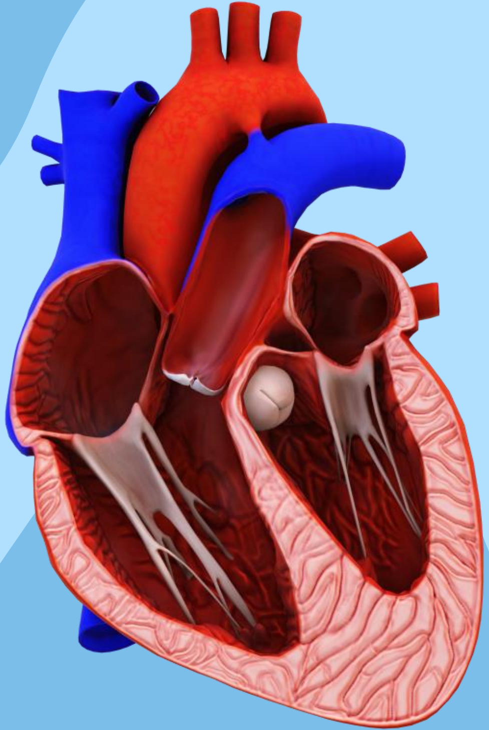


HEART DISEASE PREDICTION

USING MACHINE LEARNING

PROBLEM STATEMENT :

The Goal of this project is to develop an accurate and efficient HEART DISEASE PREDICTION using machine learning techniques

APPROACH:

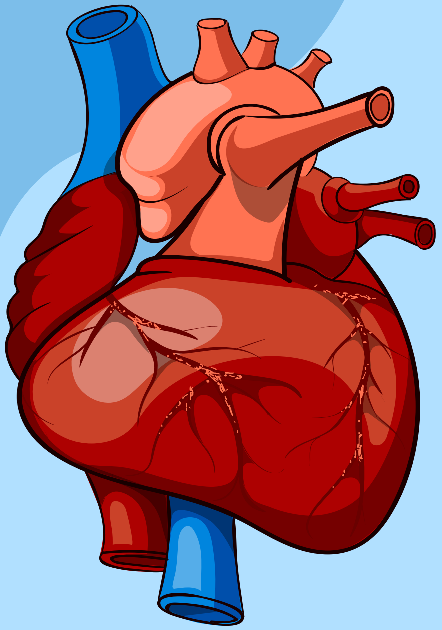
Algorithm : Logistic regression

Programming language : Python

Tools and Libraries : Jupyter Notebook,scikit-learn Numpy, Pandas, matplotlib, Sklearn .

Dataset : framingham.csv (from Kaggle website)

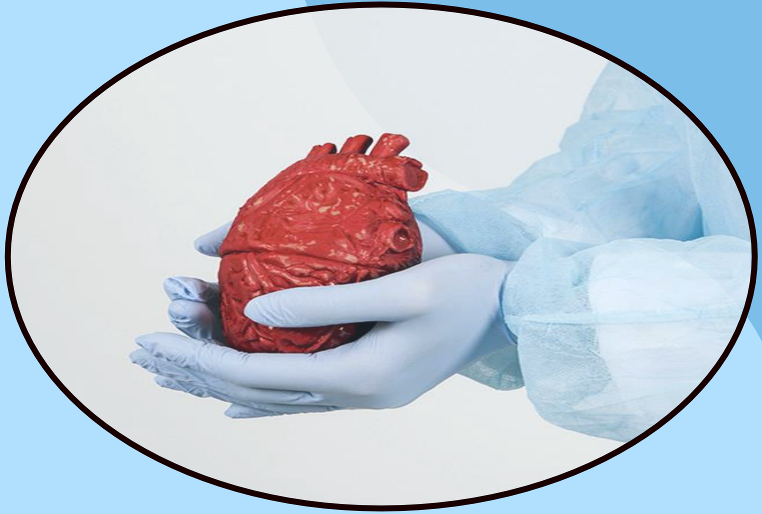
INTRODUCTION



Heart disease can be predicted based on various symptoms such as age, gender, heart rate, etc. and reduces the death rate of heart patients. Due to the increasing use of technology and data collection, we can now predict heart disease using **machine learning algorithms**.

Abstract:

Logistic regression models predicts the risk of suffering from heart disease among the elderly by exploring the feasibility of using logistic regression models. Through the technology of data mining, the main pathogenic factors of heart disease were found, and the incidence of heart disease was predicted by using the regression model. The accuracy of logistic regression model was compared with other explored algorithms, and I found that the logistic regression model was worthy of research in the field of heart disease prediction.



ALGORITHM:

LOGISTIC REGRESSION ALGORITHM:

Logistic Regression is a popular and widely used classification algorithm in machine learning. It is primarily used for binary classification tasks, where the target variable is categorical value (yes/no ,0/1,true/false,fraud/not fraud). Logistic Regression algorithm uses sigmoid function to map the predicted values to the probability of belonging to a particular class.

Logistic Function (Sigmoid Function):

•The sigmoid function is a mathematical function used to map the predicted values to probabilities.

•It maps any real value into another value within a range of 0 and 1.

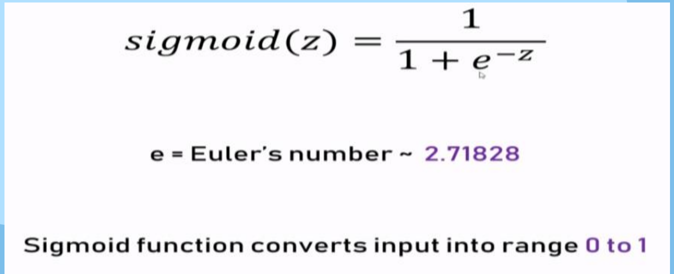
•The value of the logistic regression must be between 0 and 1, which cannot go beyond this limit,so it forms a curve like the “S” form. The S-form curve is called the Sigmoid function or the logistic function.

