

**FATİH KÜÇÜK**

**20040301027**

**YAZILIM MÜHENDİSLİĞİ**

**4. SINIF**

**MAKİNE ÖĞRENMESİ**

**VİZE ÖDEVİ**

**GÖKALP TULUM**

1. **SORU: Ara sınav ödevinde size atanan veri setine tüm algoritmalar için normalizasyon yöntemlerinden birini uygulayınız.**

**Python Kodu:**

from sklearn import preprocessing

import numpy as np

with open("veri-seti.txt", "r") as file:

lines = file.readlines()

data = []

for line in lines:

row = [float(x) for x in line.strip().split('\t')]

data.append(row)

data = np.array(data)

min\_max\_scaler = preprocessing.MinMaxScaler()

normalized\_data = min\_max\_scaler.fit\_transform(data)

print("Normalize Edilmiş Veri:")

print(normalized\_data)

1. **SORU: Veri setine PCA ve LDA algoritmalarını uygulayarak en yüksek değere sahip iki öz değer için boyut indirgeme işlemini gerçekleştiriniz. PCA ve LDA için hangi özniteliklerin en ayırt edici olduğunu raporlayınız. Alt maddelerde verilen yöntemleri ana veri setine ve PCA uygulanmış transformasyon öznitelik matrisine uygulayınız.**

**Python Kodu:**

import pandas as pd

from sklearn.decomposition import PCA

from sklearn.discriminant\_analysis import LinearDiscriminantAnalysis

import matplotlib.pyplot as plt

file\_path = "veri-seti.txt"

names = ['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome']

data = pd.read\_csv(file\_path, names=names, sep='\t')

X = data.drop('Outcome', axis=1)

y = data['Outcome']

pca = PCA(n\_components=2)

X\_pca = pca.fit\_transform(X)

lda = LinearDiscriminantAnalysis(n\_components=1) # LDA'da n\_components parametresi 1 olarak ayarlandı

X\_lda = lda.fit\_transform(X, y)

print("PCA Sonuçları:")

print(pd.DataFrame(X\_pca, columns=['PCA1', 'PCA2']).head())

print("\nLDA Sonuçları:")

print(pd.DataFrame(X\_lda, columns=['LD1']).head())

# Görselleştirme

plt.figure(figsize=(12, 5))

plt.subplot(1, 2, 1)

plt.scatter(X\_pca[:, 0], X\_pca[:, 1], c=y, cmap='viridis')

plt.title('PCA')

plt.xlabel('Principal Component 1')

plt.ylabel('Principal Component 2')

plt.subplot(1, 2, 2)

plt.scatter(X\_lda[:, 0], [0] \* len(X\_lda), c=y, cmap='viridis')

plt.title('LDA')

plt.xlabel('LD 1')

plt.tight\_layout()

plt.show()

**PCA Sonuçları:**

PCA1, glucose, insulin, body mass index özniteliklerinden önemli ölçüde etkilenmişken, PCA2, glucose ve body mass index özniteliklerine daha az ama hala önemli ölçüde bağlıdır.

**LDA Sonuçları:**

LD1, glucose, body mass index ve age özniteliklerine yoğun bir şekilde bağımlıdır.

Bu sonuçlara göre, glucose ve body mass index özniteliklerinin hem PCA’da hem de LDA’da belirgin şekilde ayırt edici olduğu görülmektedir.

1. **SORU: Veri setinizi rastgele olarak %70 eğitim %30 test olacak şekilde ayırınız. Eğitim veri seti için Çoklu Doğrusal Regresyon analizi ve Multinominal Lojistik Regresyon analizi yöntemlerini uygulayınız. Elde ettiğiniz katsayıları raporlayınız. Elde ettiğiniz regresyon denklemlerini kullanarak Test kümesi için performans metriklerini hesaplayınız.**

**Python Kodu:**

import pandas as pd

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

from sklearn.linear\_model import LinearRegression, LogisticRegression

from sklearn.metrics import accuracy\_score

# Veri setini yükleme

file\_path = "veri-seti.txt"

names = ['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome']

data = pd.read\_csv(file\_path, names=names, sep='\t')

# Bağımsız değişkenler ve hedef değişkeni ayırma

X = data.drop('Outcome', axis=1)

y = data['Outcome']

