MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
VERİ BİLİMİNE GİRİŞ DERSİ FİNAL PROJESİ

Makine Öğrenmesi Yöntemleri ile Kalp Hastalığı Tahmini

22120205067, Alper AKPINAR

22120205048, Seferhan KAYA

Ders Sorumlusu

Dr. Öğr. Üyesi Muhammet Sinan BAŞARSLAN

Ocak, 2026

İstanbul Medeniyet Üniversitesi, İstanbul

İÇİNDEKİLER

TABLOLAR LİSTESİ

ŞEKİLLER TABLOSU

1. Kütüphaneler

```
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 StandardScaler

from sklearn.metrics import accuracy_score, confusion_matrix,
classification_report

from sklearn.linear_model import LogisticRegression

from sklearn.ensemble import RandomForestClassifier
```

2. Veri Setini Okuma ve İlk İnceleme

```
df = pd.read_csv("heart.csv")
```

2.1. Veri setinin ilk 5 satırını görüntüleme

```
print("\n--- İlk 5 Satır ---")

print(df.head())
```

Çıktı:

age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
62	0	0	138	294	1	1	106	0	1.9	1	3	2	0

Tablo 1: Veri Setinin İlk 5 Satırı

2.2. Veri Bilgisi

```
print("\n--- Veri Bilgisi ---")
print(df.info())
```

Çıktı:

```
--- Veri Bilgisi ---
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1025 entries, 0 to 1024
Data columns (total 14 columns):
```

#	Column	Non-Null Count	Dtype
0	age	1025 non-null	int64
1	sex	1025 non-null	int64
2	cp	1025 non-null	int64
3	trestbps	1025 non-null	int64
4	chol	1025 non-null	int64
5	fbs	1025 non-null	int64
6	restecg	1025 non-null	int64
7	thalach	1025 non-null	int64
8	exang	1025 non-null	int64
9	oldpeak	1025 non-null	float64
10	slope	1025 non-null	int64
11	ca	1025 non-null	int64
12	thal	1025 non-null	int64
13	target	1025 non-null	int64

Tablo 2: Veri Bilgisi

```
dtypes: float64(1), int64(13)
memory usage: 112.2 KB
 None
```

2.3. Veri İstatistikleri

```
print("\n--- Veri İstatistikleri ---")
print(df.describe())
```

Çıktı:

```
--- Veri İstatistikleri ---
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
count	1025	1025	1025	1025	1025	1025	1025	1025	1025	1025	1025	1025	1025	1025
mean	54.43	0.70	0.94	131.61	246	0.15	0.53	149.11	0.34	1.07	1.39	0.75	2.32	0.51
std	9.07	0.46	1.03	17.52	51.59	0.36	0.53	23.01	0.47	1.18	0.62	1.03	0.62	0.50
min	29	0	0	94	126	0	0	71	0	0	0	0	0	0
25%	48	0	0	120	211	0	0	132	0	0	1	0	2	0
50%	56	1	1	130	240	0	1	152	0	0.8	1	0	2	1
75%	61	1	2	140	275	0	1	166	1	1.8	2	1	3	1
max	77	1	3	200	564	1	2	202	1	6.2	2	4	3	1

Tablo 3: Veri İstatistikleri

2.4. Target Değer Dağılımı

```
print("\n--- Target Değer Dağılımı ---")
print(df["target"].value_counts())
```

Çıktı:

```
--- Target Değer Dağılımı ---
```

target	
1	526
0	499

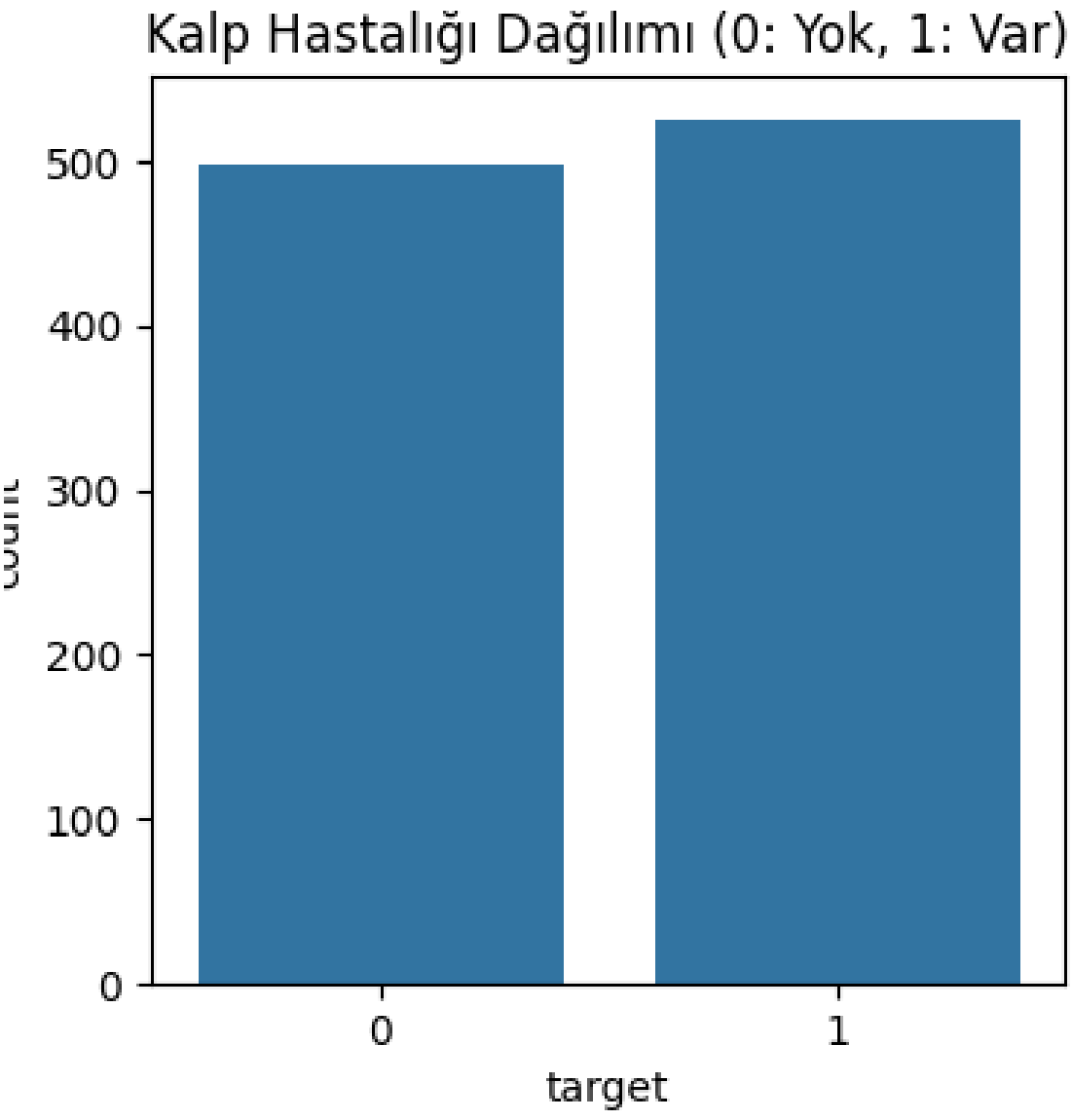
Tablo 4: Target Değer Dağılımı

```
Name: count, dtype: int64
```

3. Keşifsel Veri Analizi (EDA)

