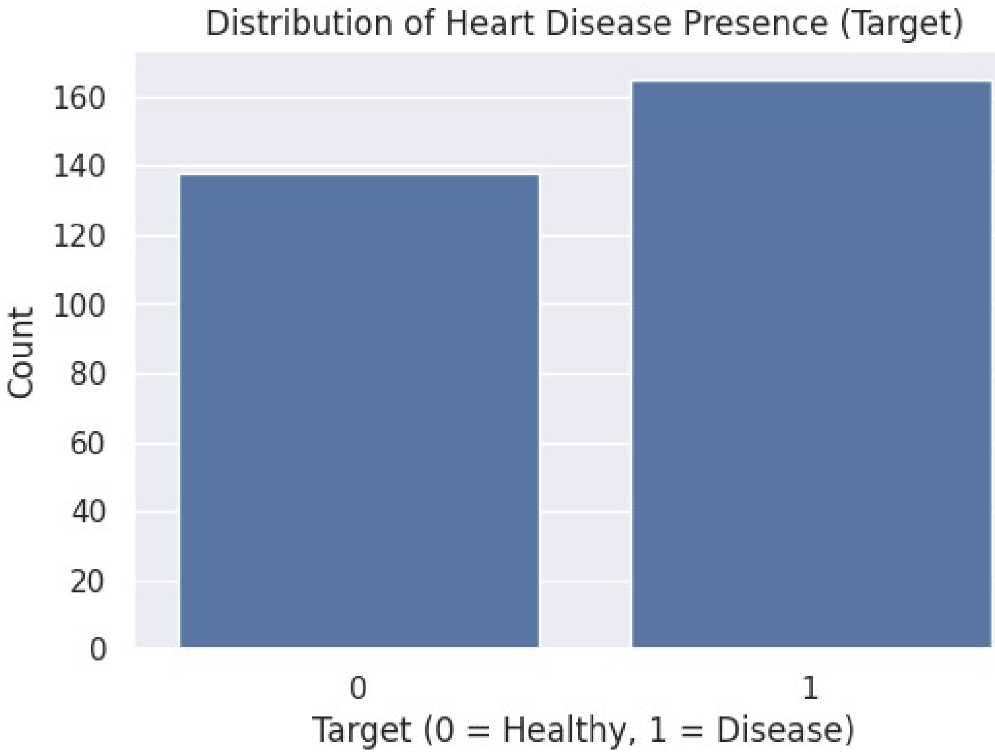
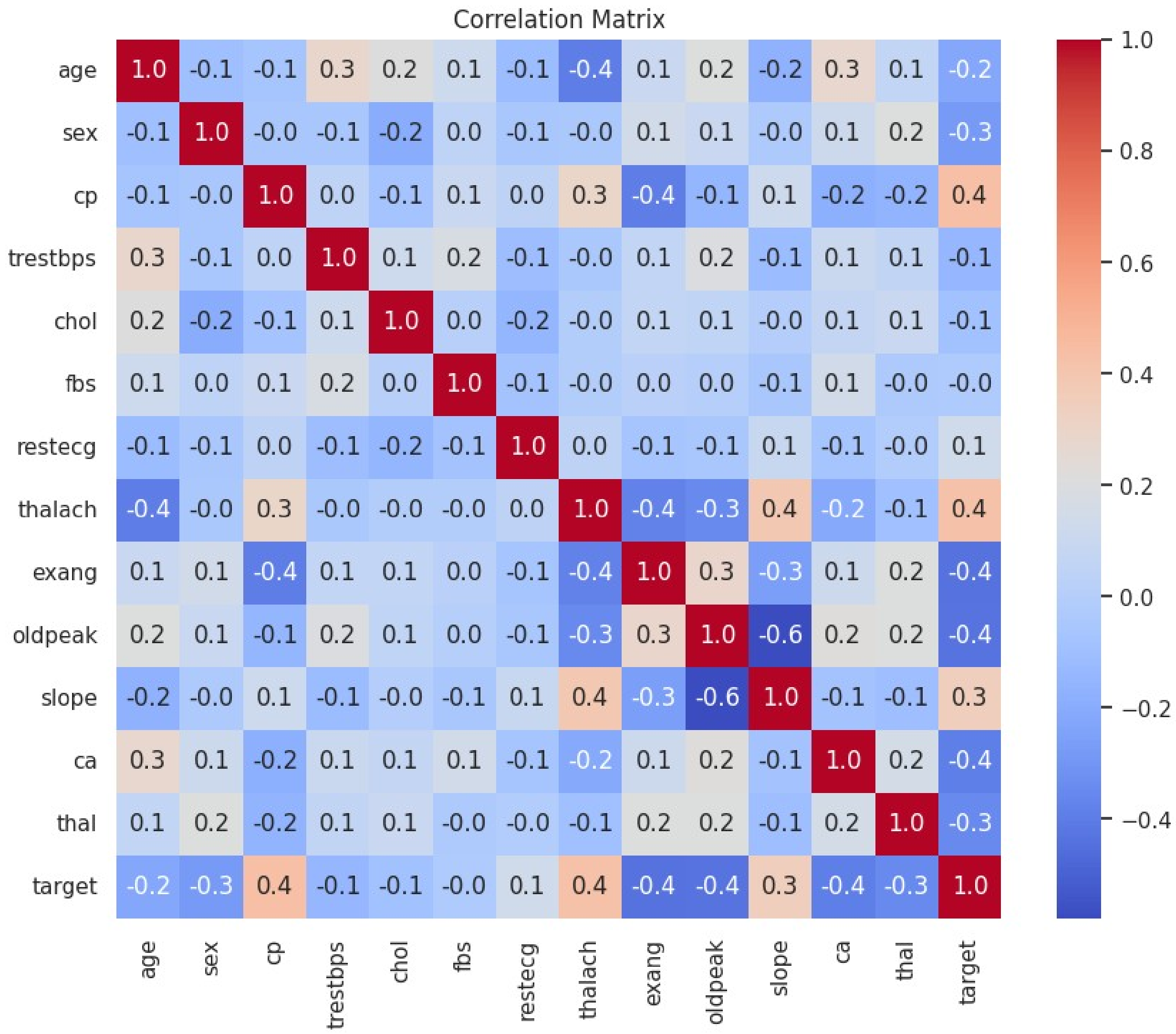
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix

# 1. Load the Data
# -----------------
try:
    df = pd.read_csv('heart_disease_data.csv')
    print("Data loaded successfully.")
    print(f"Dataset Shape: {df.shape}")
except FileNotFoundError:
    print("Error: The file 'heart_disease_data.csv' was not found.")
    exit()

# Display the first 5 rows
print("\n--- First 5 Rows ---")
print(df.head())

# Display basic statistical details
print("\n--- Data Description ---")
print(df.describe())

# 2. Data Visualization
# ----------------------
# Set the style for seaborn
sns.set()

# Plot the distribution of the 'target' variable
plt.figure(figsize=(6, 4))
sns.countplot(x='target', data=df)
plt.title('Distribution of Heart Disease Presence (Target)')
plt.xlabel('Target (0 = Healthy, 1 = Disease)')
plt.ylabel('Count')
plt.savefig('target_distribution.png')
plt.show()

# Plot a Correlation Heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm', fmt='.1f')
plt.title('Correlation Matrix')
plt.savefig('correlation_matrix.png')
plt.show()

# 3. Data Preprocessing & Splitting
# ----------------------------------
# Separating features (X) and target label (Y)
X = df.drop(columns='target', axis=1)
Y = df['target']

# Split data into Training and Testing sets (80% Train, 20% Test)
X_train, X_test, Y_train, Y_test = train_test_split(X, Y,
test_size=0.2, stratify=Y, random_state=2)

print(f"\nTraining Set Size: {X_train.shape}")
print(f"Testing Set Size: {X_test.shape}")

# 4. Model Training (Logistic Regression)
# ---------------------------------------
# Logistic Regression is used for binary classification problems
model = LogisticRegression(max_iter=1000)
model.fit(X_train, Y_train)

# 5. Model Evaluation
# -------------------
# Accuracy on training data
X_train_prediction = model.predict(X_train)
training_data_accuracy = accuracy_score(X_train_prediction, Y_train)

# Accuracy on test data
X_test_prediction = model.predict(X_test)
test_data_accuracy = accuracy_score(X_test_prediction, Y_test)

print("\n--- Model Performance ---")
print(f"Accuracy on Training data: {training_data_accuracy:.2%}")
print(f"Accuracy on Test data:     {test_data_accuracy:.2%}")

print("\n--- Classification Report ---")
print(classification_report(Y_test, X_test_prediction))

print("\n--- Confusion Matrix ---")
print(confusion_matrix(Y_test, X_test_prediction))
```

```
Data loaded successfully.
Dataset Shape: (303, 14)

--- First 5 Rows ---
   age  sex  cp  trestbps  chol  fbs  ...  exang  oldpeak  slope  ca
thal  target
0   63    1   3       145   233    1  ...      0      2.3      0   0
1       1
1   37    1   2       130   250    0  ...      0      3.5      0   0
2       1
2   41    0   1       130   204    0  ...      0      1.4      2   0
2       1
3   56    1   1       120   236    0  ...      0      0.8      2   0
```

```
2       1
4   57    0   0       120   354    0  ...      1      0.6      2   0
2       1

[5 rows x 14 columns]

--- Data Description ---
              age         sex          cp  ...          ca        thal
target
count  303.000000  303.000000  303.000000  ...  303.000000  303.000000
303.000000
mean    54.366337    0.683168    0.966997  ...    0.729373    2.313531
0.544554
std      9.082101    0.466011    1.032052  ...    1.022606    0.612277
0.498835
min     29.000000    0.000000    0.000000  ...    0.000000    0.000000
0.000000
25%     47.500000    0.000000    0.000000  ...    0.000000    2.000000
0.000000
50%     55.000000    1.000000    1.000000  ...    0.000000    2.000000
1.000000
75%     61.000000    1.000000    2.000000  ...    1.000000    3.000000
1.000000
max     77.000000    1.000000    3.000000  ...    4.000000    3.000000
1.000000

[8 rows x 14 columns]
```


Distribution of Heart Disease Presence (Target)
Count
0
20
40
60
80
100
120
140
160
0
1
Target (0 = Healthy, 1 = Disease)



```
Training Set Size: (242, 13)
Testing Set Size: (61, 13)

--- Model Performance ---
Accuracy on Training data: 85.54%
Accuracy on Test data:     80.33%

--- Classification Report ---
              precision    recall  f1-score   support

           0       0.79      0.79      0.79        28
           1       0.82      0.82      0.82        33

    accuracy                           0.80        61
   macro avg       0.80      0.80      0.80        61
weighted avg       0.80      0.80      0.80        61
```

```
--- Confusion Matrix ---
[[22  6]
 [ 6 27]]
```