# Veri setini eğitim ve test alt kümelerine ayırma

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

# Çoklu Doğrusal Regresyon analizi uygulama

linear\_reg = LinearRegression()

linear\_reg.fit(X\_train, y\_train)

# Katsayıları raporlama

print("Çoklu Doğrusal Regresyon Katsayıları:")

coefficients\_linear = pd.DataFrame(linear\_reg.coef\_, X.columns, columns=['Katsayı'])

print(coefficients\_linear)

# Multinominal Lojistik Regresyon analizi uygulama

logistic\_reg = LogisticRegression(multi\_class='multinomial', solver='lbfgs', max\_iter=1000)

logistic\_reg.fit(X\_train, y\_train)

# Katsayıları raporlama

print("\nMultinominal Lojistik Regresyon Katsayıları:")

coefficients\_logistic = pd.DataFrame(logistic\_reg.coef\_.transpose(), X.columns, columns=['Katsayı'])

print(coefficients\_logistic)

# Test veri seti için performans metriklerini hesaplama

y\_pred\_linear = linear\_reg.predict(X\_test)

y\_pred\_logistic = logistic\_reg.predict(X\_test)

# Performans metriklerini hesaplama

accuracy\_linear = accuracy\_score(y\_test, y\_pred\_linear.round())

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

print("\nTest Veri Seti Performans Metrikleri:")

print("Çoklu Doğrusal Regresyon Doğruluk:", accuracy\_linear)

print("Multinominal Lojistik Regresyon Doğruluk:", accuracy\_logistic)

1. **SORU: Veri setinizi rastgele olarak %70 eğitim %30 test olacak şekilde ayırınız. Veri setinize karar ağaç sınıflandırma algoritmasını uygulayarak ağaç yapısını ve kestirim sonuçlarını veriniz. Test verisi için performans metriklerini hesaplayınız.**

**Python Kodu:**

import pandas as pd

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

from sklearn.tree import DecisionTreeClassifier, plot\_tree

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

import matplotlib.pyplot as plt

file\_path = "veri-seti.txt"

names = ['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome']

data = pd.read\_csv(file\_path, names=names, sep='\t')

X = data.drop('Outcome', axis=1)

y = data['Outcome']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

decision\_tree = DecisionTreeClassifier(random\_state=42)

decision\_tree.fit(X\_train, y\_train)

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

plot\_tree(decision\_tree, feature\_names=names[:-1], class\_names=['0', '1'], filled=True)

plt.show()

y\_pred = decision\_tree.predict(X\_test)

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

print("Doğruluk:", accuracy)

print("Kestirim Sonuçları:")

for i, (real, pred) in enumerate(zip(y\_test, y\_pred)):

print(f"Test {i+1}: Gerçek={real}, Tahmin={pred}")

#Test verisi için performans metriklerini hesaplama.

classification\_report\_result = classification\_report(y\_test, y\_pred)

print("Performans Metrikleri:")

print(classification\_report\_result)

1. **SORU: Veri setinizi rastgele olarak %70 eğitim %30 test olacak şekilde ayırınız. Eğitim veri seti için Naive bayes sınıflandırıcısını uygulayınız. Elde ettiğiniz sonucları raporlayınız. Test verisi için performans metriklerini hesaplayınız.**

**Python Kodu:**

import pandas as pd

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

from sklearn.naive\_bayes import GaussianNB

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

file\_path = "veri-seti.txt"

names = ['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome']

data = pd.read\_csv(file\_path, names=names, sep='\t')

X = data.drop('Outcome', axis=1)

y = data['Outcome']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

naive\_bayes = GaussianNB()

naive\_bayes.fit(X\_train, y\_train)

train\_predictions = naive\_bayes.predict(X\_train)

train\_accuracy = accuracy\_score(y\_train, train\_predictions)

print("Eğitim Veri Seti Doğruluğu:", train\_accuracy)

test\_predictions = naive\_bayes.predict(X\_test)

test\_accuracy = accuracy\_score(y\_test, test\_predictions)

print("Test Veri Seti Doğruluğu:", test\_accuracy)

print("\nTest Veri Seti Sınıflandırma Raporu:")

print(classification\_report(y\_test, test\_predictions))