•In logistic regression, we use the concept of the threshold value, which defines the probability of either 0 or 1. Such as values above the threshold value tends to 1, and a value below thethreshold values tends to 0.

Assumptions for Logistic Regression:

•The dependent variable must be categorical in nature.

•The independent variable should not have multi-collinearity.

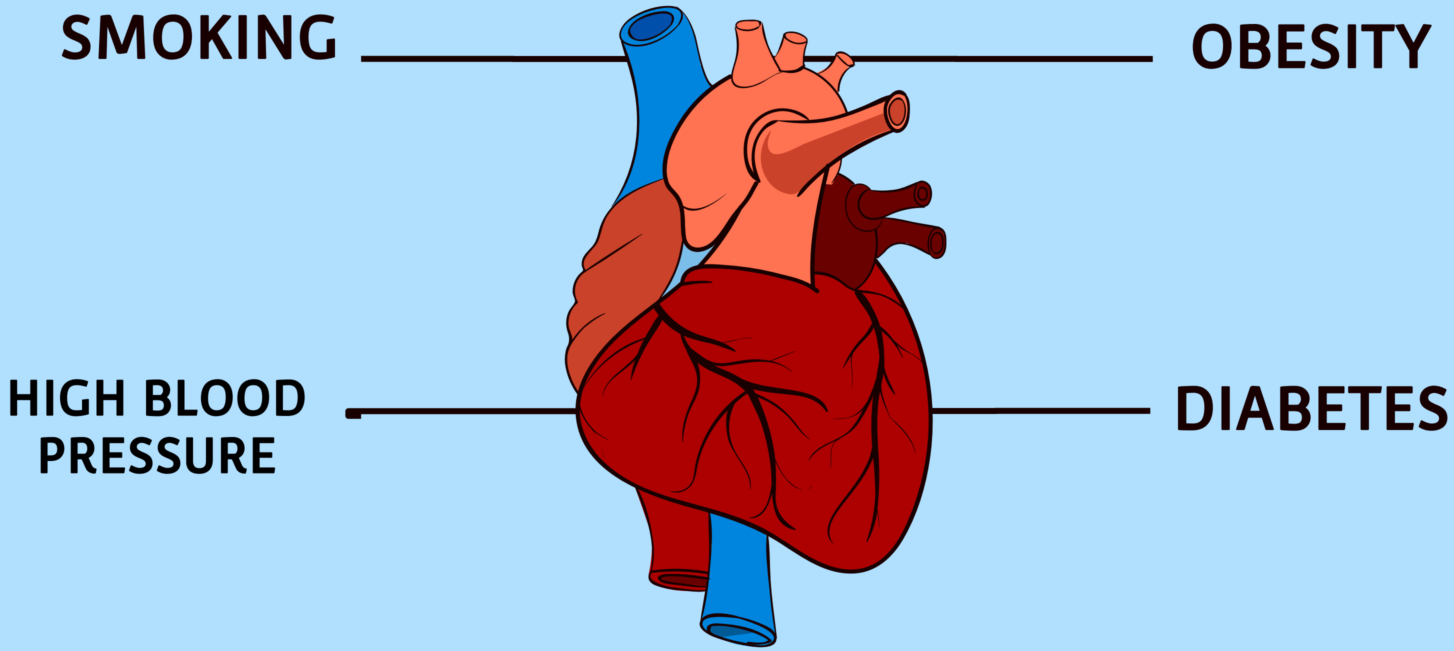


$$sigmoid(z) = \frac{1}{1 + e^{-z}}$$

e = Euler's number ~ 2.71828

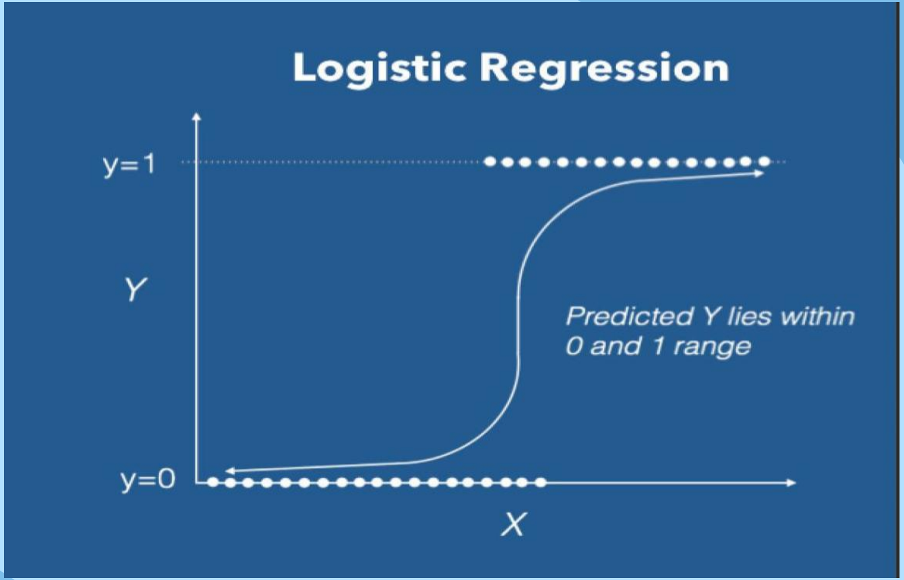
Sigmoid function converts input into range 0 to 1

