HEART DISEASE PREDICTION PROJECT

USING
MACHINE LEARNING

DONE BY

DHIVYA .V
DHARUNYA .K
KEERTHANA .S

INDEX

Table of content

INTRODUCTION :

Heart disease describes a range of conditions that affect your heart. Diseases under the heart disease umbrella include blood vessel diseases, such as coronary artery disease, heart rhythm problems (arrhythmias) and heart defects you're born with (congenital heart defects), among others.

The term "heart disease" is often used interchangeably with the term "cardiovascular disease". Cardiovascular disease generally refers to conditions that involve narrowed or blocked blood vessels that can lead to a heart attack, chest pain (angina) or stroke. Other heart conditions, such as those that affect your heart's muscle, valves or rhythm, also are considered forms of heart disease.

Heart disease is one of the biggest causes of morbidity and mortality among the population of the world. Prediction of cardiovascular disease is regarded as one of the most important subjects in the section of clinical data analysis. The amount of data in the healthcare industry is huge. Data mining turns the large collection of raw healthcare data into information that can help to make informed decisions and predictions.

Heart disease is the leading cause of death for both men and women.This makes heart disease a major concern to be dealt with. But it is difficult to identify heart disease because of several contributory risk factors such as diabetes, high blood pressure, high cholesterol, abnormal pulse rate, and many other factors. Due to such constraints, scientists have turned towards modern approaches like Data Mining and Machine Learning for predicting the disease.

Machine learning (ML) proves to be effective in assisting in making decisions and predictions from the large quantity of data produced by the healthcare industry.

In this project, We will be applying Machine Learning approaches for classifying whether a person is suffering from heart disease or not, using one of the most used dataset — heart.csv dataset from the UCI Machine Learning Repository Kaggle.

- **0 represents no heart disease present**
- **1 represents heart disease present**

DATASET :

The dataset used in this project is the heart.csv dataset taken from the UCI repository.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	**age**	**sex**	**cp**	**trestbps**	**chol**	**fbs**	**restecg**	**thalach**	**exang**	**oldpeak**	**slope**	**ca**	**thal**	**target**
2	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
3	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
4	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
5	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
6	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
7	57	1	0	140	192	0	1	148	0	0.4	1	0	1	1
8	56	0	1	140	294	0	0	153	0	1.3	1	0	2	1

The dataset consists of 303 individuals data. There are 14 columns(13 features and 1 target variable) in the dataset, which are described below .

- Age: age in years
- Gender: gender (1 = male; 0 = female)
- Cp: chest pain type
  - Value 1: typical angina
  - Value 2: atypical angina
  - Value 3: non-anginas pain
  - Value 4: asymptomatic
- Trestbps: resting blood pressure (in mm Hg on admission to the hospital)
- Chol: serum cholesterol in mg/dl
- Fbs: fasting blood sugar > 120 mg/dl (1 = true; 0 = false)
- Restecg: resting electrocardiographic results
  - Value 0: normal
  - Value 1: having ST-T wave abnormality
  - Value 2: showing probable or definite left ventricular hypertrophy
- thalach: maximum heart rate achieved in beats per minute (bpm)
- Exang: exercise induced angina (1 = yes; 0 = no)
- Oldpeak: ST depression induced by exercise relative to rest

