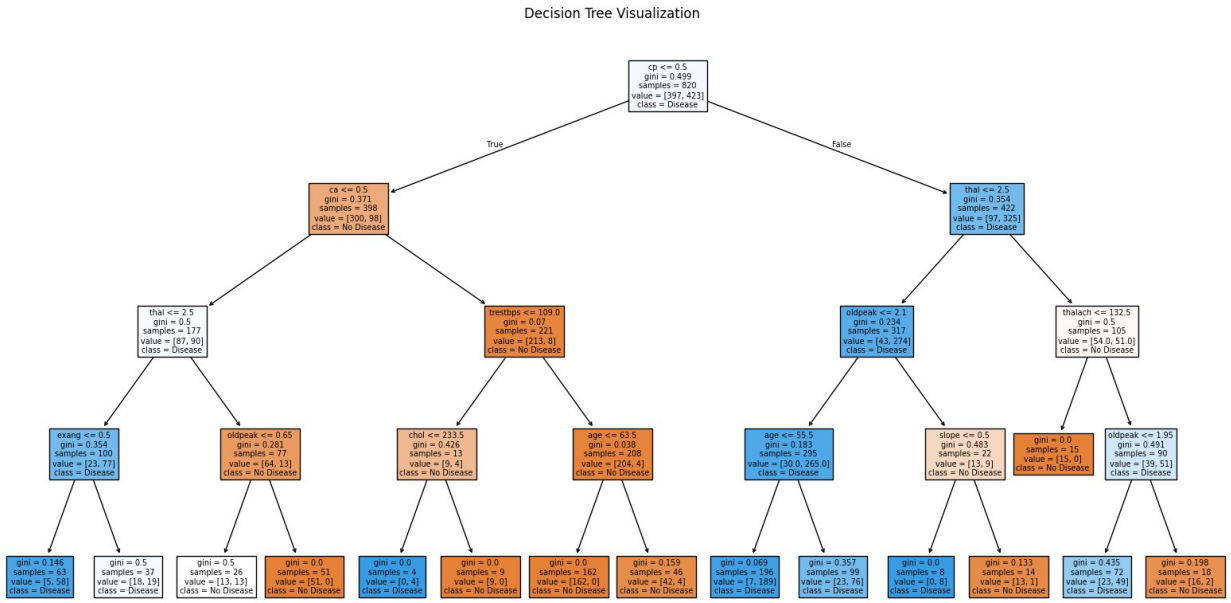
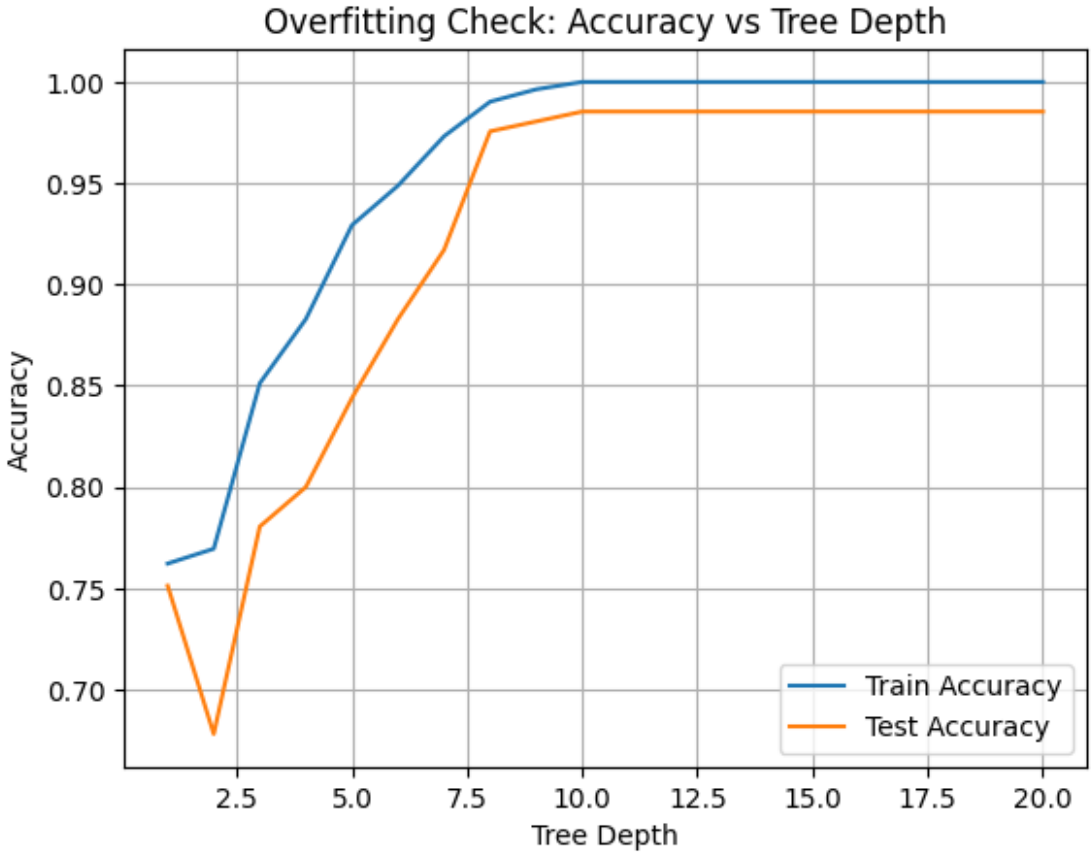
```python
# 📦 Import Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix

# 📂 Load Dataset
df = pd.read_csv("heart.csv")  # Make sure heart.csv is in the working
directory
print("Dataset Shape:", df.shape)
print(df.head())

# 🔍 Check for missing values
print("\nMissing values:\n", df.isnull().sum())

# 🎯 Define Features and Target
X = df.drop('target', axis=1)
y = df['target']

# 🔀 Train-Test Split
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# 🌳 Train Decision Tree Classifier
dt_model = DecisionTreeClassifier(max_depth=4, random_state=42)
dt_model.fit(X_train, y_train)
y_pred_dt = dt_model.predict(X_test)

# 📊 Accuracy & Evaluation - Decision Tree
print("\nDecision Tree Accuracy:", accuracy_score(y_test, y_pred_dt))
print("Classification Report:\n", classification_report(y_test,
y_pred_dt))

# 🌲 Visualize Decision Tree
plt.figure(figsize=(20,10))
plot_tree(dt_model, feature_names=X.columns, class_names=['No
Disease', 'Disease'], filled=True)
plt.title("Decision Tree Visualization")
plt.show()

# 🔁 Overfitting Analysis - Vary Tree Depth
train_acc = []
test_acc = []
depths = range(1, 21)
```

```
for d in depths:
    model = DecisionTreeClassifier(max_depth=d, random_state=42)
    model.fit(X_train, y_train)
    train_acc.append(model.score(X_train, y_train))
