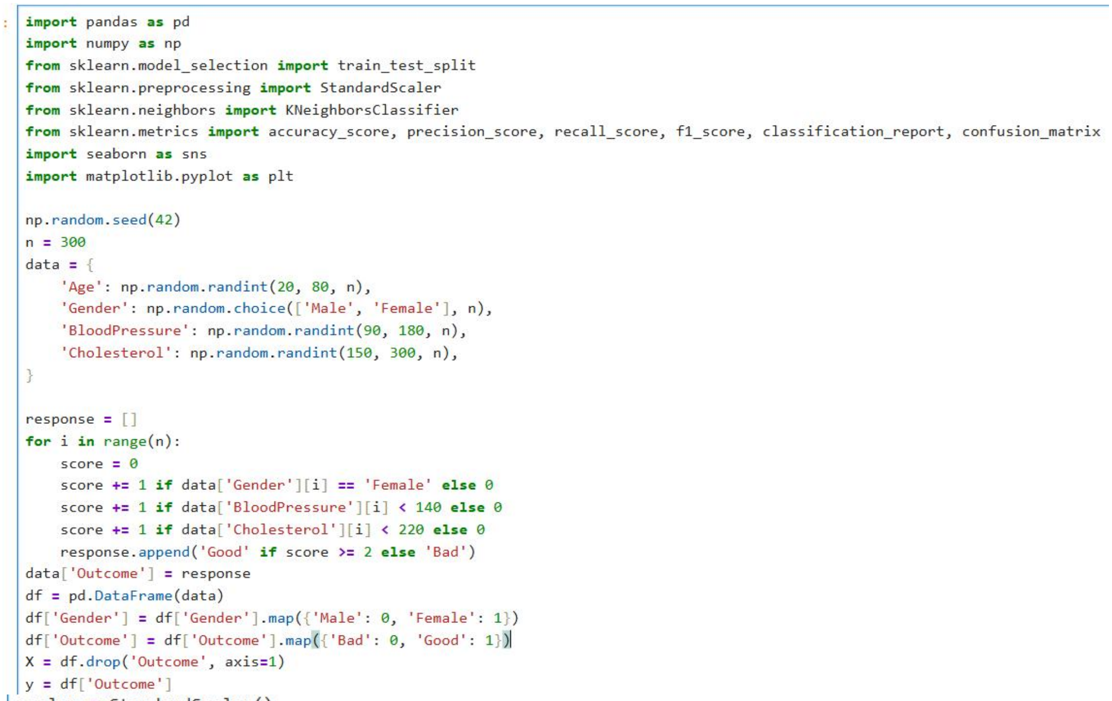
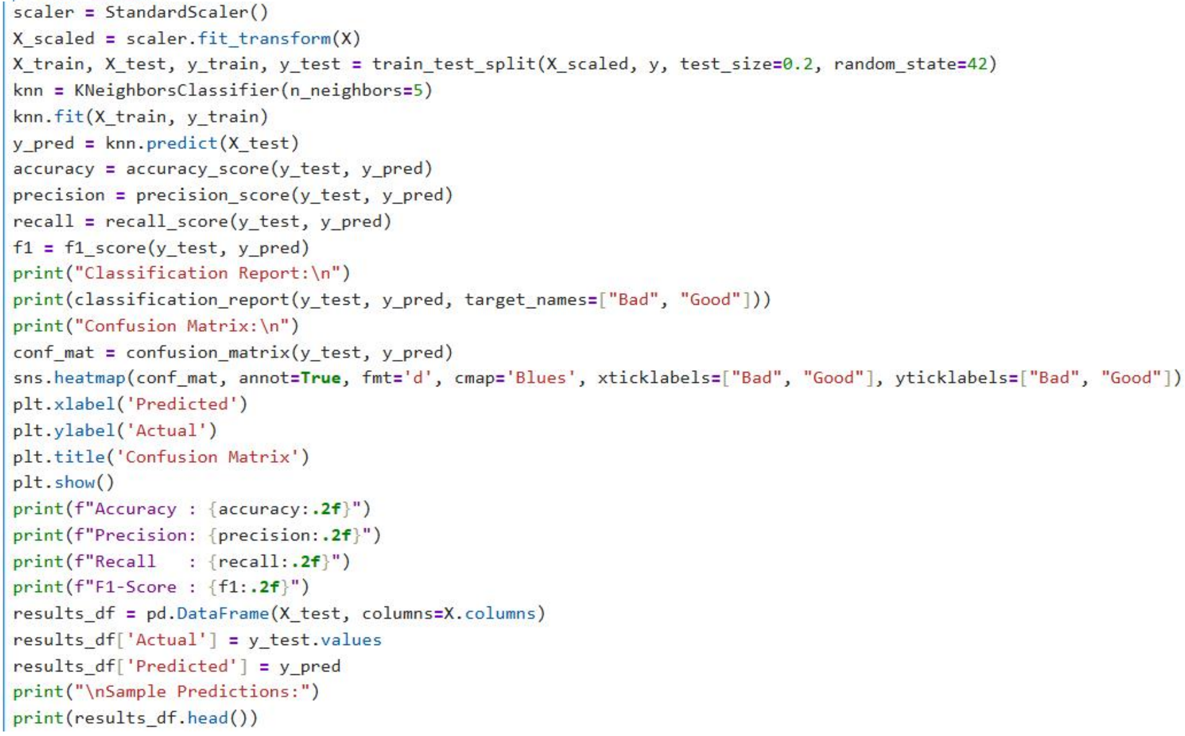


```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, classification_report, confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt

np.random.seed(42)
n = 300
data = {
    'Age': np.random.randint(20, 80, n),
    'Gender': np.random.choice(['Male', 'Female'], n),
    'BloodPressure': np.random.randint(90, 180, n),
    'Cholesterol': np.random.randint(150, 300, n),
}

response = []
for i in range(n):
    score = 0
    score += 1 if data['Gender'][i] == 'Female' else 0
    score += 1 if data['BloodPressure'][i] < 140 else 0
    score += 1 if data['Cholesterol'][i] < 220 else 0
    response.append('Good' if score >= 2 else 'Bad')
data['Outcome'] = response
df = pd.DataFrame(data)
df['Gender'] = df['Gender'].map({'Male': 0, 'Female': 1})
df['Outcome'] = df['Outcome'].map({'Bad': 0, 'Good': 1})
X = df.drop('Outcome', axis=1)
y = df['Outcome']
```



```
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)
knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train, y_train)
y_pred = knn.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
print("Classification Report:\n")
print(classification_report(y_test, y_pred, target_names=["Bad", "Good"]))
print("Confusion Matrix:\n")
conf_mat = confusion_matrix(y_test, y_pred)
sns.heatmap(conf_mat, annot=True, fmt='d', cmap='Blues', xticklabels=["Bad", "Good"], yticklabels=["Bad", "Good"])
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()
print(f"Accuracy : {accuracy:.2f}")
print(f"Precision: {precision:.2f}")
print(f"Recall   : {recall:.2f}")
print(f"F1-Score : {f1:.2f}")
results_df = pd.DataFrame(X_test, columns=X.columns)
results_df['Actual'] = y_test.values
results_df['Predicted'] = y_pred
print("\nSample Predictions:")
print(results_df.head())
```

