

```
In [1]: from google.colab import drive
drive.mount('/content/drive')

Mounted at /content/drive

DATA PREPROCESSING

In [2]: # Import necessary libraries
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.preprocessing import StandardScaler

file_path = '/content/drive/MyDrive/CSV/framingham.csv'
df = pd.read_csv(file_path)

# Display the head of the dataframe to understand its structure
print(df.head())

# Preprocess the data
# Drop rows with missing values
df.dropna(inplace=True)

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

# Convert categorical variables to dummy variables
X = pd.get_dummies(X, drop_first=True)

# 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)

# Standardize the features
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

male age education currentSmoker cigsPerDay BPMeds prevalentStroke \
0 1 39 4.0 0 0.0 0.0 0
1 0 46 2.0 0 0.0 0.0 0
2 1 48 1.0 1 20.0 0.0 0
3 0 61 3.0 1 30.0 0.0 0
4 0 46 3.0 1 23.0 0.0 0

prevalentHyp diabetes totChol sysBP diaBP BMI heartRate glucose \
0 0 0 195.0 106.0 70.0 26.97 80.0 77.0
1 0 0 250.0 121.0 81.0 28.73 95.0 76.0
2 0 0 245.0 127.5 80.0 25.34 75.0 70.0
3 1 0 225.0 150.0 95.0 28.58 65.0 103.0
4 0 0 285.0 130.0 84.0 23.10 85.0 85.0

TenYearCHD
0 0
1 0
2 0
3 1
4 0

MODEL TRAINING

In [3]: # Train the model
model = LogisticRegression()
model.fit(X_train, y_train)

Out[3]:
▼ LogisticRegression
LogisticRegression()
```

```
MODEL TESTING

In [5]: # Test the model
y_pred = model.predict(X_test)

# Evaluate the model
print("Confusion Matrix:")
print(confusion_matrix(y_test, y_pred))
print("Classification Report:")
print(classification_report(y_test, y_pred))

# Save the model and scaler for future use
import joblib
joblib.dump(model, 'logistic_regression_model.pkl')
joblib.dump(scaler, 'scaler.pkl')

print('Analysis completed!')

Confusion Matrix:
[[606  4]
 [113  9]]
Classification Report:
              precision    recall  f1-score   support

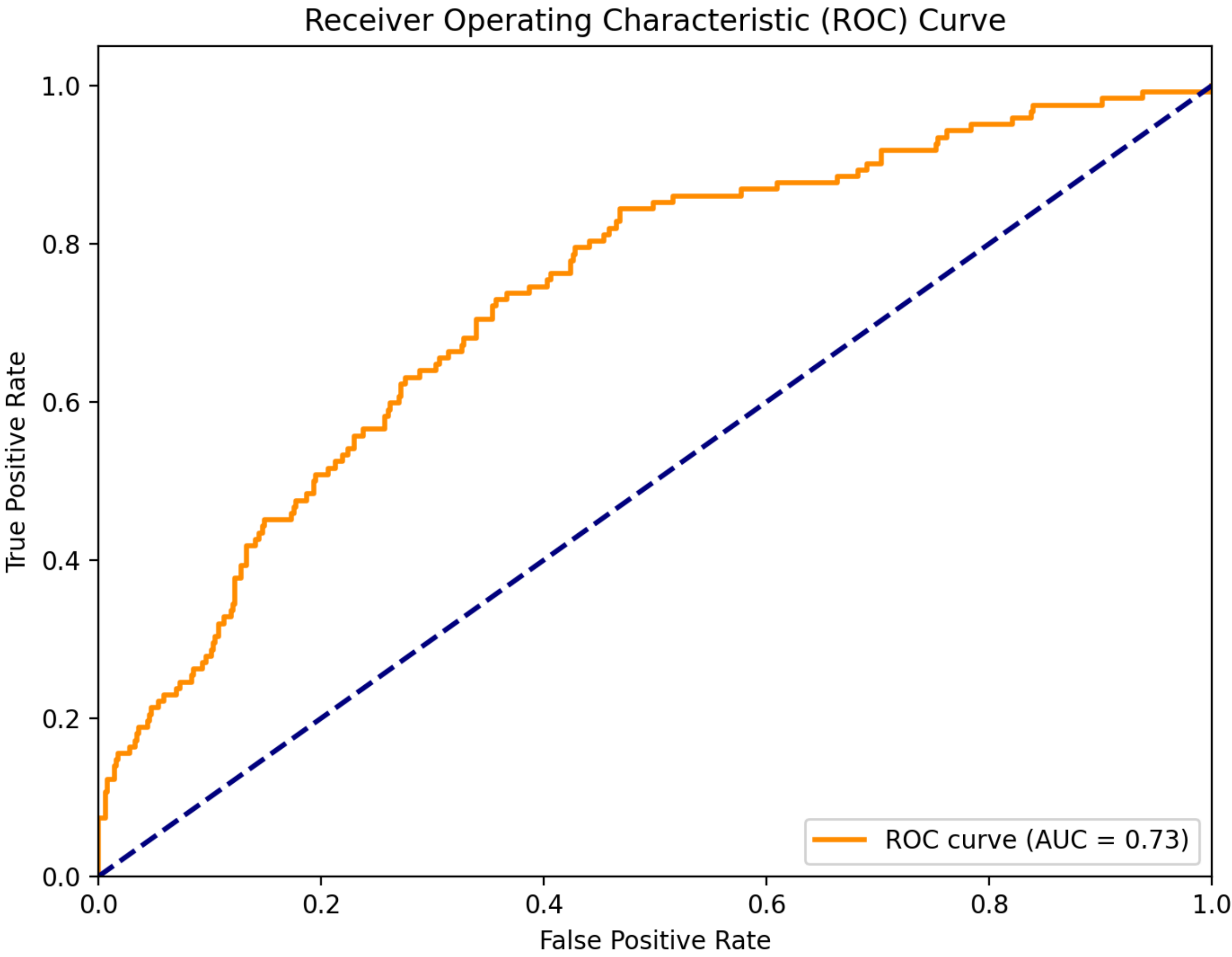
      0       0.84      0.99      0.91        610
      1       0.69      0.07      0.13        122

 accuracy      0.77
 macro avg     0.77      0.53      0.52        732
weighted avg     0.82      0.84      0.78        732

Analysis completed!

In [6]: from IPython.display import Image
Image('/content/drive/MyDrive/CSV/chart (1).png')
```

Out[6]:



```
In [7]: from IPython.display import Image
Image('/content/drive/MyDrive/CSV/chart (2).png')
```

Out[7]:

