T.C.

BİTLİS EREN ÜNİVERSİTESİ

MÜHENDİSLİK-MİMARLIK FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

Veri Madenciliği Projesi

PROJENİN ÖNERİSİNİN ADI

MEME KANSERİ TESPİTİ

HAZIRLAYAN

Ad Soyad

Gülay AKTAŞ

DANIŞMAN

Unvan, Ad, Soyad

Dr. Öğr. Üyesi Erdal AKIN

Ocak 2023

ÖZET

MEME KANSERİ TESPİTİ

SOYADI, Adı (Öğrenci)

AKTAŞ Gülay

Proje Raporı

Bitlis Eren Üniversitesi Mühendislik Mimarlık Fakültesi

Bilgisayar Mühendisliği Bölümü

Danışman: Dr. Öğr. Üyesi Erdal AKIN

Ocak 2023, 23 sayfa

**Özet**

Modern bilgisayarların daha iyi analizler ve tahminler yapmasına izin veren çeşitli makine öğrenimi teknikleri geliştirilmektedir. Makine öğrenimi teknikleri, başarılı sınıflandırma ve teşhis işlevleriyle, hastalıkların teşhisinde hızla tıp uzmanlarına yardımcı oluyor. Kanser teşhisinde kullanımı hızla artan makine öğrenimi sayesinde başarılı araştırmalar yapılabiliyor. Test sonuçlarını yorumlayarak meme kanserini teşhis etmek ve teşhis etmek uzman insan bilgisi gerektirir, ancak meme kanserini teşhis etmek için evrimsel makine öğrenimi teknikleri kullanılarak başarılı araştırmalar yapılmaktadır. Makine öğrenimi, bilgisayarların mevcut verilerden öğrenerek karmaşık ve büyük veri kümelerindeki kalıpları hızlı bir şekilde tanımlamasına olanak tanıyan bir yapay zekâ dalıdır. Bu yeteneği nedeniyle, makine öğrenimi kanserin, özellikle meme kanserinin tanı ve tedavisinde yaygın olarak kullanılmaktadır. Eğitim sonrasında Naive Bayes, Random forest, K-en yakın komşu ve DecisionTreeClassifier yöntemleri kullanılarak Karışıklık matrisi, f1 scorlarrı, precission ve recal değerleri hesaplandı. Her tekniğin başarısı karşılaştırıldı. Bu karşılaştırmada f1 scorları baz alındğında Random forest 0.91 doğrulukla en başarılı yöntem olduğunu ortaya koydu.[11]

**Anahtar kelimeler**: Meme Kanseri, Gögüs Kanseri, Kanser Tespiti

TEŞEKKÜR

Proje çalışması sırasında her türlü bilgi, teşvik ve deneyimleri ile yardımlarını esirgemeyen Dr. Öğr. Üyesi Erdal AKIN ‘a proje önerisi süresince her türlü maddi ve manevi destekleri ile göstermiş oldukları sabırdan dolayı teşekkür ederim.

İÇİNDEKİLER DİZİNİ

**Sayfa**

[ÖZET ii](#_Toc124330127)

[TEŞEKKÜR iii](#_Toc124330128)

[İÇİNDEKİLER DİZİNİ iv](#_Toc124330129)

[TABLOLAR LİSTESİ v](#_Toc124330130)

[ÇİZELGELER DİZİNİ vi](#_Toc124330131)

[ŞEKİLLER DİZİNİ vii](#_Toc124330132)

[KISALTMALAR DİZİNİ viii](#_Toc124330133)

[1. GİRİŞ 1](#_Toc124330134)

[2. MATERYAL VE YÖNTEM 2](#_Toc124330135)

[2.1 Veri Seti 2](#_Toc124330136)

[2.2 k-Karar Ağacı — DecisionTreeClassifier 3](#_Toc124330137)

[2.3 Rastgele Orman algoritması nedir? (Random Forest) 3](#_Toc124330138)

[2.3.1 Random Forest algoritması nasıl çalışır? 3](#_Toc124330139)

[2.4 k-En Yakın Komşuluk — k Nearest Neighbours 5](#_Toc124330140)

[2.5 Naive Bayes 6](#_Toc124330141)

[2.6 Matris Hatası Nedir? -Confusion Matrix 7](#_Toc124330142)

[2.6.1 Accuracy(Doğruluk): 7](#_Toc124330143)

