EARLY STAGE LUNG CANCER PREDICTION USING VARIOUS MACHINE LEARNING TECHNIQUES

omurgasız, organizma, su altı, derisi dikenli içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.

## PROJENİN AMACI

Bu projenin temel amacı, akciğer kanseri teşhisine yönelik güvenilir bir makine öğrenmesi modeli geliştirmek ve farklı algoritmaların bu sınıflandırma görevindeki başarımını karşılaştırmaktır. Bu kapsamda, çeşitli semptomlara ve demografik verilere dayanan bir veri seti kullanılarak, bireylerin akciğer kanseri riski taşıyıp taşımadığını tahmin etmek hedeflenmiştir. Kanser gibi hayati bir hastalıkta özellikle yanlış negatif tahminlerin (FN) ciddi sonuçlar doğurabileceği göz önünde bulundurularak modeller bu açıdan da değerlendirilmiştir.

## METODOLOJİ

**1. Veri Seti ve Yükleme**

* Kaynak: lung\_cancer.csv ([https://www.kaggle.com/datasets/mysarahmadbhat/lung-cancer/data](https://www.google.com/url?q=https%3A%2F%2Fwww.kaggle.com%2Fdatasets%2Fmysarahmadbhat%2Flung-cancer%2Fdata))
* Özellikler: Yaş, cinsiyet, sigara içme durumu, nefes darlığı, anksiyete, alkol kullanımı vb.
* Hedef değişken: LUNG\_CANCER (0 = Hayır, 1 = Evet)

**2. Veri Temizleme ve Ön İşleme**

* “YES/NO” ve “M/F” gibi kategorik veriler sayısallaştırıldı.
* Tüm sütunlardaki 1 ve 2 değerleri 0 ve 1’e çevrildi.
* Eksik değer ve tekrar eden satırlar kontrol edildi ve verisetinden çıkarıldı.
* Eğitim verisi StandardScaler ile ölçeklendirildi.

**3. Veri Görselleştirme**

* Korelasyon matrisi oluşturuldu; en yüksek ilişkili değişken çiftleri belirlendi.
* Cinsiyet, yaş ve sigara kullanımı gibi özniteliklere göre kanser dağılımı analiz edildi.

**4. Modelleme: Sınıflandırma Algoritmaları**

**a. Destek Vektör Makineleri (SVM)**

* Doğrusal (linear) ve doğrusal olmayan (rbf) çekirdekler test edildi.
* Linear SVM daha başarılı sonuçlar verdi (%92+ doğruluk).

**b. K-En Yakın Komşu (KNN)**

* 1-10 arasında komşu sayıları test edildi.
* k=2 en yüksek doğruluk verdi (%92.8 doğruluk).

**c. Random Forest (RF)**

* 100 ağaçlı model uygulandı.
* Dengeli performans gösterdi, %92+ doğruluk elde edildi.

**d. Voting Classifier (Toplu Oylama)**

* SVM (linear), KNN, RF modelleri birlikte kullanıldı.
* En dengeli ve yüksek doğruluk (%94.6) bu modelde elde edildi.
* Özellikle yanlış negatif tahminlerin minimumda olması (FN = 1) sebebiyle tercih edilebilir.

## KODLAR

# \*\*Libs & Data Laod\*\*

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

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

from sklearn.preprocessing import LabelEncoder, StandardScaler

from sklearn.metrics import classification\_report, confusion\_matrix, accuracy\_score

from sklearn.svm import SVC

from sklearn.ensemble import RandomForestClassifier, VotingClassifier

from sklearn.neighbors import KNeighborsClassifier

df = pd.read\_csv('/content/drive/MyDrive/lung\_cancer.csv')

df.head()