CAUSES OF HEART DISEASE



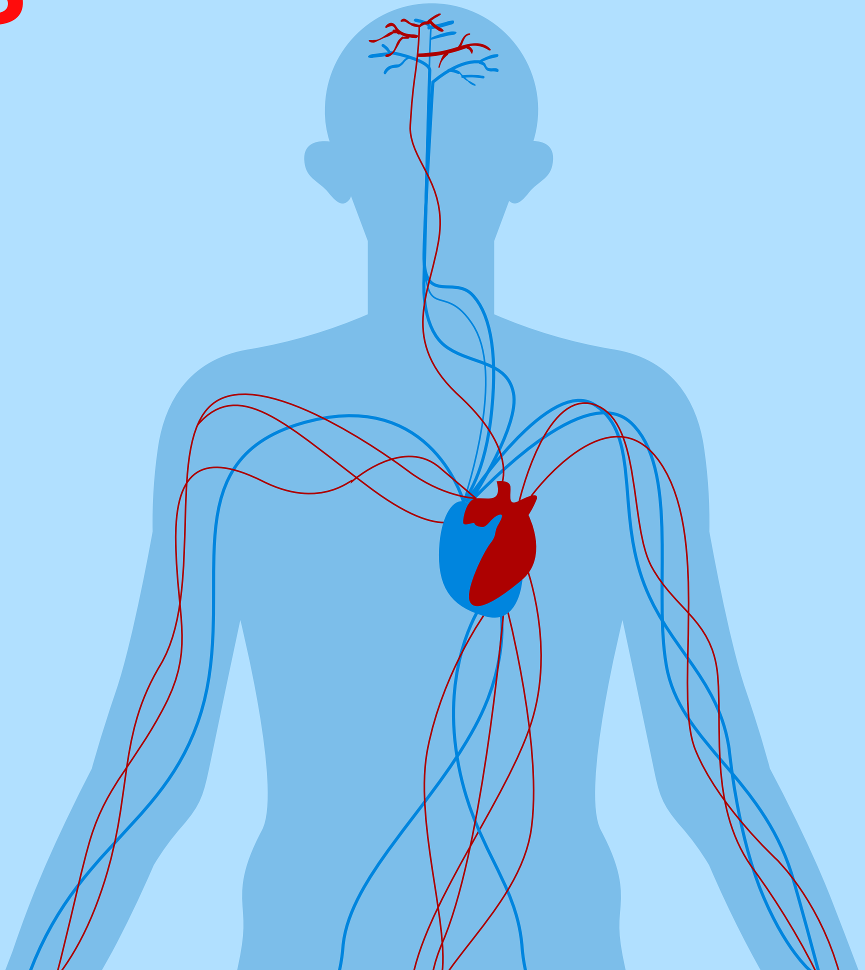
Steps in heart disease prediction

- Data Pre-processing step
- Fitting Logistic Regression to the Training set
- Predicting the test result
- Test accuracy of the result
- Visualizing the test set result.



DATASETS

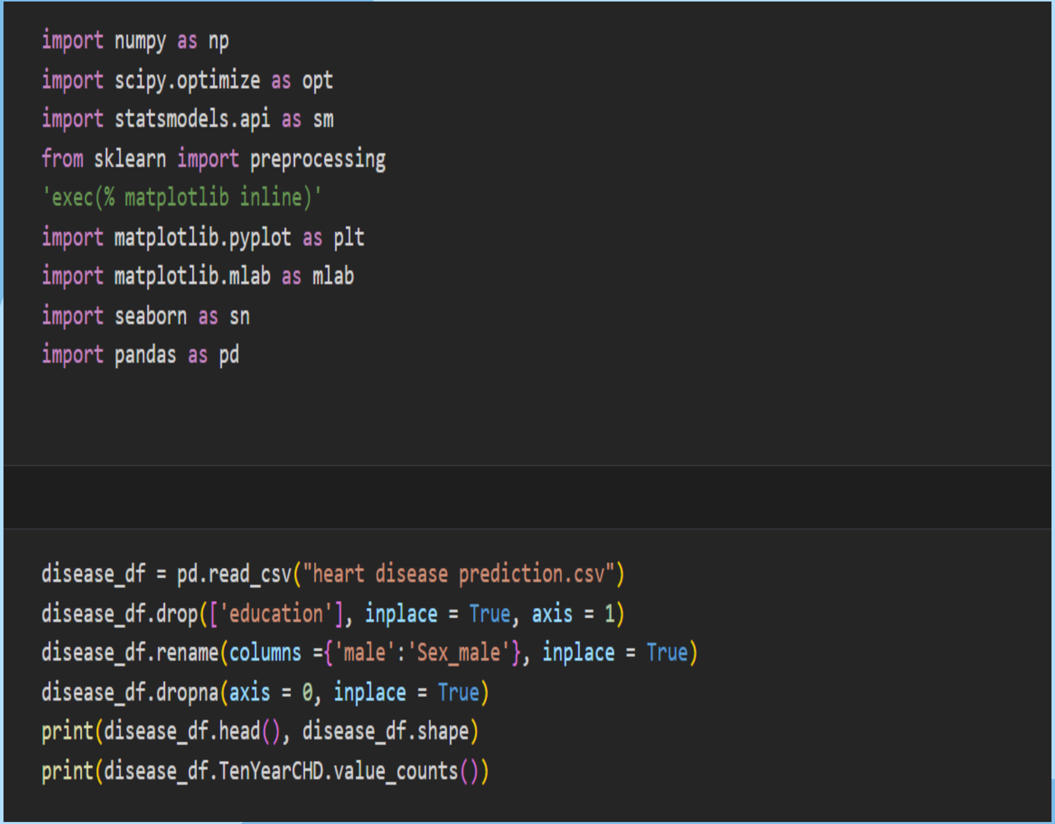
- Male
- Age
- Education
- CurrentSmoker
- CigsPerDay
- BPMeds
- PrevalentStroke
- PrevalentHyp
- diabetes
- totChol
- sysBP
- diaBP
- BMI
- heartRate
- glucose
- TenYearCHD