    test_acc.append(model.score(X_test, y_test))

plt.plot(depths, train_acc, label="Train Accuracy")
plt.plot(depths, test_acc, label="Test Accuracy")
plt.xlabel("Tree Depth")
plt.ylabel("Accuracy")
plt.title("Overfitting Check: Accuracy vs Tree Depth")
plt.legend()
plt.grid()
plt.show()

# □ Train Random Forest Classifier
rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)
y_pred_rf = rf_model.predict(X_test)

# □ Accuracy & Evaluation - Random Forest
print("\nRandom Forest Accuracy:", accuracy_score(y_test, y_pred_rf))
print("Classification Report:\n", classification_report(y_test,
y_pred_rf))

# □ Feature Importance from Random Forest
importances = pd.Series(rf_model.feature_importances_,
index=X.columns)
importances.sort_values(ascending=False).plot(kind='bar',
figsize=(10,5), title="Feature Importances")
plt.ylabel("Importance Score")
plt.show()

# □ Cross-Validation Accuracy
cv_scores = cross_val_score(rf_model, X, y, cv=5)
print("\nRandom Forest Cross-Validation Accuracy: {:.2f}
%".format(np.mean(cv_scores) * 100))
```

```
Dataset Shape: (1025, 14)
   age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak
slope  \
0   52    1   0       125   212    0        1      168      0      1.0
2
1   53    1   0       140   203    1        0      155      1      3.1
0
2   70    1   0       145   174    0        1      125      1      2.6
0