"""# \*\*Data cleaning & preprocessing\*\*"""

df.shape

# '1' → 0, '2' → 1 dönüşümünü tüm sütunlarda (LUNG\_CANCER hariç)

for col in df.columns:

    if col != 'LUNG\_CANCER':

        df[col] = df[col].replace({1: 0, 2: 1})

df.head()

df['LUNG\_CANCER'] = df['LUNG\_CANCER'].replace({'YES': 1, 'NO': 0}).astype(int) #lung cancer i yes no dan 0 1 e çevirme

df['GENDER'] = df['GENDER'].replace({'M': 1, 'F': 0}).astype(int)# cinsiyet erkek = 1, female = 0

df.info()

df.head()

df.isna().sum() # eksik veri kontrolü

df.duplicated().sum() # tekrarlanmış veri kontrol

df.drop\_duplicates(inplace=True) #tekrarlanmış verileri sil

df.duplicated().sum()

df.head()

df.tail() # yeni son 5

"""# \*\*Data Visualization\*\*"""

plt.figure(figsize=(12, 8))

sns.heatmap(df.corr(), annot=True, cmap="coolwarm", fmt=".2f") # fmt=".2f" ile 2 basamaklı gösterim

plt.title("Correlation Matrix")

plt.show()

"""\*\*En Yüksek İlişkili Özellikler (Features)\*\*

1.   Sarı Parmaklar - Anksiyete \*\*(%56)\*\*

2.   Yutkunma Güçlüğü - Anksiyete \*\*(%48)\*\*

3.   Nefes Darlığı - Yorgunluk \*\*(%43)\*\*

4.   Alkol Tüketimi - Alerji \*\*(%38)\*\*

"""

# Cinsiyete göre Kanser dağılımı

sns.countplot(data=df, x="GENDER", hue="LUNG\_CANCER")

plt.xticks([0, 1], ['Kadın', 'Erkek'])

plt.title("Cinsiyete göre Kanser")

plt.xlabel("Cinsiyet")

plt.ylabel("Sayı")

plt.legend(title="Lung Cancer")

plt.show()

"""Ufak bir farkla erkeklerde akciğer kanseri görülümünün daha fazla olduğu gözüküyor"""

# Yaşa göre kanser dağılımı (Box Plot)

sns.boxplot(data=df[df['LUNG\_CANCER'] == 1], x="LUNG\_CANCER", y="AGE") #sadece 'lung cancer = 1' olanlar

plt.title("Akciğer Kanseri Olanların Yaş Dağılımı")

plt.xlabel("Akciğer Kanseri")

plt.ylabel("Yaş")

plt.xticks([1])

plt.show()

# Sigara kullanımına göre kanser dağılımı

sns.countplot(data=df, x="SMOKING", hue="LUNG\_CANCER")

plt.xticks([0, 1], ['Kullanmayan', 'Kullanan'])

plt.title("Sigara Kullanımına Göre Kanser")

plt.xlabel("Sigara")

plt.ylabel("Sayı")

plt.legend(title="Lung Cancer")

plt.show()

"""# \*\*Machine Learning\*\*

## \*\*Train Test Split\*\*

"""

X = df.drop('LUNG\_CANCER', axis=1)

y = df['LUNG\_CANCER']

# Eğitim ve test ayırma (test %20)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42, stratify=y)

# standartlaştırma (-1, 1)

scaler = StandardScaler()

X\_train\_scaled = scaler.fit\_transform(X\_train)

X\_test\_scaled = scaler.transform(X\_test)

"""## \*\*SVM\*\*"""

#yine svm ama rbf (doğrusal olmayan) modeli deniyoruz

svm\_model\_rbf = SVC(kernel='rbf')  # doğrusal olmayan model

svm\_model\_rbf.fit(X\_train\_scaled, y\_train)

# Test verisiyle tahmin

y\_pred = svm\_model\_rbf.predict(X\_test\_scaled)

# Score

print("SVM (rbf) Accuracy:", accuracy\_score(y\_test, y\_pred))

print("\nConfusion Matrix:\n", confusion\_matrix(y\_test, y\_pred))

print("\nClassification Report:\n", classification\_report(y\_test, y\_pred))

svm\_model = SVC(kernel='linear')  # doğrusal model

svm\_model.fit(X\_train\_scaled, y\_train)

# Test verisiyle tahmin

y\_pred = svm\_model.predict(X\_test\_scaled)

print("SVM Accuracy:", accuracy\_score(y\_test, y\_pred))

print("\nConfusion Matrix:\n", confusion\_matrix(y\_test, y\_pred))

print("\nClassification Report:\n", classification\_report(y\_test, y\_pred))

"""sınıflar doğrusal ayrıldığı için linear, rbf ye göre daha iyi sonuç veriyor dolayısıyla linear ile devam edelim. Ayrıca precision,recall ve f1 değerleri birbirine oldukça yakın ve yüksek olduğundan dengeli ve başarılı olduğunu söylebiliriz."""