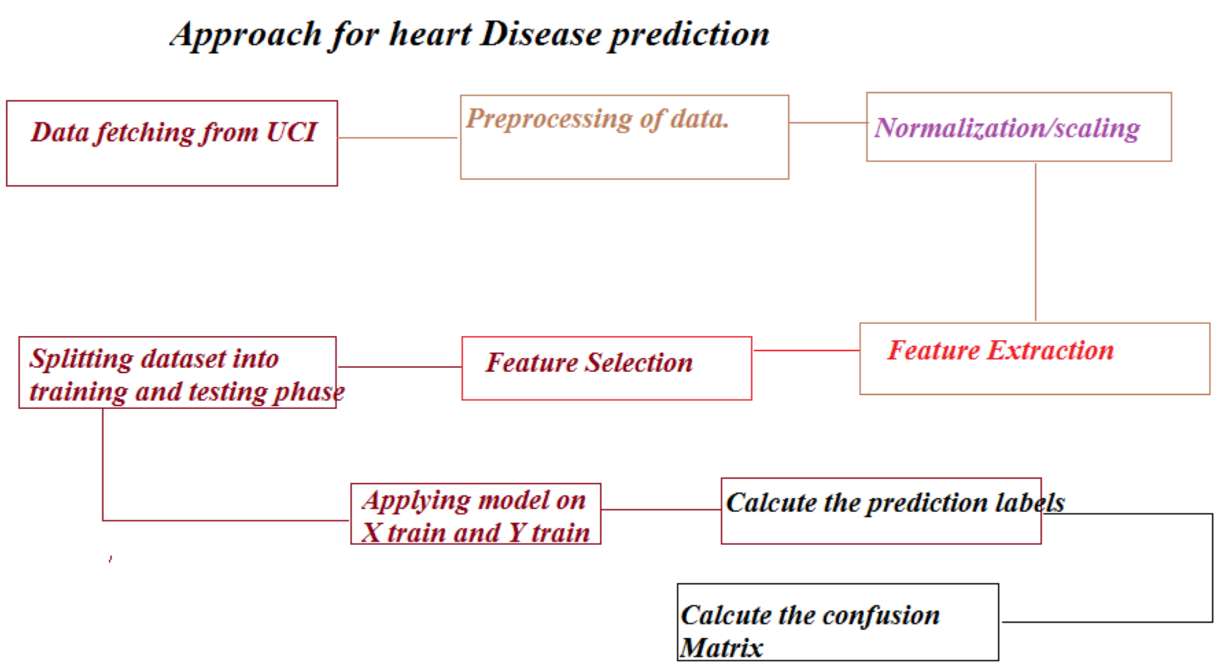
- Slope: the slope of the peak exercise ST segment
    - Value 1: up-sloping
    - Value 2: flat
    - Value 3: down-sloping
- Ca: number of major vessels (0-3) coloured by fluoroscopy
- Thal: 3 = normal; 6 = fixed defect; 7 = reversible defect

OBJECTIVES :

- High Accuracy
- Overcomes risk factors like diabetes, high blood pressure etc
- More efficiency
- Computes very large quantity of data

METHODS :

STEPS INVOLVED



IMPORT DATASET :

The code is implemented in Python.

After downloading the dataset from Kaggle, I saved it to my working directory with the name heart.csv Next, I used read_csv() to read the dataset and save it to the dataset variable.

Data Preprocessing :

HANDLING NULL VALUES :

Let us check the null values. I used the info() method to look at the data for null values .

```
In [5]: dataset.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
age         303 non-null int64
sex         303 non-null int64
cp          303 non-null int64
trestbps    303 non-null int64
chol        303 non-null int64
fbs         303 non-null int64
restecg     303 non-null int64
thalach     303 non-null int64
exang       303 non-null int64
oldpeak     303 non-null float64
slope       303 non-null int64
ca          303 non-null int64
thal        303 non-null int64
target      303 non-null int64
dtypes: float64(1), int64(13)
memory usage: 33.2 KB
```

As you can see from the output above, there are a total of 13 features and 1 target variable. Also, there are no missing values so we don't need to take care of any null values. Next, I used describe() method.