TRAINING MODEL

FIND THE ACCURACY AND PRECISION OF HEART DISEASE

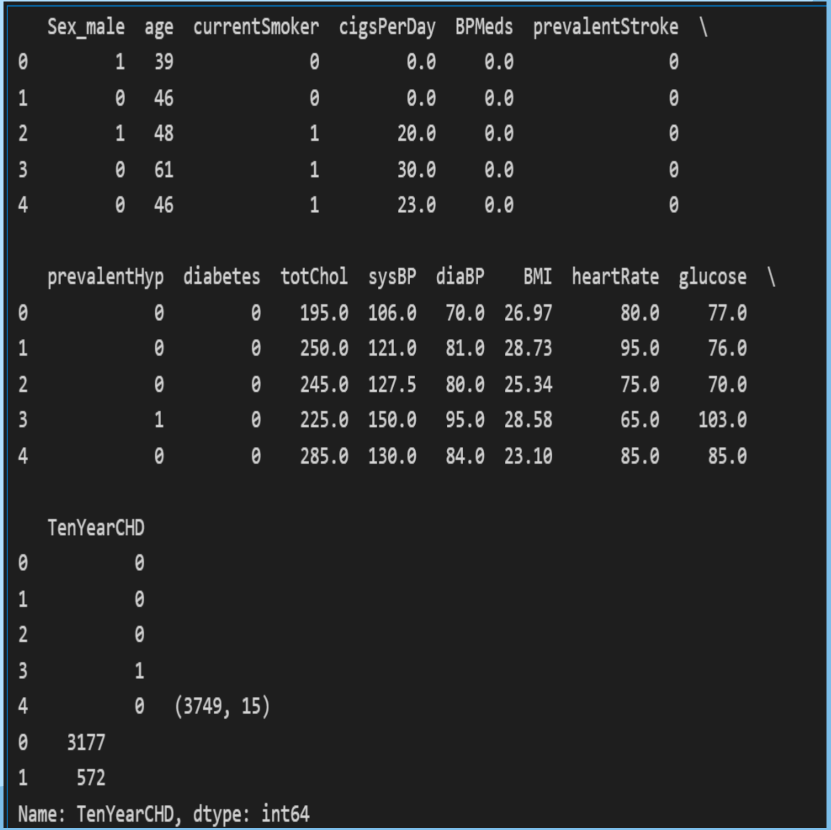
CODE



```python
import numpy as np
import scipy.optimize as opt
import statsmodels.api as sm
from sklearn import preprocessing
'exec(% matplotlib inline)'
import matplotlib.pyplot as plt
import matplotlib.mlab as mlab
import seaborn as sn
import pandas as pd
```

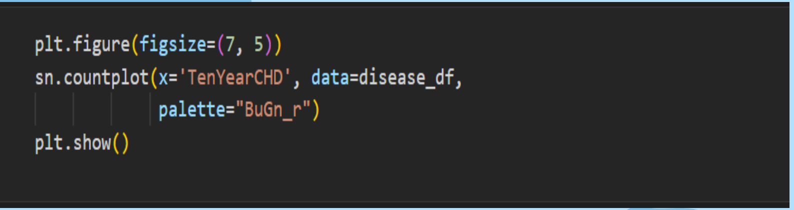
```python
disease_df = pd.read_csv("heart disease prediction.csv")
disease_df.drop(['education'], inplace = True, axis = 1)
disease_df.rename(columns ={'male':'Sex_male'}, inplace = True)
disease_df.dropna(axis = 0, inplace = True)
print(disease_df.head(), disease_df.shape)
print(disease_df.TenYearCHD.value_counts())
```

OUTPUT



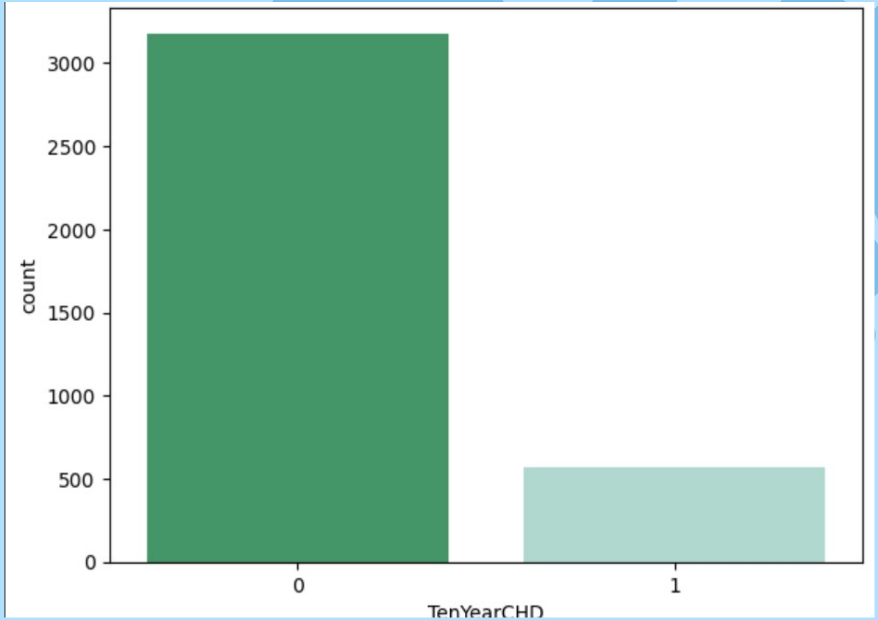
```
   Sex_male  age  currentSmoker  cigsPerDay  BPMeds  prevalentStroke  \
0         1   39              0         0.0     0.0                0
1         0   46              0         0.0     0.0                0
2         1   48              1        20.0     0.0                0
3         0   61              1        30.0     0.0                0
4         0   46              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  (3749, 15)
0    3177
1     572
Name: TenYearCHD, dtype: int64
```