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

# Sınıf isimlerini (0 ve 1) tanımla

class\_names = ['0', '1'] # (pozitif, negatif)

# 2. Karmaşıklık Matrisini Görselleştir

plt.figure(figsize=(8, 6))

sns.heatmap(cm, annot=True, fmt="d", cmap="Blues",

            xticklabels=class\_names, yticklabels=class\_names)

plt.title("SVM Model Confusion Matrix")

plt.ylabel("Gerçek Etiketler")

plt.xlabel("Tahmin Edilen Etiketler")

plt.show()

"""SVM model;

\* 7 TN

\* 45 TP

\* 3 FN

\* 1 FP

ile 56 veride 4 yanlış ile oldukça başarılı tahmin yapıyor ve

## \*\*KNN\*\*

"""

training\_accuracy = []

test\_accuracy = []

# n\_neighbors değerlerini 1 den 10 a dene

neighbors\_settings = range(1, 11)

for n\_neighbors in neighbors\_settings:

    clf = KNeighborsClassifier(n\_neighbors=n\_neighbors)

    # Fit the model

    clf.fit(X\_train\_scaled, y\_train)

    # training accuracy

    training\_accuracy.append(clf.score(X\_train\_scaled, y\_train))

    # test accuracy

    test\_accuracy.append(clf.score(X\_test\_scaled, y\_test))

plt.plot(neighbors\_settings, training\_accuracy, label="training accuracy")

plt.plot(neighbors\_settings, test\_accuracy, label="test accuracy")

plt.ylabel("Accuracy")

plt.xlabel("n\_neighbors")

plt.legend()

plt.show()

"""En yüksek test doğrulu n\_neighbors = 2 de alıyoruz dolayısıyla modeli 2 ile oluşturalım"""

# KNN model

knn\_model = KNeighborsClassifier(n\_neighbors=2)

knn\_model.fit(X\_train\_scaled, y\_train)

# Tahmin yap

y\_pred\_knn = knn\_model.predict(X\_test\_scaled)

# Performans değerlendirmesi

print("KNN Accuracy:", accuracy\_score(y\_test, y\_pred\_knn))

print("\nConfusion Matrix:\n", confusion\_matrix(y\_test, y\_pred\_knn))

print("\nClassification Report:\n", classification\_report(y\_test, y\_pred\_knn))

"""0 sınıfında recall'ın 1 olması ve precision ile arasında fark olması ayrıca 1 sınıfında da precisionun 1 olması belki overfitting göstergesi olabilir. True pozitifleri tamamen yakalıyor."""

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

class\_names = ['0', '1']

# KNN Confusion Matrisi

plt.figure(figsize=(8, 6))

sns.heatmap(cm\_knn, annot=True, fmt="d", cmap="Blues",

            xticklabels=class\_names, yticklabels=class\_names)

plt.title("KNN Model Confusion Matrix")

plt.ylabel("Gerçek Etiketler")

plt.xlabel("Tahmin Edilen Etiketler")

plt.show()

"""\*\*KNN model;\*\*

\* 44 TP

\* 8 TN

\* 0 FP

\* 4 FN

KNN negatif sınıfı çok iyi ayırıyor. Model %93 doğrulukla çalışıyor. Hatalı pozitif tahmini daha önemliyse SVM yerine KNN tercih etmek daha mantıklı gözüküyor. Hatalı negatif tahmin daha kritik ise (örneğin hastayı atlamak) SVM daha kullanışlı.

## \*\*Random Forest\*\*

"""

rf\_model = RandomForestClassifier(n\_estimators=100, random\_state=42)  # n\_estimators: ağaç sayısı

rf\_model.fit(X\_train\_scaled, y\_train)

y\_pred\_rf = rf\_model.predict(X\_test\_scaled)

print("Random Forest")

print("Accuracy:", accuracy\_score(y\_test, y\_pred\_rf))

print("\nConfusion Matrix:\n", confusion\_matrix(y\_test, y\_pred\_rf))

print("\nClassification Report:\n", classification\_report(y\_test, y\_pred\_rf))

"""Metriklerin bukadar yakın olması dengeli bir performans olduğunu gösterir.