[2.6.2 Precision(Kesinlik) : 7](#_Toc124330144)

[2.6.3 Recall(Duyarlılık) : 7](#_Toc124330145)

[2.6.4 F-measure: Precision 7](#_Toc124330146)

[3 BULGULAR 8](#_Toc124330147)

[4 SONUÇ 9](#_Toc124330148)

[5 KAYNAKLAR 10](#_Toc124330149)

[6 EKLER 11](#_Toc124330150)

[6.1 Kodlar 11](#_Toc124330151)

TABLOLAR LİSTESİ

[Tablo 1 Modellerin Sınıflandırma Başarıları 8](#_Toc124330152)

ÇİZELGELER DİZİNİ

Çizelge Sayfa

[Çizelge 2.1. Random Forrest bazı avantajları ve dezavantajları 5](#_Toc124330153)

ŞEKİLLER DİZİNİ

**Şekil Sayfa**

[Şekil 1Random Forest Algoritması [7] 4](#_Toc124330159)

[Şekil 2 KNN Örnek Çizim [5] 6](#_Toc124330160)

[Şekil 3 KNN Adımları [5] 6](#_Toc124330161)

KISALTMALAR DİZİNİ

SVM Destek Vektör Makineleri Support Vector Machines

KNN k-En Yakın Komşuluk k Nearest Neighbours

DP Doğru Pozitif True Positive

YP Yanlış Pozitif True Negative

DN Doğru Negatif False Positive

YN Yanlış Negatif False Negative

DT Karar Ağacı DecisionTreeClassifier

# 1. GİRİŞ

Meme kanseri her geçen gün artan bir sağlık sorunudur ve kadınlarda en sık görülen sağlık sorunudur. Batılı ülkelerde her 8- 9 kadına yaşamları boyunca meme kanseri teşhisi konulduğu gözlemlenmiştir. Bu oran daha büyüktür. Yaygın olması, erken dönemde tedavi edilebilmesi ve mevcut koşullarda tanı konulabilmesi meme kanserinin önemini artırmaktadır [1]. Meme kanseri karmaşık ve tek taraflı bir genetik hastalıktır. Kalıtsal, ailesel ve sporadik (random / random) olmak üzere üç grupta incelenir. Bir ailede en az. Kişide meme kanseri varsa buna ailesel meme kanseri denir. Meme kanseri ile bire bir ilişkisi olan BRCA1 ve BRCA2 genlerindeki mutasyonlar, kanserin ailesinde erken dönemde ortaya çıktığını ve ailesindeki erkeklerde meme kanserinin hastalığın kalıtsal olduğunu düşündürdüğünü gösterir. Kalıtsal meme kanseri olan kişilerde yaşam boyu meme kanseri gelişme riski çok yüksektir. Bu iki grubun dışında sporadik meme kanseri görülür, aile veya kalıtsal bağlantısı yoktur [2]. Meme kanserinin altında yatan genetik faktörlerin belirlenmesi bu hastalığın tanı ve tedavisine önemli katkılar sağlayabilir. Son yıllarda DNA analizinde büyük birgelişme yaşanıyor. On binlerce genin etkileşimini aynı anda ölçebilen, mikroarray teknolojisi olarak da bilinen gen çipleri bunlardan biri. Mikroçip teknolojisi sayesinde, hastalıklarının arkasındaki genetik faktörleri belirlemek ve hastalığın erken teşhisinzsağlamak mümkündür. Microarray, aynı anda onbinlerce gen ifade verisi ölçebilir. Bununla birlikte, gen ifade verileri küçük örneklerdir ve çoğu gürültü olarak adlandırılan. Gen için on binlerce ilgisiz veri içerir. Ayrıca fonksiyonlarının olması veri hacmini ve analiz ve sınıflandırma problemini arttırmaktadır. Bu durum mikro dizileri ile çalışabilmek için öncelikle boyut küçültme ve gen seçimi gibi işlemlerin yapılmasını gerekli kılmaktadır. Bu nedenle, özellik seçimi genetik analizde kritik öneme sahiptir [9]. M. Bilen ve arkadaşları çalışmalarında mikrodizi verisine boyut küçültme uygulamışlardır. Yeni veri setinin genetik algoritma ve yapay sinir ağı ile sınıflandırılması sonucunda doğruluk derecesinin olduğu bulunmuştur [4].