3   61    1   0       148   203    0        1      161      0      0.0
2
```

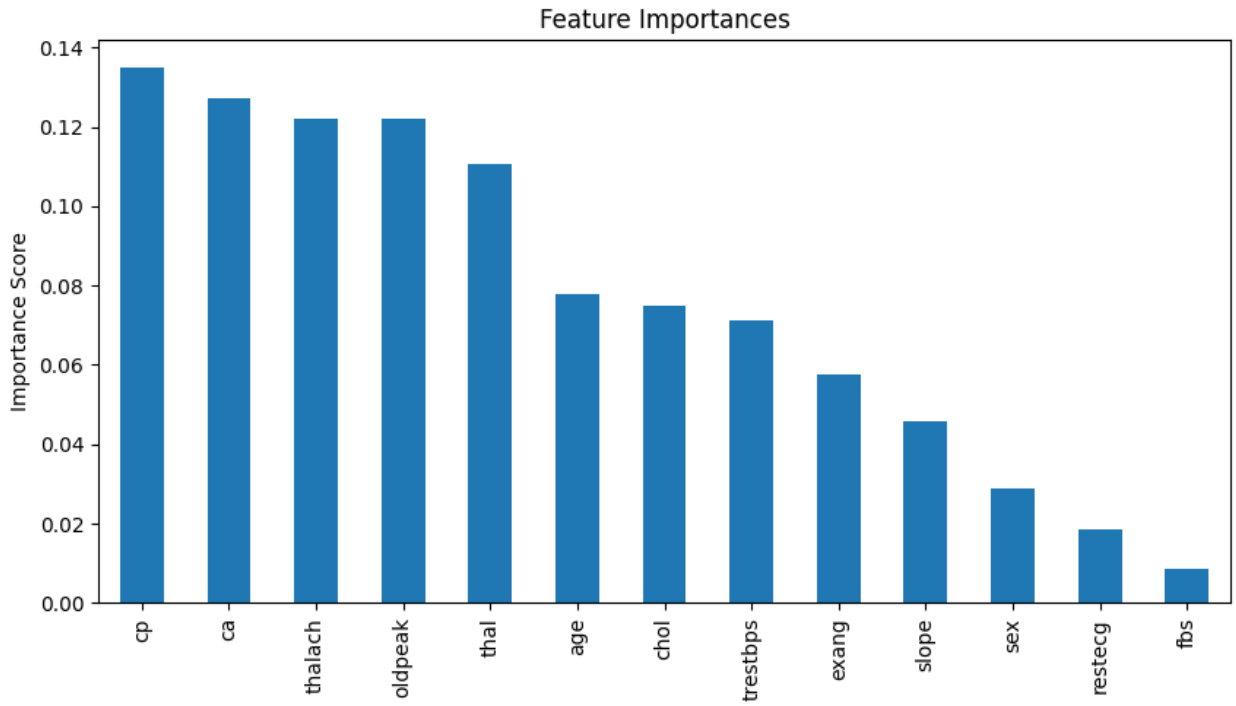
```
4   62    0   0       138   294    1        1      106      0      1.9
1

   ca  thal  target
0   2     3       0
1   0     3       0
2   0     3       0
3   1     3       0
4   3     2       0

Missing values:
 age         0
sex          0
cp           0
trestbps     0
chol         0
fbs          0
restecg      0
thalach      0
exang        0
oldpeak      0
slope        0
ca           0
thal         0
target       0
dtype: int64

Decision Tree Accuracy: 0.8
Classification Report:
               precision    recall  f1-score   support

           0       0.88      0.70      0.78       102
           1       0.75      0.90      0.82       103

    accuracy                           0.80       205
   macro avg       0.81      0.80      0.80       205
weighted avg       0.81      0.80      0.80       205
```






```
Random Forest Accuracy: 0.9853658536585366
Classification Report:
```

```
                precision    recall  f1-score   support

           0       0.97      1.00      0.99       102
           1       1.00      0.97      0.99       103

    accuracy                           0.99       205
   macro avg       0.99      0.99      0.99       205
weighted avg       0.99      0.99      0.99       205
```




```
Random Forest Cross-Validation Accuracy: 99.71%
```