"""

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

class\_names = ['0', '1']

# RF Confusion Matris

plt.figure(figsize=(8, 6))

sns.heatmap(cm\_rf, annot=True, fmt="d", cmap="Blues",

            xticklabels=class\_names, yticklabels=class\_names)

plt.title("Random Forest Confusion Matrix")

plt.ylabel("Gerçek Etiketler")

plt.xlabel("Tahmin Edilen Etiketler")

plt.show()

"""\*\*RF model;\*\*

\* 46 TP

\* 6 TN

\* 2 FP

\* 2 FN

TP ve TN değerleri oldukça başarılı gözüküyor.

## \*\*Voting Classifier\*\*

"""

svc\_clf = SVC(kernel='linear', probability=True, random\_state=42)

knn\_clf = KNeighborsClassifier(n\_neighbors=2)

rf\_clf  = RandomForestClassifier(n\_estimators=100, random\_state=42)

voting\_clf = VotingClassifier(estimators=[('svm', svc\_clf),('knn', knn\_clf),('rf', rf\_clf)],voting='hard')

voting\_clf.fit(X\_train\_scaled, y\_train)

y\_pred\_voting = voting\_clf.predict(X\_test\_scaled)

print("\n\n=== Voting Classifier (SVM + KNN + RF) ===")

print("Accuracy:", accuracy\_score(y\_test, y\_pred\_voting))

print("\nConfusion Matrix:\n", confusion\_matrix(y\_test, y\_pred\_voting))

print("\nClassification Report:\n", classification\_report(y\_test, y\_pred\_voting))

"""accuracy ve diğer metrikler oldukça yüksek ve dengeli, model oldukça iyi öğrenmiş. Özellikle 1 sınıfında tahminler çok tutarlı"""

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

class\_names = ['0', '1']

# VC Confusion Matrisi

plt.figure(figsize=(8, 6))

sns.heatmap(cm\_voting, annot=True, fmt="d", cmap="Blues",

            xticklabels=class\_names, yticklabels=class\_names)

plt.title("Voting Classifier Confusion Matrix")

plt.ylabel("Gerçek Etiketler")

plt.xlabel("Tahmin Edilen Etiketler")

plt.show()

"""\*\*VC model;\*\*

\* 47 TP

\* 6 TN

\* 2 FP

\* 1 FN

Kanser teşhisinde yanlış negatif en önemli faktör olduğundan '1 FN' ile tercih edilebilecek en iyi model olduğunu düşünüyorum. """

## DENEYSEL SONUÇLAR

metin, ekran görüntüsü, yazı tipi, sayı, numara içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.

**REFERANS SONUÇ**

metin, ekran görüntüsü, yazı tipi, sayı, numara içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.

Benim modellerimin accuracy oranlarının Referans olarak aldığım dokümandakinden daha düşük olmasının sebebinin veri sayımın daha az olduğundan kaynaklandığını düşünüyorum.

### **SVM (rbf)**

metin, ekran görüntüsü, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.

* **Accuracy oranı kabul edilebilir düzeyde (%85.7) ancak diğer modellere göre daha düşük kalıyor.**
* **Pozitif sınıfta (kanserli bireyler) yüksek başarı sağlanmış (recall = 0.96), yani hastalar çoğunlukla doğru tespit edilmiş.**
* **Ancak negatif sınıfta (sağlıklı bireyler) ciddi bir başarısızlık mevcut (recall = 0.25), yani model sağlıklı bireyleri genellikle hasta sanıyor.**
* **Bu durum, modelin pozitif sınıfa fazla odaklandığını gösteriyor. Bu veri dengesizliği veya RBF kernel’in bu veri setine uygun olmamasından kaynaklanabilir.**
* **Precision ve recall değerleri arasında sınıflar arası fark fazla. Bu, modelin dengesiz performans gösterdiğini ortaya koyar.**
* **56 denemede toplam 8 yanlış tahmin yapılmış (2 FN, 6 FP) – bu da modelin çok sayıda hatalı pozitif ürettiğini gösteriyor.**
* **Overfitting veya underfitting net şekilde gözlenmiyor; ancak veri sayısının azlığı modelin genel davranışını etkilemiş olabilir.**

### **SVM (Linear)**

metin, ekran görüntüsü, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.

metin, ekran görüntüsü, diyagram içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.