Bu çalışma, her biri 30 özellik içeren 567 meme kanseri veri örneğini 5 farklı makine öğrenimi tekniği kullanarak sınıflandırdı ve modelin performansını karşılaştırdı. Aşağıdaki bölümlerde materyal ve yöntem, sonuçlar ve sonuçlar yer almaktadır.

# MATERYAL VE YÖNTEM

Kanser, hücreler kontrolsüz bir şekilde bölündüğünde ve tümör adı verilen kitleler oluşturduğunda ortaya çıkar. Tümörler iyi huylu (benign) ve kötü huylu (malignant) olabilir. Kötü huylu tümörler hızla büyür, çevre dokuları istila eder ve hasara neden olur. Meme dokusundaki anormallikler, meme şekli ve cilt rengindeki değişiklikler meme kanserine işaret edebilir. Tüm kanser türlerinde olduğu gibi meme kanserinde de erken teşhis çok önemlidir. Bu çalışmada, Wisconsin Üniversitesi Hastanesi'nden William H. Wolberg, Dr. Veri seti, eğitim setinin bir test setine bölünmesi ve dört farklı makine öğrenme modeline sınıflandırılması ile test edilmiştir. DecisionTreeClassifier, Naive Bayes (NB), Rastgele Orman (Random Forest - RF), ve K En Yakın Komşu (K En Yakın Komşu - KNN) performansları karşılaştırılmıştır. Model üretimi ve testi, Python programlama dili kullanılarak gerçekleştirilmiştir.

### 2.1 Veri Seti

Kullanılan veritabanı 567 örnek içermektedir. Her veri örneğinin toplam 33 özelliği vardır ve bunların 30'u tanımlayıcı özellikler ve bir tanı kategorisidir. 30 tanımlayıcı özellik meme görüntülerinin sayısallaştırılmasından elde edilen dokunun çapı, şekli, pürüzsüzlüğü ve alanı gibi verilerden oluşmaktadır. 30 özelliğin 10'u doğrudan tümör hücresinin çekirdeği üzerinden ölçülür ve 20'si bunlardan hesaplanan sayısal değerlerdir. Doğrudan ölçülen 10 özelliği şunlardır:

1) Yarıçap

2) Doku

3) Çap

4) Alan

5) Pürüzsüzlük

6) Yoğunluk

7) İçbükeylik

8) İçbükey nokta sayısı

9) Simetri

10) Fraktal boyut

Diğer fonksiyonlar, bu özelliklerden türetilen ortalamalar, standart hatalar, en kötü ve maksimum değerlerden oluşur. Bu değere dayalı olarak, tümörün iyi huylu veya kötü huylu olduğunu gösteren B (iyi huylu) ve M (kötü huylu) ile gösterilen bir teşhis sınıfı vardır. 567 verinin sınıf dağılımı 356 iyi huylu ve 211 kötü huyludur.

### 2.2 k-Karar Ağacı — DecisionTreeClassifier

DecisionTreeClassifier, scikit-learn kütüphanesinde yer alan bir sınıflandırma ağacı (decision tree) yöntemi. Bu yöntem, veri kümesi içinde veri noktalarını sınıflandırmak için karar ağacı oluşturur. Karar ağacı, veri noktalarının özellikleri ve sınıf etiketleri gibi verileri kullanarak veri noktalarını sınıflandırmak için kuralları belirler.

DecisionTreeClassifier sınıfı, aşağıdaki özelliklere sahiptir:

criterion: Karar ağacının ne şekilde bölünmesi gerektiğini belirlemek için kullanılan ölçüt. İki seçenek vardır: "gini" ve "entropy".

max\_depth: Karar ağacının maksimum derinliği. Bu, ağacın kaç katmanlı olduğunu belirler.

min\_samples\_split: Bir düğümün bölünmesi için gerekli olan minimum örnek sayısı.

min\_samples\_leaf: Bir düğümün sonuncu katmanına ulaşması için gerekli olan minimum örnek sayısı.

max\_features: Bir düğüm bölünürken kullanılacak özelliklerin maksimum sayısı. [8]

### Rastgele Orman algoritması nedir? (Random Forest)

Rastgele Orman algoritması denetimli bir sınıflandırma algoritmasıdır. Rastgele olarak bir orman yaratıyor. Algoritmadaki ağaç sayısı ve elde edebileceği sonuç arasında doğrudan bir ilişki bulunmaktadır. Ağaç sayısı arttıkça kesin bir sonuç elde ederiz. Basit kelimelerle söylemek gerekirse: Rastgele orman, birden fazla karar ağacını oluşturur ve daha doğru ve istikrarlı bir tahmin elde etmek için onları birleştirir. [7]

#### 2.3.1 Random Forest algoritması nasıl çalışır?

Rastgele orman algoritması aşağıdaki adımları tamamlayarak çalışır:

Adım 1 : Algoritma, sağlanan veri kümesinden rastgele örnekler seçer.

Adım 2: Algoritma, seçilen her örnek için bir karar ağacı oluşturacaktır. Ardından oluşturulan her karar ağacından bir tahmin sonucu alacaktır.

Adım 3: Daha sonra tahmin edilen her sonuç için seslendirme yapılacaktır. Bir sınıflandırma problemi için modu , regresyon problemi için ortalama kullanacaktır .

Adım 4 : Son olarak, algoritma, son tahmin olarak en çok oylanan tahmin sonucunu seçecektir.

Örnek: Birden fazla meyve görüntüsü içeren bir veri kümesi olduğunu varsayalım. Böylece, bu veri kümesi Random orman sınıflandırıcısına verilir. Veri kümesi alt kümelere bölünür ve her bir karar ağacına verilir. Eğitim aşamasında, her karar ağacı bir tahmin sonucu üretir ve yeni bir veri noktası oluştuğunda, sonuçların çoğuna dayalı olarak Random Forest sınıflandırıcı nihai kararı tahmin eder. Aşağıdaki görüntüyü göz önünde bulundurun. [7]



Şekil Random Forest Algoritması [7]

Çizelge 2.. Random Forrest bazı avantajları ve dezavantajları

|  |  |
| --- | --- |
| **Random Forrest Bazı Avantajları** | **Random Forrest Bazı Dezavantajları** |
| Random Forest, hem Sınıflandırma hem  de Regresyon görevlerini gerçekleştirebilir. | Rastgele orman hem sınıflandırma hem de regresyon görevleri için kullanılabilse de, Regresyon görevleri için daha uygun değildir. |
| Yüksek boyutluluk ile büyük veri  kümelerini işleme yeteneğine sahiptir. |
| Modelin doğruluğunu artırır ve  fazla uyum sorununu önler. |
| Random Forest, hem Sınıflandırma hem  de Regresyon görevlerini gerçekleştirebilir. |

### 2.4 k-En Yakın Komşuluk — k Nearest Neighbours

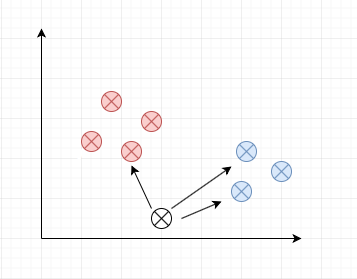
K-en yakın komşuluk algoritması, denetimli öğrenme algoritmalarındandır. Sınıflandırma ve regresyon problemlerinde kullanılır. Ayrıca endüstride genellikle sınıflandırma sorunlarının çözümünde kullanılır.

KNN algoritması , sınıfları belli olan bir örnek kümesindeki veriler kullanılarak yapılmaktadır. Örnek veri setine girecek olan yeni verinin, mevcut verilere göre uzaklığı hesaplanıp, k sayıda yakın komşuluğuna bakılır. Uzaklık hesapları için genelde 3 tip uzaklık fonksiyonu kullanılmaktadır:

“Euclidean” Uzaklık

“Manhattan” Uzaklık

“Minkowski” Uzaklığı’dır.[5]



Şekil KNN Örnek Çizim [5]



Şekil KNN Adımları [5]

### 2.5 Naive Bayes

Naive Bayes sınıflandırıcısı, İngiliz matematikçi Thomas Bayes Co. Sunulan eşitliğe dayanmaktadır [3].

P(G|X) = (P(X|G) P(G))/(P(X) )

Formülde P(G|X), G olayının verilen X olayına göre olma olasılığıdır. P(X|G) G olayı meydana geldiğinde X olayının meydana gelme olasılığı. P(G) ve P(X) ise G ve X olayların apriori olasılıklarıdır.

### 2.6 Matris Hatası Nedir? -Confusion Matrix

Confusion matrix ( hata matrisi), veri setindeki var olan durum ile sınıflandırma modelimizin doğru ve yanlış tahminlerinin sayısını tablo olarak göstermektedir.