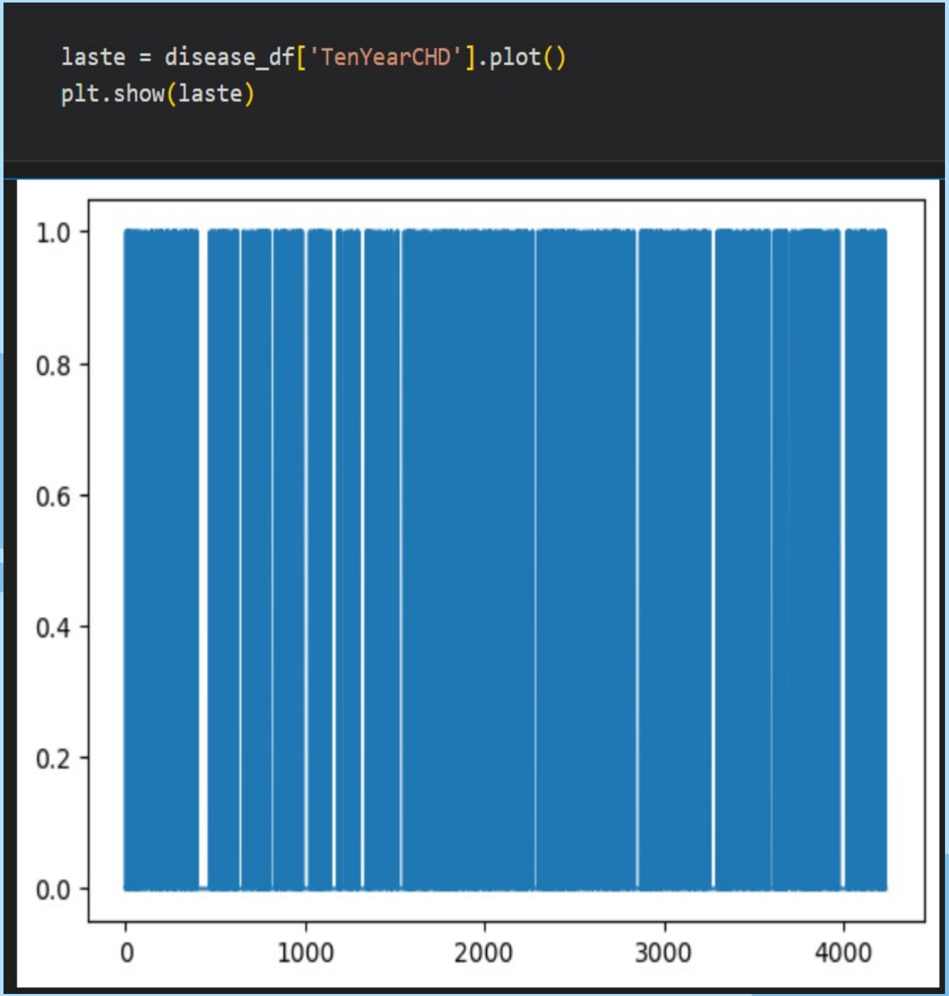
CODE



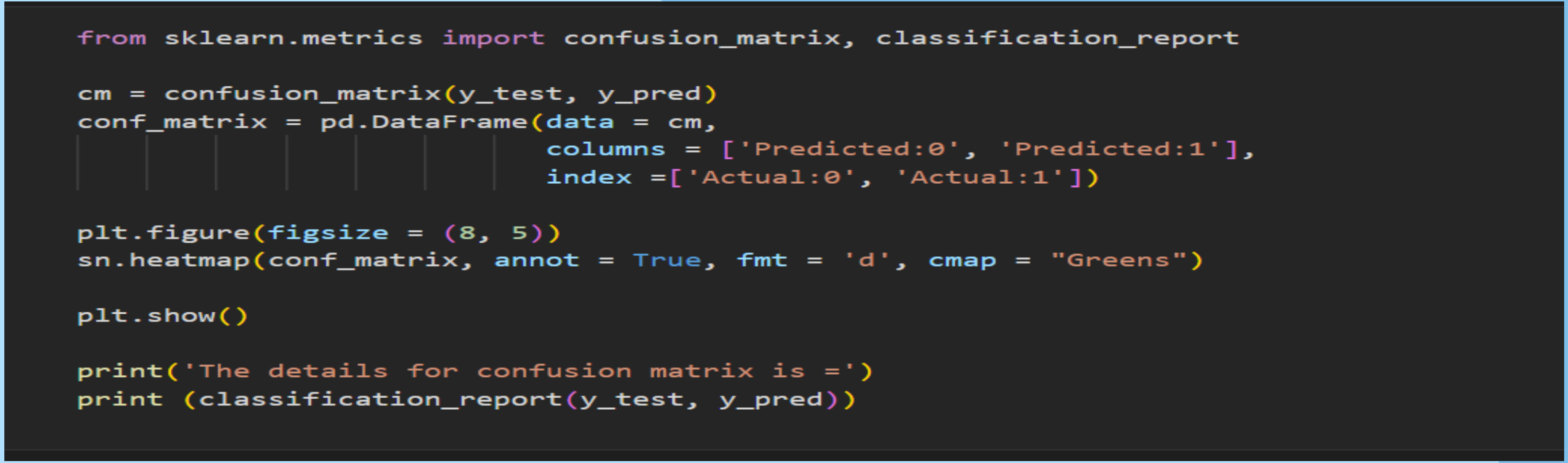
```
plt.figure(figsize=(7, 5))
sn.countplot(x='TenYearCHD', data=disease_df,
             palette="BuGn_r")
plt.show()
```

OUTPUT

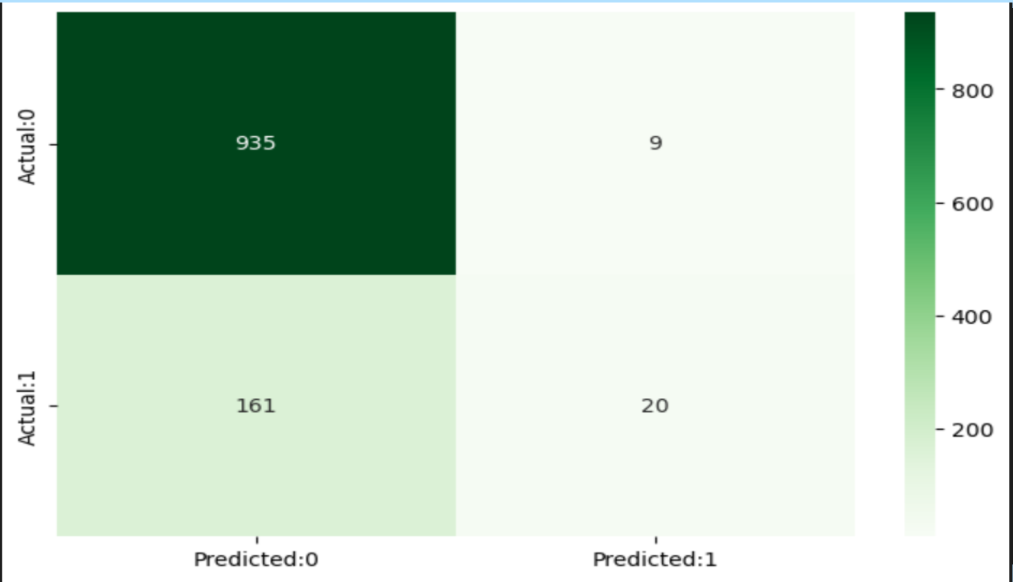


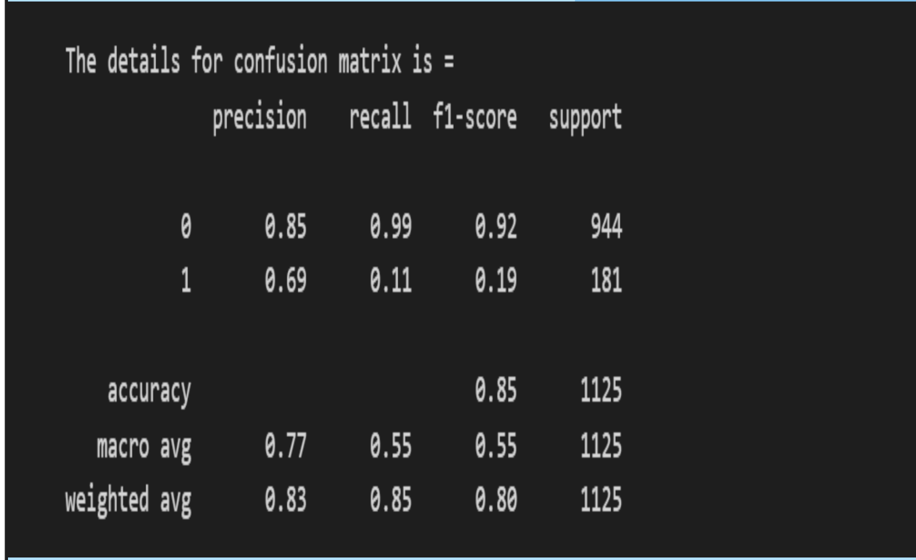


```
laste = disease_df['TenYearCHD'].plot()
plt.show(laste)
```

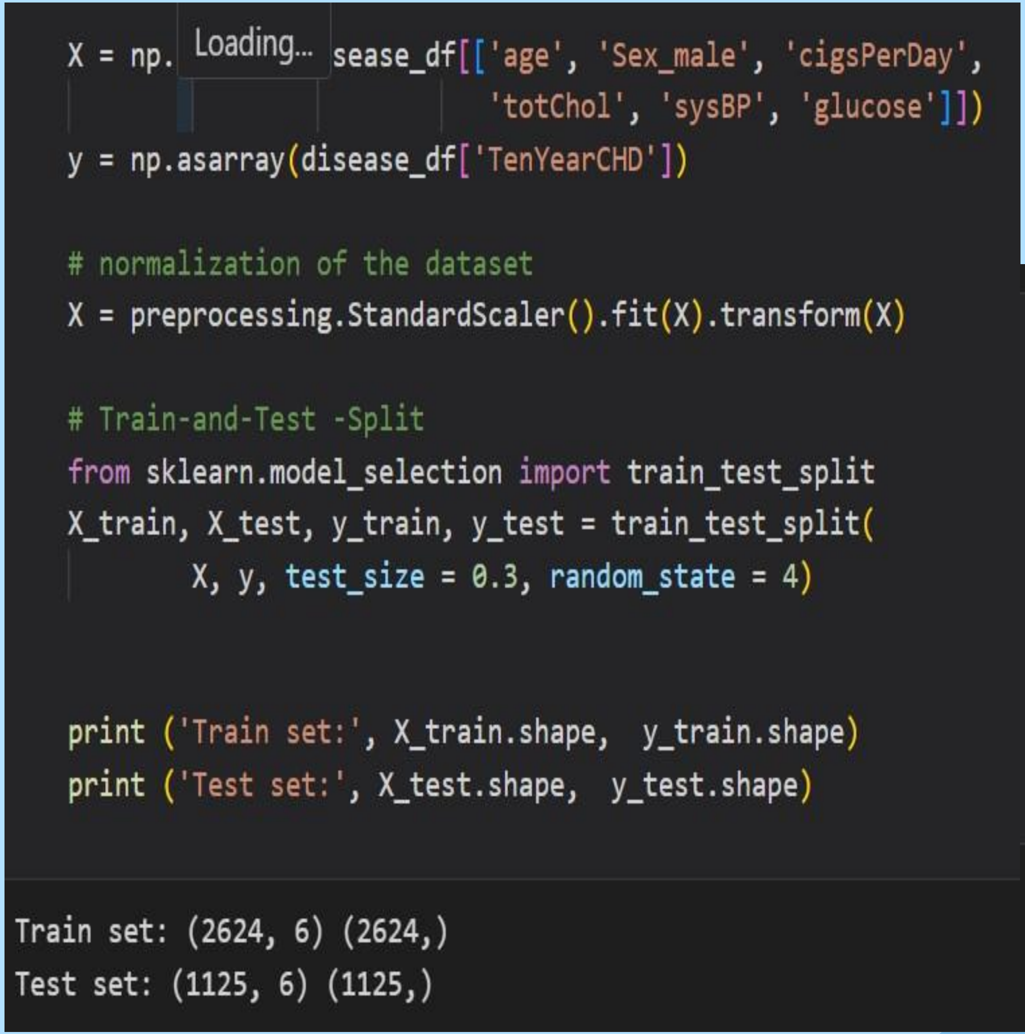


```
from sklearn.metrics import confusion_matrix, classification_report

cm = confusion_matrix(y_test, y_pred)
conf_matrix = pd.DataFrame(data = cm,
                           columns = ['Predicted:0', 'Predicted:1'],
                           index =['Actual:0', 'Actual:1'])

plt.figure(figsize = (8, 5))
sn.heatmap(conf_matrix, annot = True, fmt = 'd', cmap = "Greens")

plt.show()

print('The details for confusion matrix is =')
print (classification_report(y_test, y_pred))
```