OUTPUT

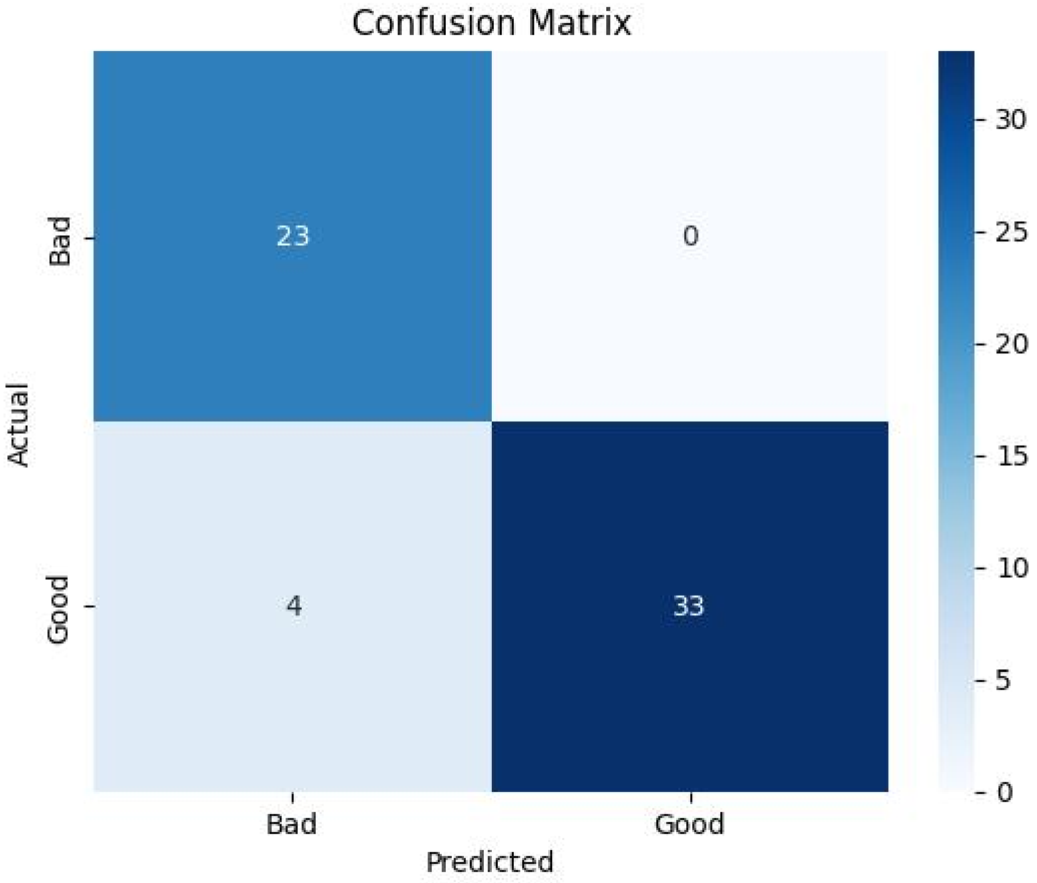
```
Classification Report:

              precision    recall  f1-score   support

         Bad       0.85      1.00      0.92        23
        Good       1.00      0.89      0.94        37

    accuracy                           0.93        60
   macro avg       0.93      0.95      0.93        60
weighted avg       0.94      0.93      0.93        60
```

Confusion Matrix:



```
Accuracy : 0.93
Precision: 1.00
Recall   : 0.89
F1-Score : 0.94

Sample Predictions:
        Age    Gender  BloodPressure  Cholesterol  Actual  Predicted
0 -0.502719 -1.083473      -0.409438     1.229183       0          0
1 -1.628630 -1.083473      -0.714609     0.557878       0          0
2  1.355033 -1.083473       1.154564    -1.085663       0          0
3 -0.671606  0.922958      -1.057927    -0.946772       1          1
4 -1.403447 -1.083473       1.192711     0.071760       0          0
```