```
plt.figure(figsize=(4,4))
sns.countplot(x="target", data=df)
plt.title("Kalp Hastalığı Dağılımı (0: Yok, 1: Var)")
plt.show()
```

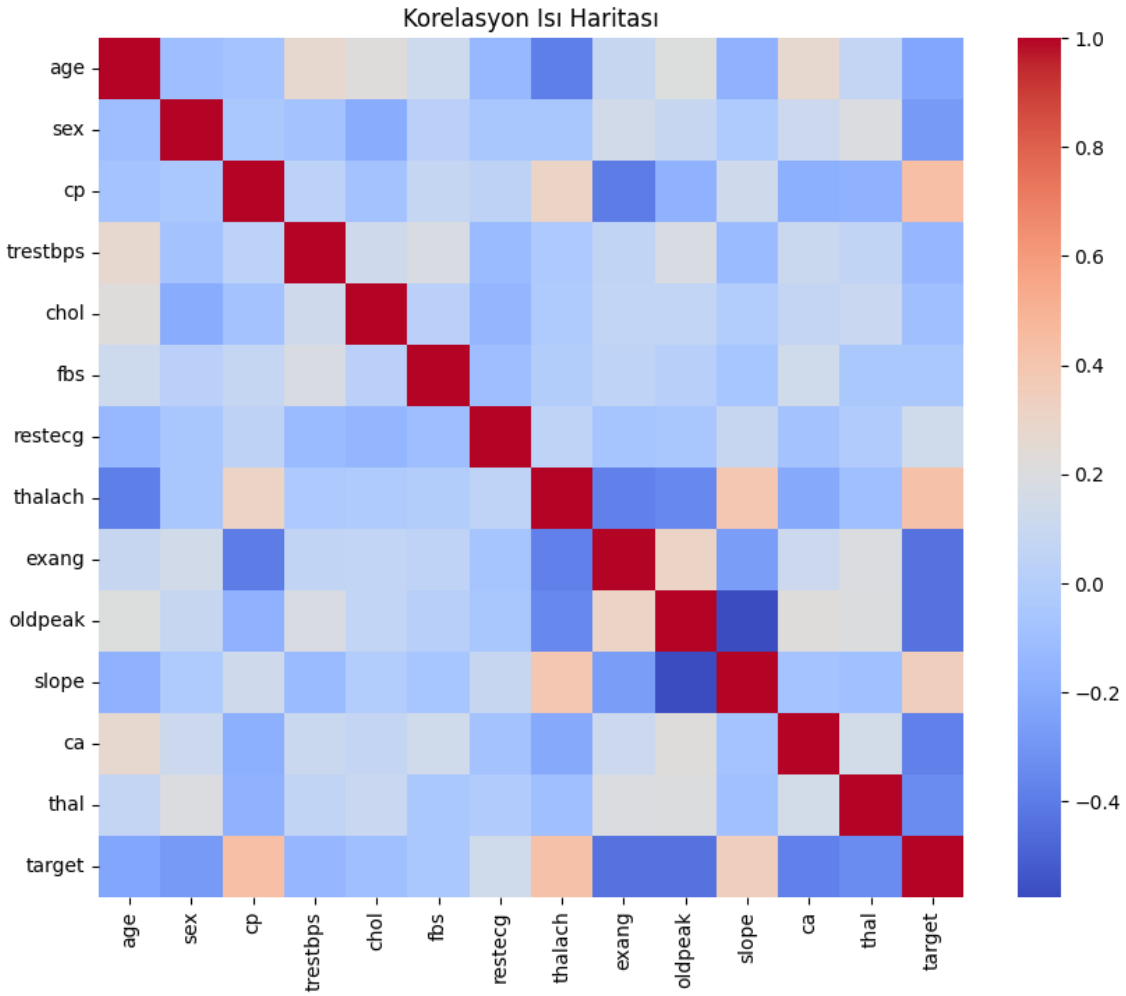
Çıktı:



Şekil 1: Kalp Hastalığı Dağılımı

```
plt.figure(figsize=(10,8))
sns.heatmap(df.corr(), annot=False, cmap="coolwarm")
plt.title("Korelasyon Isı Haritası")
plt.show()
```

Çıktı:



Şekil 2: Korelasyon Isı Haritası

4. Özellik / Hedef Ayrımı

```
X = df.drop("target", axis=1)
y = df["target"]

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)

print("\nEğitim seti boyutu:", X_train.shape)
print("Test seti boyutu:", X_test.shape)
```

Çıktı:

```
Eğitim seti boyutu: (820, 13)
Test seti boyutu: (205, 13)
```

5. Ölçekleme (Scaler)

```
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

6. Lojistik Regresyon Modeli

```
log_model = LogisticRegression(max_iter=1000)

log_model.fit(X_train_scaled, y_train)


y_pred_log = log_model.predict(X_test_scaled)


print("\n==============================")

print(" LOJİSTİK REGRESYON")

print("==============================")

print("Accuracy:", accuracy_score(y_test, y_pred_log))

print("\nClassification   Report:\n",   classification_report(y_test,
y_pred_log))


cm_log = confusion_matrix(y_test, y_pred_log)

print("\nConfusion Matrix:\n", cm_log)


plt.figure(figsize=(4,4))

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

plt.title("Lojistik Regresyon - Confusion Matrix")

plt.show()
```

