```
#unzip dataset

import zipfile zip_ref =zipfile.ZipFile("/content/archive (8).zip",'r') zip_ref.extractall('/content')
zip_ref.close()


import pandas as pd

import train_test_split from sklearn.linear_model

import LogisticRegression from sklearn.metrics import accuracy_score, classification_report

df=pd.read_csv("heart.csv")

df


no_samples = len(df) no_samples

num_classes = df['target'].nunique() print("Number of classes:", num_classes)

df.info()

df.shape

df.isnull().sum()

X= df.drop('target',axis=1) #feature(x) y=df['target']  #labely

X


y

# Split the data into 50% training and 50% testing

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.5, random_state=40)


# Initialize and train the classifier

from sklearn.ensemble import RandomForestClassifier model =
RandomForestClassifier(random_state=42) model.fit(X_train, y_train)

# Make predictions

y_pred = model.predict(X_test)

# Evaluate the model

accuracy = accuracy_score(y_test, y_pred) report = classification_report(y_test, y_pred)

print(f'Accuracy: {accuracy:.2f}') print('Classification Report:') print(report)


from sklearn.preprocessing import MinMaxScaler, StandardScaler, Normalizer
```

```
# Min-Max Scaling

min_max_scaler = MinMaxScaler()

X_train_minmax = min_max_scaler.fit_transform(X_train)

X_test_minmax = min_max_scaler.transform(X_test)

# Train Random Forest with Min-Max scaled data rf_model_minmax =
RandomForestClassifier(random_state=42) rf_model_minmax.fit(X_train_minmax, y_train)
y_pred_minmax = rf_model_minmax.predict(X_test_minmax)

# Evaluate Min-Max Random Forest

 accuracy_minmax = accuracy_score(y_test, y_pred_minmax) report_minmax =
classification_report(y_test, y_pred_minmax)

print("Min-Max Scaling:") print(f'Accuracy: {accuracy_minmax:.2f}') print('Classification Report:')
print(report_minmax)


standard_scaler = StandardScaler()

X_train_standard = standard_scaler.fit_transform(X_train)

X_test_standard = standard_scaler.transform(X_test)

# Train Random Forest with Z-score standardized data rf_model_standard =
RandomForestClassifier(random_state=42) rf_model_standard.fit(X_train_standard, y_train)
y_pred_standard = rf_model_standard.predict(X_test_standard)

# Evaluate Z-score Random Forest

 accuracy_standard=accuracy_score(y_test,y_pred_standard)

report_standard=classification_report(y_test,y_pred_standard)

print("\nZ-score Standardization:") print(f'Accuracy: {accuracy_standard:.2f}') print('Classification
Report:') print(report_standard)
```