FEATURE SCALING :

describe() method is used to get the statistical details like count ,mean ,std etc.

```
dataset.describe()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpe
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.0(
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.528053	149.646865	0.326733	1.039(
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.525860	22.905161	0.469794	1.161(
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.000(
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.500000	0.000000	0.000(
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	153.000000	0.000000	0.800(
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000	166.000000	1.000000	1.600(
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.200(

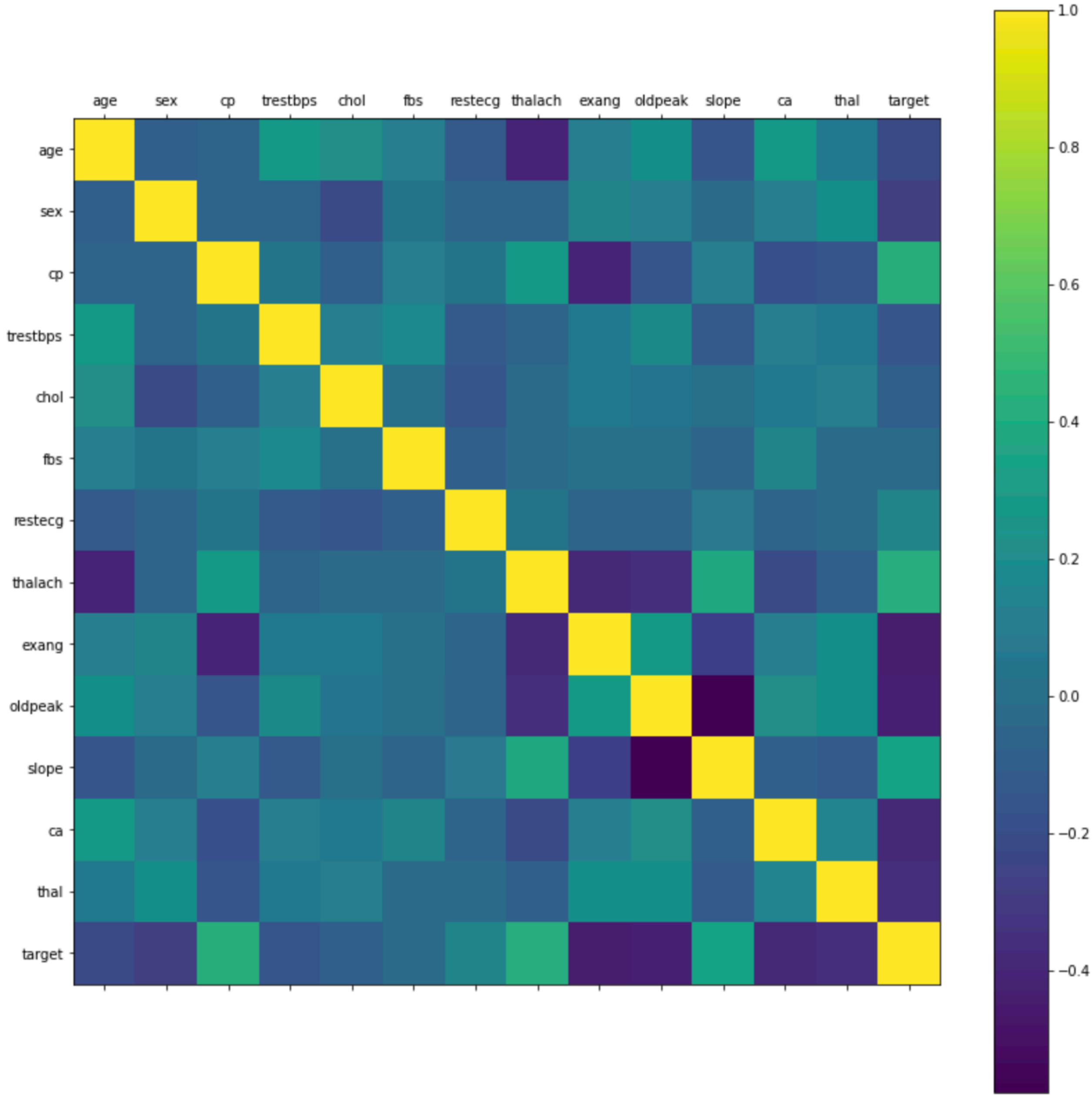
The method revealed that the range of each variable is different. The maximum value of age is 77 but for chol it is 564. Thus, feature scaling must be performed on the dataset.

UNDERSTANDING THE DATA :

CORRELATION MATRIX :