Gerçek pozitif (True Positive – TP) – Olumlu tahmin ettiniz ve bu doğru

Gerçek negatif (True Negative – TN) – Olumsuz tahmin ettiniz ve bu doğru

Yanlış pozitif (False Positive – FP) – Olumlu tahmin ettiniz ve bu yanlış

Yanlış negatif (False Negative – FN) – Olumsuz tahmin ettiniz ve bu yanlış

GERÇEK POZİTİFLER = TP + FN

GERÇEK NEGATİFLER = TN + FP

TOPLAM = TP + TN + FP + FN [6]

2.6.1 Accuracy(Doğruluk): Sistemde doğru olarak yapılan tahminlerin tüm tahminlere oranıdır.

Accuracy = (TP+TN) / (TP+TN+FN+FP)

2.6.2 Precision(Kesinlik) : Pozitif olarak tahmin edilen bir durumdaki başarıyı gösteren durum.

Precision = TP / (TP + FP)

2.6.3 Recall(Duyarlılık) : Pozitif durumların ne kadar başarılı tahmin edildiğini gösterir.

Recall = TP / (TP + FN)

2.6.4 F-measure: Precision ve Recall metriklerinin harmonik ortalamasıdır.

F-Measure. = 2. Precision. Recall / Precision + Recall [6]

# BULGULAR

426 tane eğitim ve 141 tanede test verisiyle elde edilen hata matrisleri, kesinlik, duyarlılık, f1 skor ve doğruluk Tablo 1’ de gösterilmiştir. Bu hata matrisleri Doğru Pozitif (DP), Yanlış Pozitif (YP), Doğru Negatif (DN) ve Yanlış Negatif (YN) olmak üzere dört indeksten oluşur. DP, gerçekte kanser olup, önerilen modelinde kanser olarak tespit ettiği örnek sayısıdır. YP, gerçekte kanser olup, önerilen modelin kanser olmayan olarak tahmin ettiği örnek sayısını ifade etmektedir. DN gerçekte kanser olmayıp önerilen modelin de kanser olmayan olarak tahmin ettiği örnek sayısıdır. YN, gerçete kanser olmayıp önerilen modelin kanser olarak tahmin ettiği örnek sayısıdır. Bu dört indeks kullanılarak performans metrikleri elde edilmiştir.

Hata matrisindeki değerler kullanılarak Doğruluk, Kesinlik, duyarlılık ve F1-Skor performans ölçütleriyle modellerin performansları ölçülmüştür. Random Forest yöntemi %91 doğrulukla en başarılı model olmuştur. Kanser sınıfını tespit etmede Kesinlik metriği %93‘lik başarım KNN yöntemleri en başarılı model olmuştur.

Tablo Modellerin Sınıflandırma Başarıları [11]

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Model | Doğru Pozitif | Yanlış Pozitif | Yanlış Negatif | Doğru Negatif | Kesinlik | Duyarlılık | Doğruluk | F1-Skor |
| DT | 83 | 5 | 5 | 49 | 0.90 | 0.90 | 0.92 | 0.90 |
| NB | 67 | 21 | 4 | 50 | 0.70 | 0.92 | 0.82 | 0.79 |
| KNN | 85 | 3 | 10 | 44 | 0.93 | 0.81 | 0.90 | 0.87 |
| RF | 83 | 5 | 4 | 50 | 0.90 | 0.92 | 0.93 | 0.91 |

# SONUÇ

Yüksek hassasiyetli meme kanseri prognozu, kanser teşhisi ve tedavisi için önemli bir eşiğidir.

Makine öğrenimi birçok farklı alanda olduğu gibi tıp alanında da yaygın olarak kullanılıyor ve hastalıkların teşhisinde destek sistemi rolü oynuyor. Özellikle kanser teşhisinde kullanımı giderek artmaktadır. Meme kanseri, kanser türü arasında en sık görülen kanser türüdür ve doğru ve erken teşhis edilmezse ölümcül olabilir. Bu nedenden dolayı meme kanserinin doğru ve etkin teşhisi çok önemlidir. Bu çalışmada, Wisconsin Üniversitesi Hastanesi'nde toplanan meme kanseri bulgularından oluşan 567 örneklik bir veri seti, dört farklı makine öğrenimi modelinde sınıflandırıldı ve modellerin performansı karşılaştırıldı (Tablo 1). En yüksek doğruluk oranını random forrestin verildiği görüldü. Daha sonraki araştırmalarda yeni sınıflandırma yöntemi sırasında elde edilen özelliklere göre kullanılması planlanmaktadır.

