```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report, roc_curve, auc
import matplotlib.pyplot as plt
import seaborn as sns

# Step 2: Load the dataset
# You can download the dataset from: https://www.kaggle.com/ronitf/heart-disease-uci
df = pd.read_csv("heart.csv")

# Step 3: Explore the dataset
print(df.head())
print(df.info())
print(df.describe())
print(df.isnull().sum())

# Step 4: Separate features and target
X = df.drop("target", axis=1)
y = df["target"]

# Step 5: Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Step 6: Feature scaling
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Step 7: Train the model (Random Forest Classifier)
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Step 8: Make predictions
y_pred = model.predict(X_test)
y_prob = model.predict_proba(X_test)[:,1]

# Step 9: Evaluate the model
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))

# Step 10: ROC Curve and AUC
fpr, tpr, thresholds = roc_curve(y_test, y_prob)
```

```
roc_auc = auc(fpr, tpr)

plt.figure(figsize=(8,6))
plt.plot(fpr, tpr, color='darkorange', lw=2, label='ROC curve (area = %0.2f)' % roc_auc)
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic - Heart Disease')
plt.legend(loc="lower right")
plt.show()

# Step 11: Predict for new data
# Example new patient data: [age, sex, cp, trestbps, chol, fbs, restecg, thalach, exang, oldpeak, slope, ca, thal]
new_data = np.array([[63,1,3,145,233,1,0,150,0,2.3,0,0,1]])
new_data = scaler.transform(new_data)
prediction = model.predict(new_data)
print("Heart Disease Prediction (0=No, 1=Yes):", prediction[0])
```