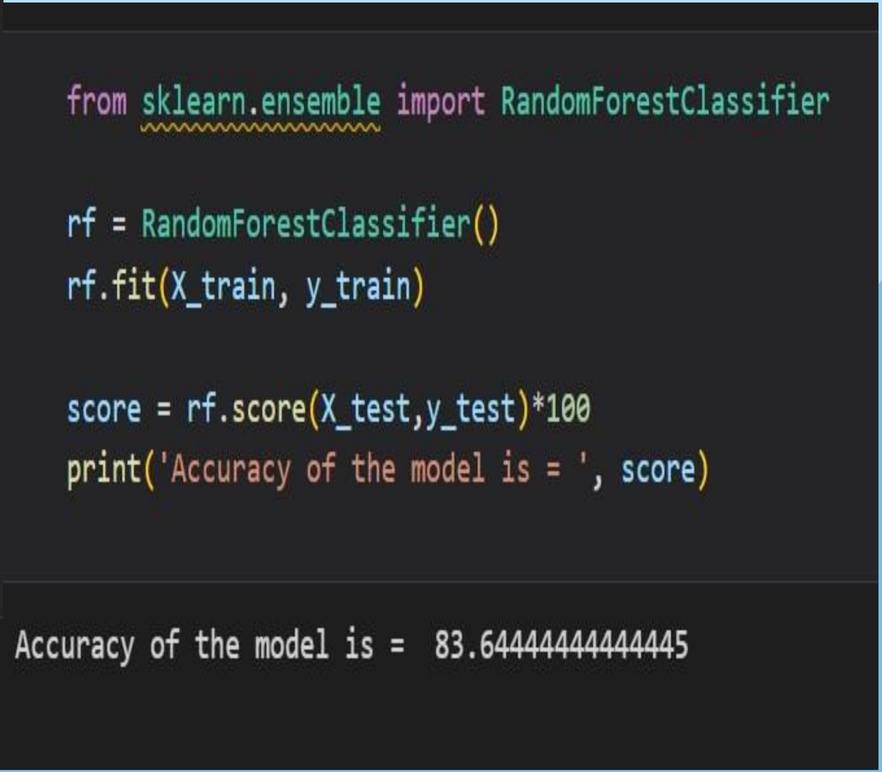
```
The details for confusion matrix is =
              precision    recall  f1-score   support

           0       0.85      0.99      0.92       944
           1       0.69      0.11      0.19       181

    accuracy                           0.85      1125
   macro avg       0.77      0.55      0.55      1125
weighted avg       0.83      0.85      0.80      1125
```



```
X = np. Loading... sease_df[['age', 'Sex_male', 'cigsPerDay',
                          'totChol', 'sysBP', 'glucose']])
y = np.asarray(disease_df['TenYearCHD'])

# normalization of the dataset
X = preprocessing.StandardScaler().fit(X).transform(X)

# Train-and-Test -Split
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
        X, y, test_size = 0.3, random_state = 4)


print ('Train set:', X_train.shape,  y_train.shape)
print ('Test set:', X_test.shape,  y_test.shape)
```

```
Train set: (2624, 6) (2624,)
Test set: (1125, 6) (1125,)
```

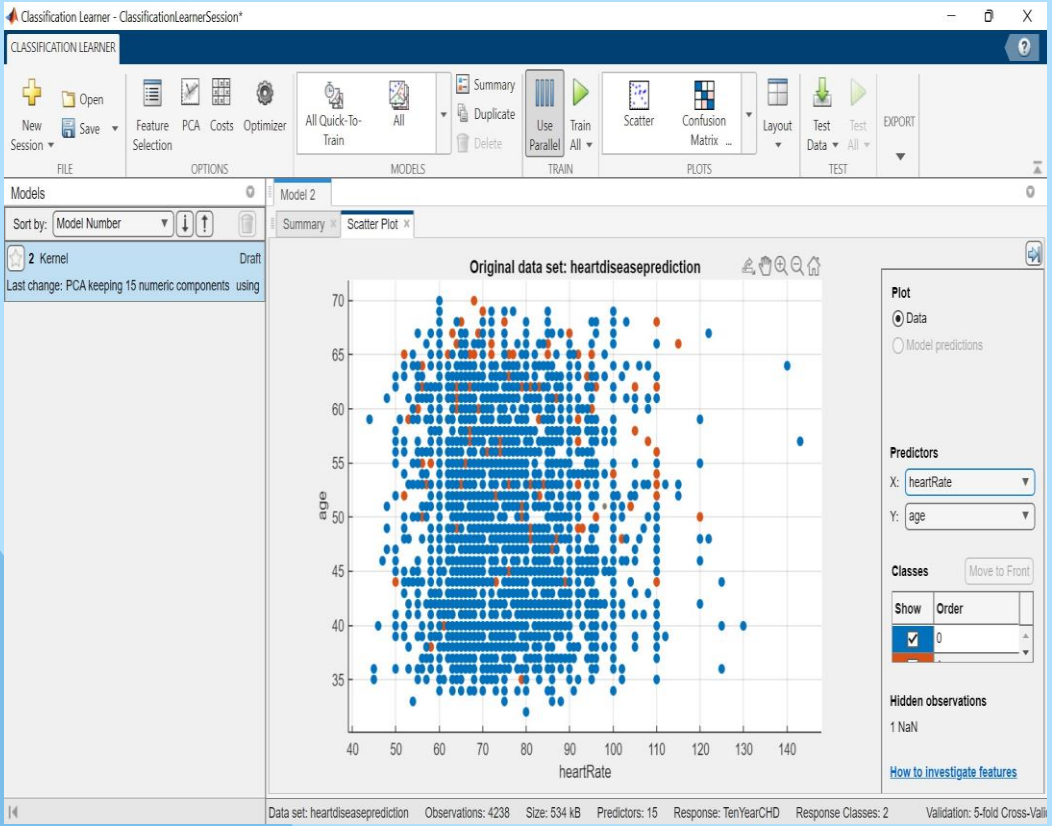
Accuracy



```
from sklearn.ensemble import RandomForestClassifier

rf = RandomForestClassifier()
rf.fit(X_train, y_train)

score = rf.score(X_test,y_test)*100
print('Accuracy of the model is = ', score)
```

```
Accuracy of the model is =  83.64444444444445
```

ACCURACY :


Classification Learner - ClassificationLearnerSession*

Models — Sort by: Model Number

2 Kernel — Draft

Last change: PCA keeping 15 numeric components using

Model 2 — Summary | Scatter Plot

Original data set: heartdiseaseprediction

Plot: (● Data) Model predictions

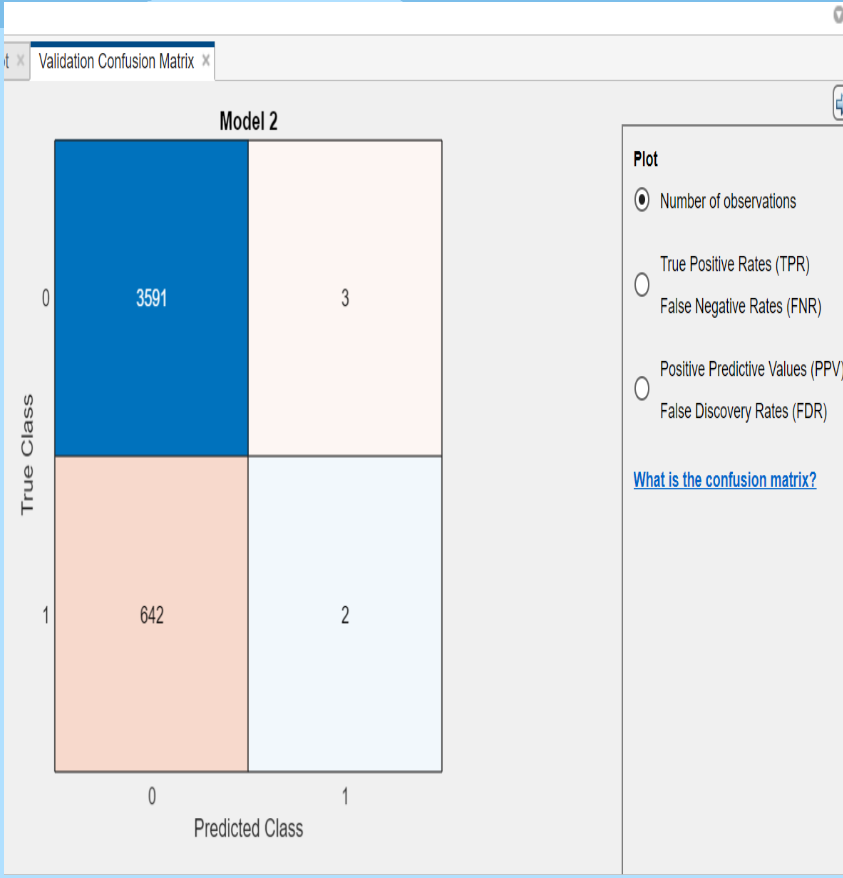
Predictors — X: heartRate; Y: age

Classes — Show / Order: ✓ 0

Hidden observations: 1 NaN

How to investigate features

Data set: heartdiseaseprediction   Observations: 4238   Size: 534 kB   Predictors: 15   Response: TenYearCHD   Response Classes: 2   Validation: 5-fold Cross-Vali


Validation Confusion Matrix

Model 2

True Class \ Predicted Class	0	1
0	3591	3
1	642	2

Plot

(● Number of observations)

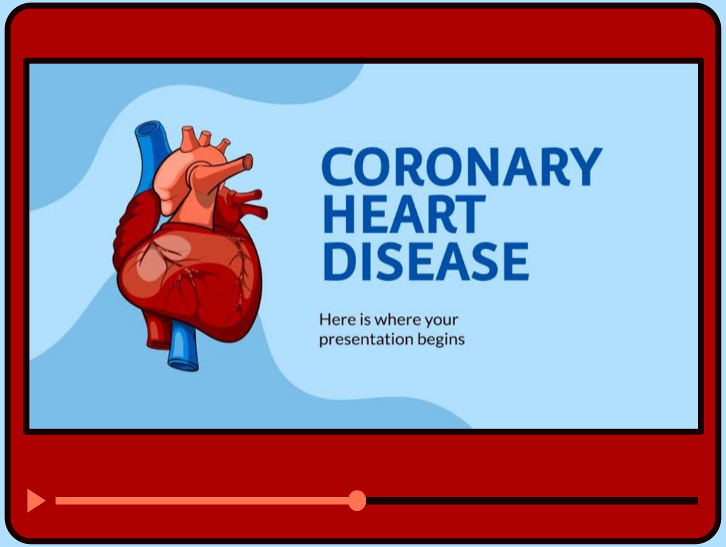
True Positive Rates (TPR) / False Negative Rates (FNR)

Positive Predictive Values (PPV) / False Discovery Rates (FDR)

What is the confusion matrix?

CONCLUSION

From the above model accuracy, **LOGISTIC REGRESSION** is giving us the accuracy which is 84.4%.



THANK YOU