# KAYNAKLAR

[1] Aydıntuğ S., “Meme Kanserinde Erken Tanı”, Sürekli Tıp Eğitimi Dergisi, Cilt. 13, Sayı. 6, ss 228, 2014.

[2] Öztemur Y, Aydos A, Gür-Dedeoğlu B., “Meme kanseri mikrodizin verilerinin biyoinformatik yöntemler ile bir araya getirilmesi - Metaanaliz yaklaşımları”, Turk Hijyen Deneysel Biyoloji Dergisi, 72(2), 155-162, 2015.

[3] Yang, Y., Loog, M., A benchmark and comparison of active learning for logistic regression, Pattern Recognition, 83, 401-415, 2018.

[4] Bı̇len, Mehmet, Ali Hakan Işik, and Tuncay Yı̇ğı̇t. "A hybrid Artificial Neural Network-Genetic Algorithm approach for classification of microarray data", 2015 23nd Signal Processing and Communications Applications Conference (SIU). IEEE, 2015.

[5] Ulgen , E. Kaan. 16 Ekim 2017. “ Makine Öğrenimi Bölüm-2 (k-En Yakın Komşuluk)”,[Çevrimiçi]. Erişilebilir: https://medium.com/@k.ulgen90/makine-%C3%B6%C4%9Frenimi-b%C3%B6l%C3%BCm-2-6d6d120a18e1 [Erişim Tarihi: 15-Kasım-2022]

[6] Küçük , Zeynep. “ Matris Hatası Nedir? -Confusion Matrix”,[Çevrimiçi]. Erişilebilir: https://womaneng.com/matris-hatasi-nedir-confusion-matrix/ [Erişim Tarihi: 17-Aralık-2022]

[7] Mbaabu, Onesmus. 11 Aralık 2020. “ Introduction to Random Forest in Machine Learning”, [Çevrimiçi]. Erişilebilir: https://www.section.io/engineering-education/introduction-to-random-forest-in-machine-learning/ [Erişim Tarihi: 5-Aralık-2022]

[8] Akca , M. Fatih. 7 Eylül 2020. “ Karar Ağaçları (Makine Öğrenmesi Serisi-3)”,[Çevrimiçi]. Erişilebilir:https://medium.com/deep-learning-turkiye/karar-a%C4%9Fa%C3%A7lar%C4%B1-makine-%C3%B6%C4%9Frenmesi-serisi-3-a03f3ff00ba5 [Erişim Tarihi: 20-Kasım-2022]

[9] Yıldız, O., Tez, M., Bilge, H. Ş., Akcayol, M. A., & Güler, İ., “ Meme Kanseri Sınıflandırması İçin Veri Füzyonu Ve Genetik Algoritma Tabanlı Gen Seçimi”, Journal of the Faculty of Engineering & Architecture of Gazi University, 27(3), 2012.

[11] Kodlar

# EKLER

#### Kodlar

**import numpy as np**

**import pandas as pd**

**import numpy as np**

**from sklearn.metrics import accuracy\_score**

**from sklearn.feature\_extraction.text import CountVectorizer**

**from sklearn.model\_selection import train\_test\_split**

**from sklearn.preprocessing import Normalizer**

**from sklearn.metrics import confusion\_matrix**

**from sklearn.metrics import precision\_score**

**from sklearn.metrics import recall\_score**

**from sklearn.metrics import f1\_score**

**from sklearn.tree import DecisionTreeClassifier**

**dataset = pd.read\_csv('BrestCancerData.csv')**

**dataset.shape**

**dataset.info()**

**diagnosis\_verisi = list(dataset.shape)[0] #567 erimizi diagnosis\_verisi liste şeklinde değişkenine atadık**

**diagnosis\_kategori = list(dataset['diagnosis'].value\_counts()) #diagnosis etiketindeki aynı verileri counts ile saydık**

**print("\n \t toplam veri {} tane, diagnosis etiketinde, {} kanser ve {} tane kanser değil verisi mevcut.".format(diagnosis\_verisi, diagnosis\_kategori[1], diagnosis\_kategori[0]))**

