```
# Import necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

# ------------------ Step 1: Load Dataset ------------------
file_path = '/content/adult.csv'  # update path as needed
data = pd.read_csv(file_path)

print("First 5 rows of the dataset:\n", data.head())

# ------------------ Step 2: BEFORE Preprocessing ------------------

data_before = data.dropna()

label_encoders = {}
for column in data_before.select_dtypes(include=['object']).columns:
    le = LabelEncoder()
    data_before[column] = le.fit_transform(data_before[column])
    label_encoders[column] = le

X = data_before.drop('income', axis=1)
y = data_before['income']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

model_before = RandomForestClassifier(random_state=42)
model_before.fit(X_train, y_train)
y_pred_before = model_before.predict(X_test)

accuracy_before = accuracy_score(y_test, y_pred_before)
print(f"\nModel Accuracy BEFORE Preprocessing: {accuracy_before:.4f}")

# ------------------ Step 3: AFTER Preprocessing ------------------

data = pd.read_csv(file_path)
data.replace('?', np.nan, inplace=True)
data.dropna(inplace=True)

for column in data.select_dtypes(include=['object']).columns:
    le = LabelEncoder()
    data[column] = le.fit_transform(data[column])

scaler = StandardScaler()
scaled_features = scaler.fit_transform(data.drop('income', axis=1))
X_scaled = pd.DataFrame(scaled_features, columns=data.columns[:-1])
y_scaled = data['income']

X_train_scaled, X_test_scaled, y_train_scaled, y_test_scaled = train_test_split(X_scaled, y_scaled, test_size

model_after = RandomForestClassifier(random_state=42)
model_after.fit(X_train_scaled, y_train_scaled)
y_pred_after = model_after.predict(X_test_scaled)

accuracy_after = accuracy_score(y_test_scaled, y_pred_after)
print(f"\nModel Accuracy AFTER Preprocessing: {accuracy_after:.4f}")

# ------------------ Step 4: Graphical Visualization ------------------

plt.figure(figsize=(8, 5))
plt.bar(['Before Preprocessing', 'After Preprocessing'], [accuracy_before, accuracy_after], color=['red', 'gr
plt.ylim(0, 1)
plt.title('Model Accuracy Comparison')
plt.ylabel('Accuracy Score')
plt.xlabel('Data Preprocessing Stage')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.text(0, accuracy_before + 0.02, f'{accuracy_before:.4f}', ha='center', fontsize=12)
plt.text(1, accuracy_after + 0.02, f'{accuracy_after:.4f}', ha='center', fontsize=12)
plt.tight_layout()
plt.show()
```

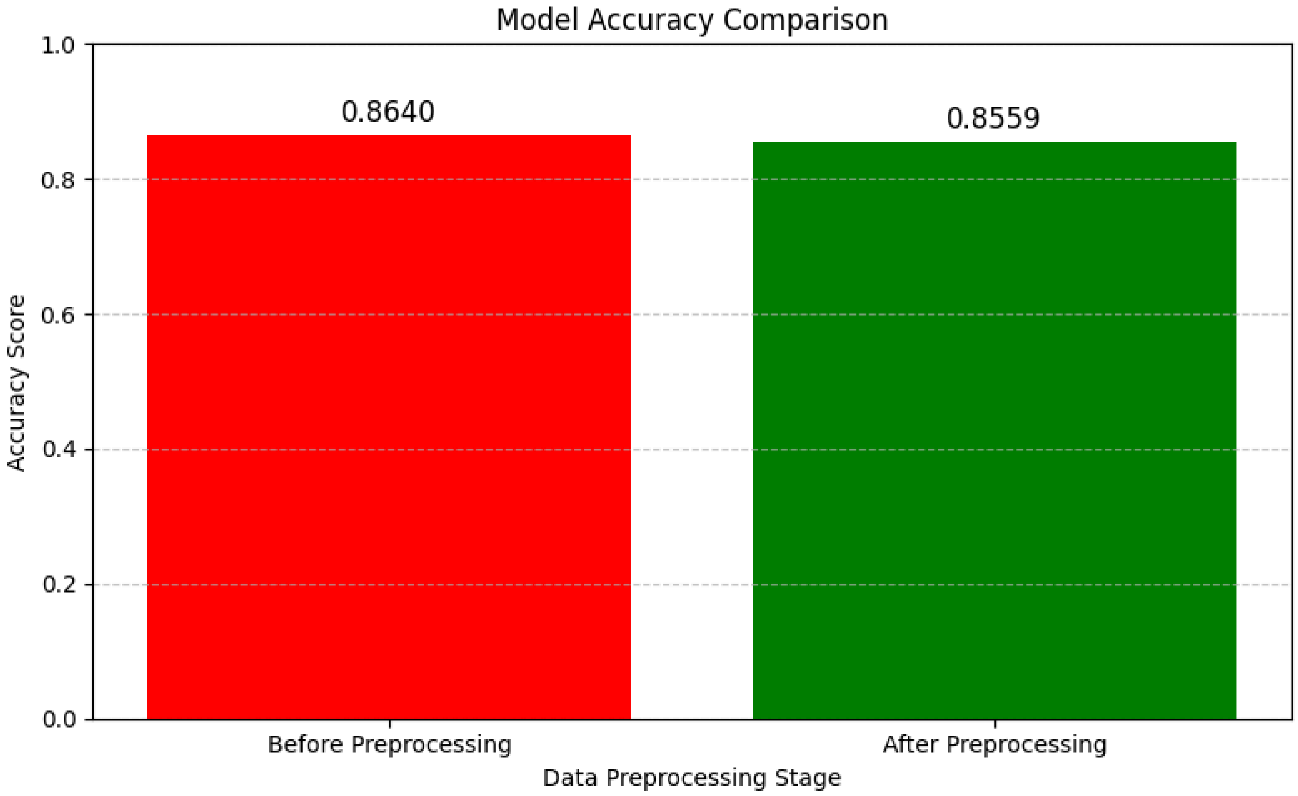
```
First 5 rows of the dataset:
    age  workclass  fnlwgt     education  educational-num      marital-status  \
0   25    Private  226802          11th                7       Never-married
1   38    Private   89814       HS-grad                9  Married-civ-spouse
2   28  Local-gov  336951    Assoc-acdm               12  Married-civ-spouse
3   44    Private  160323  Some-college               10  Married-civ-spouse
4   18          ?  103497  Some-college               10       Never-married

          occupation relationship   race  gender  capital-gain  capital-loss  \
0  Machine-op-inspct    Own-child  Black    Male             0             0
1    Farming-fishing      Husband  White    Male             0             0
2    Protective-serv      Husband  White    Male             0             0
3  Machine-op-inspct      Husband  Black    Male          7688             0
4                  ?    Own-child  White  Female             0             0

   hours-per-week native-country income
0              40  United-States  <=50K
1              50  United-States  <=50K
2              40  United-States   >50K
3              40  United-States   >50K
4              30  United-States  <=50K

Model Accuracy BEFORE Preprocessing: 0.8640

Model Accuracy AFTER Preprocessing: 0.8559
```


Model Accuracy Comparison
Accuracy Score
Before Preprocessing
After Preprocessing
Data Preprocessing Stage