Çıktı:

==========

LOJİSTİK REGRESYON

==========

Accuracy: 0.8097560975609757

Classification Report:

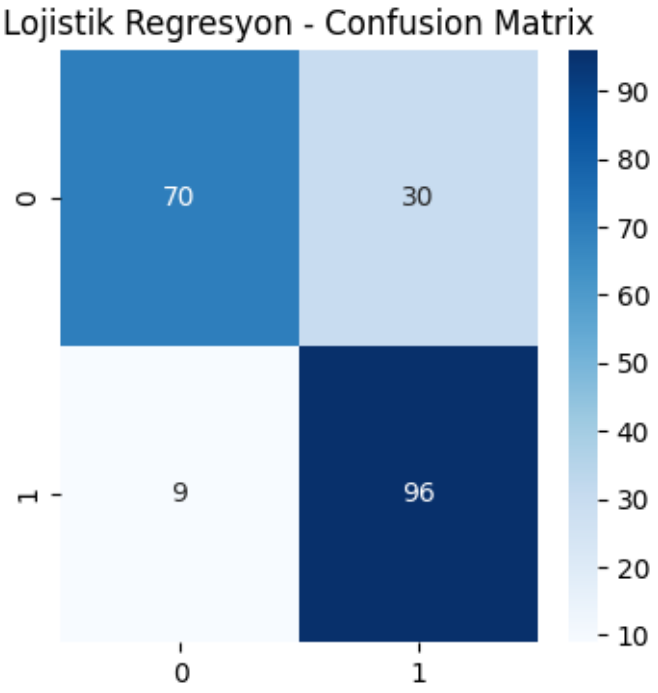
	Precision	**Recall**	**F1-Score**	**Support**
0	0.89	0.70	0.78	100
1	0.76	0.91	0.83	105
Accuracy			**0.81**	205
Macro Avg	0.82	0.81	0.81	205
Weighted Avg	0.82	0.81	0.81	205

Tablo 5: Classification Report

Confusion Matrix:

[[70 30]

[9 96]]



Şekil 3: Lojistik Regresyon - Confusion Matrix

7. Random Forest Modeli

```
rf_model = RandomForestClassifier(n_estimators=200, random_state=42)
rf_model.fit(X_train, y_train)


y_pred_rf = rf_model.predict(X_test)


print("\n==============================")
print(" RANDOM FOREST")
print("==============================")
print("Accuracy:", accuracy_score(y_test, y_pred_rf))
print("\nClassification Report:\n", classification_report(y_test,
y_pred_rf))


cm_rf = confusion_matrix(y_test, y_pred_rf)
print("\nConfusion Matrix:\n", cm_rf)


plt.figure(figsize=(4,4))
sns.heatmap(cm_rf, annot=True, fmt="d", cmap="Greens")
plt.title("Random Forest - Confusion Matrix")
plt.show()
```

Çıktı:

==========

RANDOM FOREST

==========

Accuracy: 1.0

Classification Report:

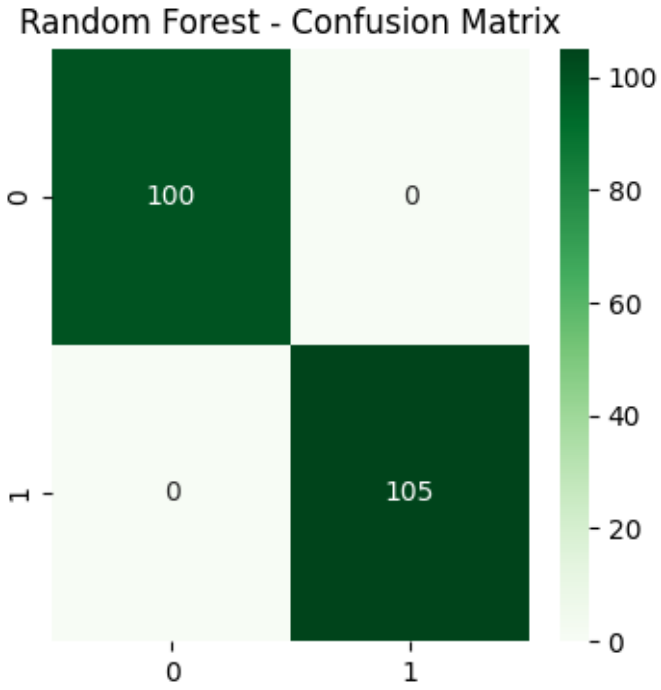
	Precision	**Recall**	**F1-Score**	**Support**
0	1.00	1.00	1.00	100
1	1.00	1.00	1.00	105
Accuracy			**1.00**	205
Macro Avg	1.00	1.00	1.00	205
Weighted Avg	1.00	1.00	1.00	205