* **Accuracy gayet iyi (%92.8). Değerler de gayet yakın, uzak olsa da az veriden kaynaklandığını düşünüyorum. Yani overfitting ve underfitting muhtemel gözükmüyor.**
* **Sınıflar doğrusal ayrıldığı için linear, rbf ye göre daha iyi sonuç veriyor**
* **Pozitif sınıfta (kanserli bireyler) yüksek başarı (recall = 0.94)**
* **Negatif sınıfta da güçlü tahmin (recall = 0.88)**
* **Dengeli ve başarılı bir model**
* **56 Denemede 4 Yanlış (7 TN, 45 TP, 3 FN 1 FP)**

### **KNN**

metin, ekran görüntüsü, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.

metin, ekran görüntüsü, diyagram, dikdörtgen içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.

metin, diyagram, çizgi, öykü gelişim çizgisi; kumpas; grafiğini çıkarma içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.

(En yüksek doğruluk n\_neighbors = 2 de alındı)

* **Genel doğruluk oranı oldukça yüksek (%92.8).**
* **Negatif sınıfta (sağlıklı bireyler) model kusursuz performans gösteriyor (recall = 1.00), hiç yanlış negatif tahmin yok.**
* **Pozitif sınıfta (kanserli bireyler) da oldukça başarılı (recall = 0.92, precision = 1.00), model hastaları doğru tespit ediyor.**
* **Precision ve recall arasındaki fark negatif sınıfta yüksek (precision 0.67, recall 1.00), bu durum negatif sınıf için sınırlı sayıda örnek olmasından veya hafif overfitting etkisinden kaynaklanabilir.**
* **56 denemede toplam 4 yanlış tahmin var (4 FN, 0 FP), yani hatalar sadece hasta bireylerin atlanması yönünde (ki bu en istemediğimiz şey).**
* **Model, özellikle yanlış pozitiflerin kritik olmadığı durumlarda tercih edilebilir.**
* **Overfitting veya underfitting belirtileri açık değil ancak sınıf dağılımı ve küçük veri seti model performansını etkileyebilir.**

### **RANDOM FOREST**

metin, ekran görüntüsü, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.

metin, ekran görüntüsü, diyagram, dikdörtgen içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.

* **Genel doğruluk seviyesi yüksek (%92.8) ve dengeli bir performans sergiliyor.**
* **Pozitif sınıfta (kanserli bireyler) başarılı tahminler yapıyor (recall = 0.96, precision = 0.96).**
* **Negatif sınıfta (sağlıklı bireyler) performans biraz daha düşük, ancak kabul edilebilir (recall = 0.75, precision = 0.75).**
* **Yanlış pozitif ve yanlış negatif sayıları dengeli (2 FP, 2 FN) ve sınıflar arası dengeyi koruyor.**
* **Model overfitting veya underfitting belirtisi göstermiyor; karar ağaçlarının birleşimi sayesinde genelleme gücü iyi.**

### **VOTING CLASSIFIER**

metin, ekran görüntüsü, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.

metin, ekran görüntüsü, diyagram, dikdörtgen içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.

* **En yüksek doğruluk oranına sahip model (%94.6).**
* **Pozitif sınıfta yüksek başarı (recall = 0.98, precision = 0.96), yani kanserli bireylerin çoğunu doğru tespit ediyor.**
* **Negatif sınıfta da kabul edilebilir performans (recall = 0.75, precision = 0.86).**
* **Yanlış negatif sayısı sadece 1. Kanser atlama riskini minimize ediyor, bu tıbbi uygulamalar için çok önemli.**
* **Yanlış pozitif sayısı 2, kabul edilebilir seviyede.**
* **Overfitting veya underfitting gözlenmiyor; model tüm algoritmaların güçlü yönlerini birleştiriyor.**
* **Kanser teşhisinde kritik olan yanlış negatif sayısının en az olduğu model olarak tercih edilebilir.**

## KAYNAKLAR

**VERİTABANI:** <https://www.kaggle.com/datasets/mysarahmadbhat/lung-cancer/data>

**DOKÜMAN:** [Early Stage Lung Cancer Prediction Using Various Machine Learning Techniques | IEEE Conference Publication | IEEE Xplore](https://ieeexplore.ieee.org/document/9297576)