correlation matrix helps us to understand the dependencies of attributes. Ranges from 1(strongly related) to -1 (badly related). Positive correlation indicates If one variable increases other increases and vice versa .Negative correlation indicates if one variable increases other variable decreases and vice versa.

Let's see the correlation matrix of features and try to analyse it. The figure size is defined to 12 x 8 by using rcParams. Then, I used pyplot to show the correlation matrix. Using xticks and yticks, I've added names to the correlation matrix. colorbox() shows the colorbar for the matrix.

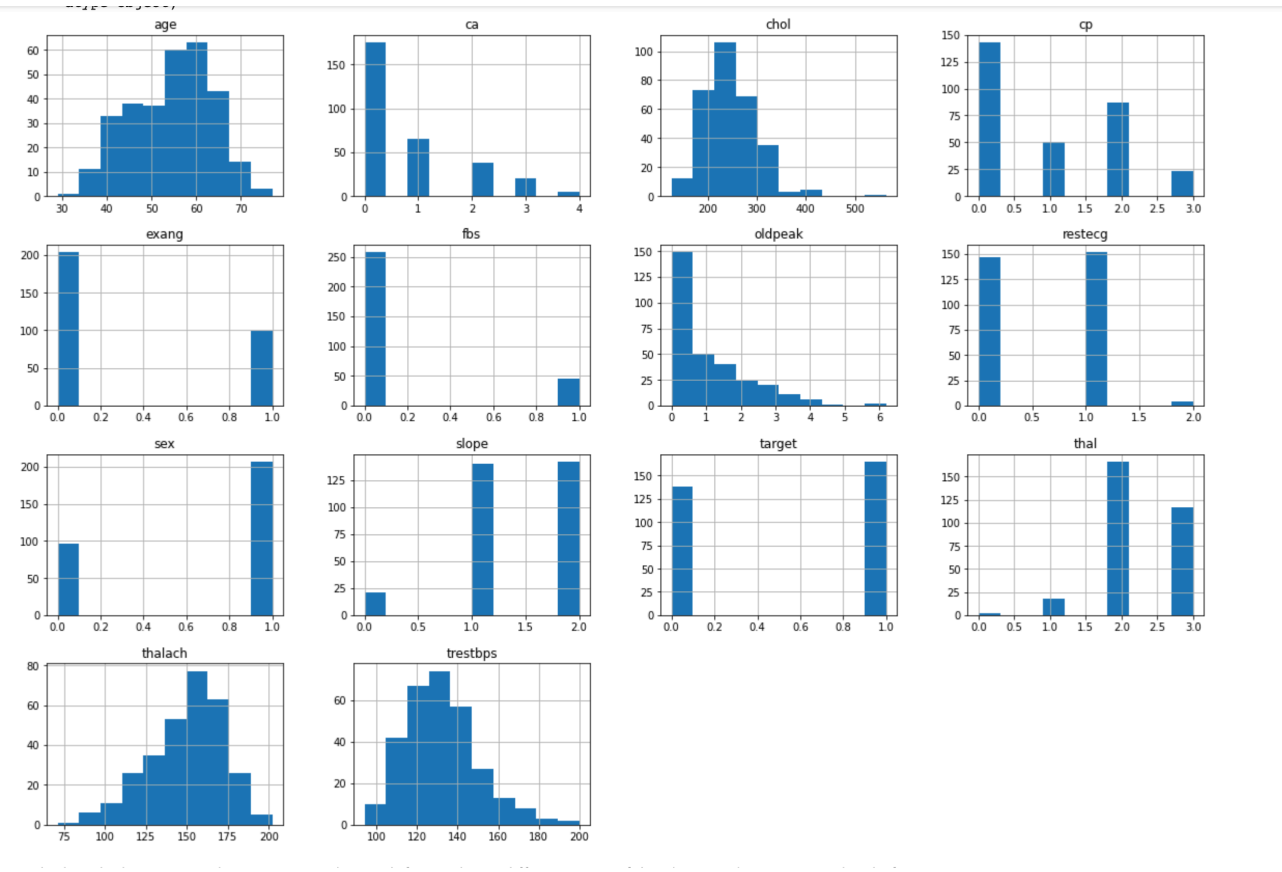


CORRELATION MATRIX

It's easy to see that there is no single feature that has a very high correlation with our target value. Also, some of the features have a negative correlation with the target value and some have positive.Next, we'll take a look at the histograms for each variable.

Histogram :

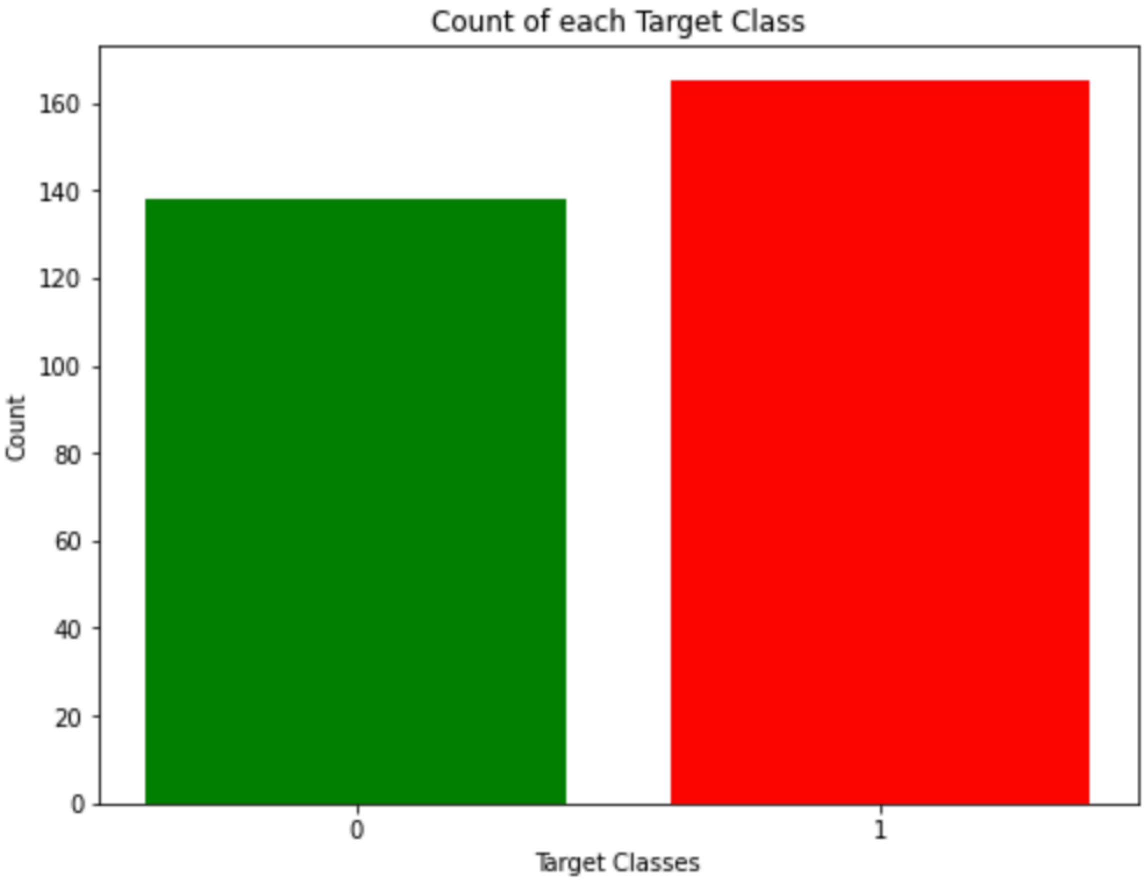
dataset.hist() :



Let's take a look at the plots. It shows how each feature and label is distributed along different ranges, which further confirms the need for scaling. Next, wherever you see discrete bars, it basically means that each of these is actually a categorical variable. We will need to handle these categorical variables before applying Machine Learning. Our target labels have two classes, 0 for no disease and 1 for disease.

