577. https://doi.org/10.1148/radiol.2015151169.

[15] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Vol. 2016-Decem, pp. 770–778).

[16] Aydıntuğ, S. (2004). Meme kanserinde erken tanı. Sted, 13(6), 226-228.

[17] Obese, H. (1998). "Body mass index (BMI)." Obes Res 6.2: 51S-209S.

[18] Saxena, N. K., & Sharma, D. (2013). Multifaceted Leptin Network: The Molecular Connection Between Obesity and Breast Cancer. Journal of Mammary Gland Biology and Neoplasia, 18(3-4), 309-320.

[19] Savaş, H. B., & Gültekin, F. (2017). İnsülin direnci ve klinik önemi. SDÜ Tıp Fakültesi Dergisi, 24(3), 116-125.

[20] Cruz, J. A., & Wishart, D. S. (2006). Applications of machine learning in cancer prediction and prognosis. Cancer Informatics, 2.

[21] Du, K. L., & Swamy, M. N. (2019). Neural Networks and Statistical Learning (2nd ed.). Springer Science & Business Media.

[22] Kshirsagar, P., & Rathod, N. (2012). “Artificial neural network,” International Journal of Computer Applications, 2012.

[23] Jadeja, Y., & Modi, K. (2012). Cloud Computing—Concepts, Architecture and Challenges. 2012 International Conference on Computing, Electronics and Electrical Technologies (ICCEET).

[24] Gupta, N. (2013). “Artificial neural network,” Network and Complex Systems, 3(1), 24-28.

[25] Mandhala, V. N., Sujatha, V., & Devi, B. R. (2014). “Scene classification using support vector machines.” IEEE International Conference on Advanced Communications, Control and Computing Technologies, 1807-1810.

[26] Lai, W. C., Huang, P. H., Lee, Y. J., & Chiang, A. (2015). “A distributed ensemble scheme for nonlinear support vector machine,” IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), pp. 1-6.

[27] Yang, Z., & Yang, Z. (2014). Comprehensive Biomedical Physics. Karolinska Institute, Stockholm, Sweden: Elsevier. ISBN 978-0-444-53633-4.

[28] Brown, N., Kaloustian, S., & Roeckle, M. (2007). Monitoring of Open Pit Mines Using Combined GNSS Satellite Receivers and Robotic Total Stations. In Proceedings of the 2007 International Symposium on Rock Slope Stability in Open Pit Mining and Civil Engineering, Perth, Australia, 12–14 September 2007; Potvin, Y., Ed.; Australian Centre for Geomechanics: Crawley, Australia, pp. 417–429.

[29] Zadka, M., & van Rossum, G. (2001). "PEP 238 – Changing the Division Operator". Python Enhancement Proposals. Python Software Foundation. Retrieved March 11, 2001.

[30] Bengio, Y., LeCun, Y., & Hinton, G. (2015). "Deep Learning". Nature, 521 (7553), 436-444. Bibcode:2015Natur.521..436L. doi:10.1038/nature14539.

[31] Schreinemachers, P., Simmons, E. B., & Wopereis, M. C. (2018). Tapping the economic and nutritional power of vegetables. Global Food Security, 16, 36–45.

[32] Kaggle. https://www.kaggle.com/.

[33] Diagrams.net. https://app.diagrams.net/#G1WWJdQYHG2N0lFrZtBP06ujdP45Lgschz.

[34] Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by Simulated Annealing. Science, 220, 671–680.

[35] Yang, Y., & Wu, Q. M. J. (2016). Extreme Learning Machine With Subnetwork Hidden Nodes for Regression and Classification. IEEE Transactions on Cybernetics, 46(12), 2885-2898.

[36] Suguna, N., & Thanushkodi, K. (2010). "An improved k-nearest neighbor classification using genetic algorithm," International Journal of Computer Science Issues, 7(2), 18-21.

[37] Chandrashekar, G., & Sahin, F. (2014). A survey on feature selection methods. Computers & Electrical Engineering, 40(1), 16–28. https://doi.org/10.1016/j.compeleceng.2013.11.024.

[38] Gierach, G. L., Li, H., Loud, J. T., Greene, M. H., Chow, C. K., Lan, L., et al. (2014). Relationships between computer-extracted mammographic texture pattern features and BRCA1/2 mutation status: A cross-sectional study. Breast Cancer Research, 16(4), 424. https://doi.org/10.1186/s13058-014-0424-8.

[39] Herent, P., Schmauch, B., Jehanno, P., Dehaene, O., Saillard, C., Balleyguier, C., et al. (2019). Detection and characterization of MRI breast lesions using deep learning. Diagnostic and Interventional Imaging, 100(4), 219–225. https://doi.org/10.1016/j.diii.2019.02.008.

[40] Kira, K., & Rendell, L. A. (1992). A Practical Approach to Feature Selection. In Machine Learning Proceedings 1992 (pp. 249–256). Elsevier.

[41] Behnel, S., Bradshaw, R., Citro, C., Dalcin, L., Seljebotn, D. S., & Smith, K. (2011). Cython: The best of both worlds. Computing in Science & Engineering, 13(2), 31-39.

[42] Loffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.

[43] Ansar, L. A., Arifin, D. Y., & Dengo, Y. J. (2017). The influence of school culture on the performance of high school English teachers in Gorontalo province. International Journal of Education Research, 5(10), 35-47.

[44] Chidambaram, S., & Srinivasagan, K. G. (2019). Performance evaluation of support vector machine classification approaches in data mining. Cluster Computing, 22(Suppl 1), 189–196.

[45] Shahverdi, V. (2023). Machine Learning and Algebraic Geometry, 18.

[46] Sobha, P., & Latifi, S. (2023). A Survey of the Machine Learning Models for Forest Fire Prediction and Detection. International Journal of Communications, Network and System Sciences, 16, 131-150.

[47] Fachrurrozi, S., Muljono, G. F. Shidik, A. Z. Fanani, Purwanto, & Zami, F. A. (2021). Increasing Accuracy of Support Vector Machine (SVM) By Applying N-Gram and Chi-Square Feature Selection for Text Classification. 2021 International Seminar on Application for Technology of Information and Communication (iSemantic).