**dataset.isnull().sum()**

**dataset.head(30)**

**dataset.drop(['id','Unnamed: 32'],axis=1,inplace=True)**

**dataset.head(5)**

**deger = {"B": 0, "M": 1}**

**dataset["diagnosis"]=dataset["diagnosis"].map(deger)**

**dataset.head(10)**

**y = dataset.iloc[:,0].values**

**a=dataset.drop("diagnosis", axis=1)**

**X= a.iloc[:,0:11]**

**X**

**Y**

**def train\_predict(dataset, classifier= DecisionTreeClassifier(criterion = 'entropy')):**

**#Verimizin %75 ini eğitim %25 ini test içi ayırdık**

**X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y,**

**test\_size = 0.25,random\_state = 5)**

**#random stateye verdğimiz değer sürekli aynı yerden verileri böler**

**#Bundan dolayı program her çalıstırılıdğında sonuçlar aynı cıkar.**

**sc = Normalizer()**

**X\_train = sc.fit\_transform(X\_train)**

**X\_test = sc.transform(X\_test)**

**classifier.fit(X\_train, y\_train,)**

**y\_pred = classifier.predict(X\_test)**

**cm = confusion\_matrix(y\_test, y\_pred)**

**pr = precision\_score(y\_test, y\_pred,zero\_division=0)**

**rc = recall\_score(y\_test, y\_pred,zero\_division=0)**

**accuracy = accuracy\_score(y\_test, y\_pred)**

**f1 = f1\_score(y\_test, y\_pred,zero\_division=0)**

**print("confusion matrix",cm)**

**print("precision",pr)**

**print("recall",rc)**

**print("accuracy", accuracy)**

**print("f1 score",f1)**

**return Classifier**

**classifier = train\_predict(dataset)**

**confusion matrix [[82 6]**

**# [ 5 49]]**

**# precision 0.8909090909090909**

**# recall 0.9074074074074074**

**# accuracy 0.9225352112676056**

**# f1 score 0.8990825688073394**

**from sklearn.naive\_bayes import GaussianNB**

**clf = GaussianNB()**

**classifier = train\_predict(dataset,classifier=clf)**

**# confusion matrix [[67 21]**

**# [ 4 50]]**

**# precision 0.704225352112676**

**# recall 0.9259259259259259**

**# accuracy 0.823943661971831**

**# f1 score 0.7999999999999999**

**from sklearn.ensemble import RandomForestClassifier**

**clf=RandomForestClassifier(random\_state =6)**

**classifier = train\_predict(dataset,classifier=clf)**

**# confusion matrix [[83 5]**

**# [ 4 50]]**

**# precision 0.9090909090909091**

**# recall 0.9259259259259259**

**# accuracy 0.9366197183098591**

**# # f1 score 0.9174311926605504**

**from sklearn.neighbors import KNeighborsClassifier**

**clf = KNeighborsClassifier(n\_neighbors=6)#n\_neighbors=6 en yakın 6 komsuya bak**

**classifier = train\_predict(dataset,classifier=clf)**

**confusion matrix [[85 3]**

**# [10 44]]**

**# precision 0.9361702117659575**

**# recall 0.8148148148148148**

**# accuracy 0.9084507042253521**

**# f1 score 0.8712871287128713**

**radius\_mean = float(input(" radius\_mean-Yarıçap :"))**

**texture\_mean = float(input(" texture\_mean-Doku :"))**

**perimeter\_mean = float(input(" perimeter\_mean-Çap :"))**

**area\_mean = float(input("area\_mean- Alan :"))**

**smoothness\_mean = float(input(" smoothness\_mean -Pürüzsüzlük:"))**

**compactness\_mean= float(input("compactness\_mean\_mean-Yoğunluk :"))**

**concavity\_mean = float(input(" concavity\_mean-İçbükeylik :"))**

**concave\_points\_mean= float(input("concave points\_mean-İçbükey nokta sayısı :"))**

**symmetry\_mean = float(input(" symmetry\_mean-Simetri :"))**

**fractal\_dimension\_mean= float(input(" fractal\_dimension\_mean-Fraktal boyut:"))**

**kullanici = [radius\_mean, texture\_mean, perimeter\_mean, area\_mean,smoothness\_mean, compactness\_mean,concavity\_mean,concave\_points\_mean,**

**symmetry\_mean,fractal\_dimension\_mean,]**

**sonuc =clf.predict([kullanici ])[0]**

**print("tahmin sonucu=",sonuc)**