Tablo 6: Classification Report

Confusion Matrix:

[[100 0]

[0 105]]



Şekil 4: Random Forest - Confusion Matrix

8. Random Forest Feature Importance

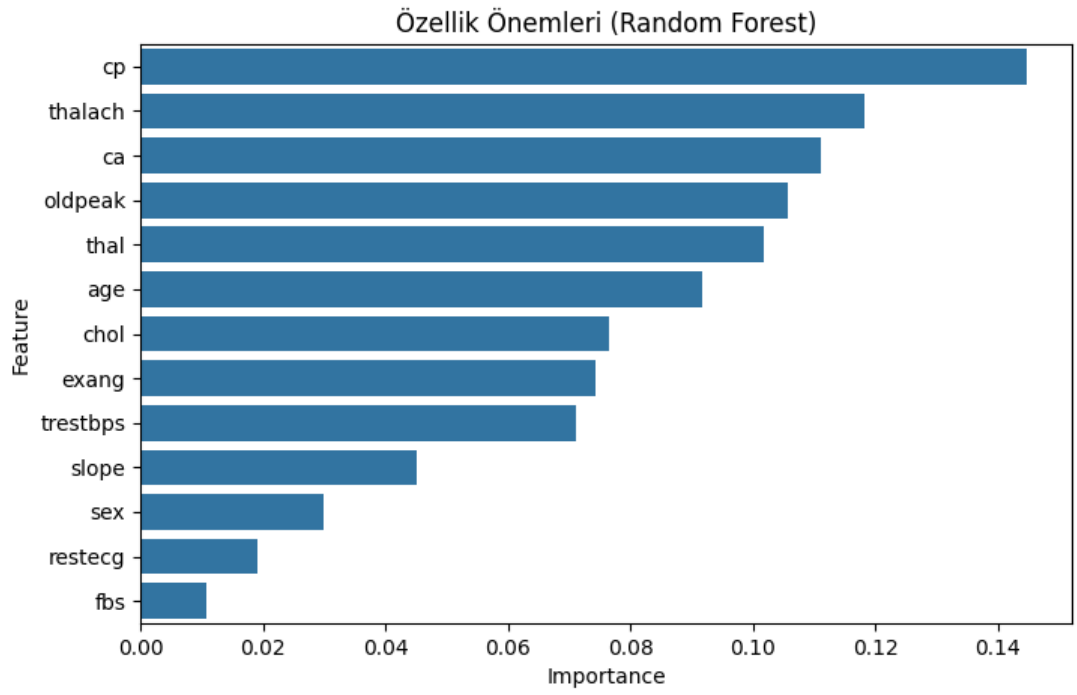
```
importances = rf_model.feature_importances_
feature_names = X.columns

feat_imp = pd.DataFrame({
    "Feature": feature_names,
    "Importance": importances
}).sort_values(by="Importance", ascending=False)

print("\n--- Özellik Önemleri (Feature Importance) ---")
print(feat_imp)

plt.figure(figsize=(8,5))
sns.barplot(x="Importance", y="Feature", data=feat_imp)
plt.title("Özellik Önemleri (Random Forest)")
plt.show()
```

Çıktı:

--- Özellik Önemleri (Feature Importance) ---

	Feature	Importance
2	cp	0.144796
7	thalach	0.118097
11	ca	0.110965
9	oldpeak	0.105590
12	thal	0.101825
0	age	0.091826
4	chol	0.076579
8	exang	0.074189
3	trestbps	0.071178
10	slope	0.045234
1	sex	0.029885
6	restecg	0.019166
5	fbs	0.010670

Tablo 7: Özellik Önemleri



Şekil 5: Özellik Önemleri

KAYNAKÇA

- https://www.kaggle.com/code/prthmgoyl/neuralnetwork-heart-disease-dataset/input