Bar Plot for Target Class :

It's really essential that the dataset we are working on should be approximately balanced. An extremely imbalanced dataset can render the whole model training useless and thus, will be of no use.



From the plot, we can see that the classes are almost balanced and we are good to proceed with data processing.

To work with categorical variables, we should break each categorical column into dummy columns with 1s and 0s and then the dataset will be ready to train.

MACHINE LEARNING ALGORITHMS :

In this project, I took 6 algorithms and varied their various parameters and compared the final models.

Now let us divide the data in the test and train set.In this project, I have divided the data into an 80: 20 ratio. That is, the training size is 80% and testing size is 20% of the whole data.

RESULTS :

Accuracy in each Algorithms :

CLASSIFICATION ALGORITHMS	TRAIN DATA	TEST DATA
SVM	89.25	86.88
Naive bayes	83.47	85.24
Logistic regression	84.71	85.24
Decision tree	100	78.68
Random forest	98.76	81.96
XGboost	99.17	85.24

We see that the highest accuracy for the test set is achieved by SVM which is equal to 86.88%.

The highest accuracy for the training set is 100% achieved by Decision Tree.

CONFUSION MATRIX :

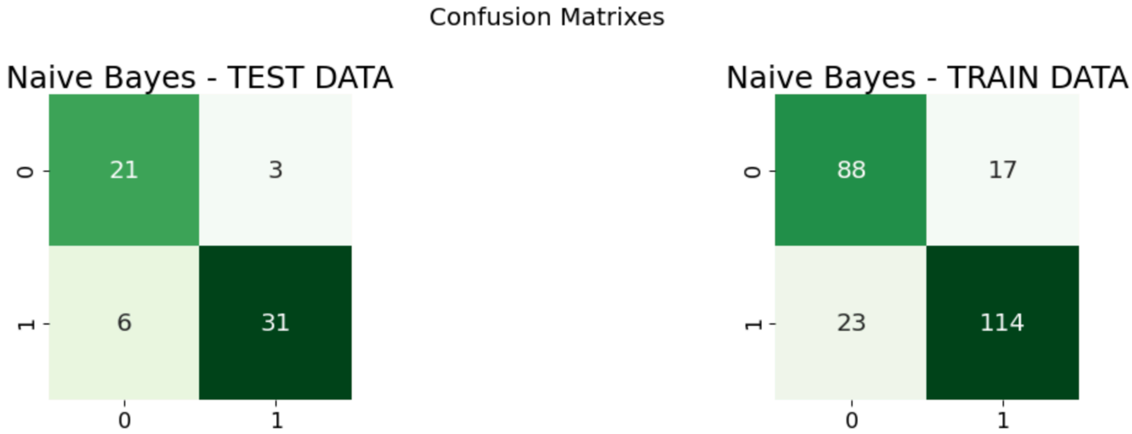
The confusion matrix displays the correctly predicted as well as incorrectly predicted values by a classifier.

The sum of TP and TN, from the confusion matrix, is the number of correctly classified entries by the classifier.

EVALUATION :

<u>NAIVE BAYES</u> :

```
Accuracy for training set for Naive Bayes = 0.8347107438016529
Accuracy for test set for Naive Bayes = 0.8524590163934426
```



ACCURACY = (TP + TN)/(TP+TN+FP+FN)

FOR TEST DATA = (21+31)/(21+31+6+3) = 52/61

=0.8524

FOR TRAIN DATA = (88+114)/(88+114+23+17) = 202/242

= 0.83471

SVM :

```
Accuracy for training set for svm = 0.8925619834710744
Accuracy for test set for svm = 0.8688524590163934
```

Confusion Matrixes

ACCURACY = (TP + TN)/(TP+TN+FP+FN)

FOR TEST DATA = (21+32)/(21+32+6+2) = 53/61

=0.86885

FOR TRAIN DATA = (91+125)/(91+125+20+6) = 216/242

= 0.89256

LOGISTIC REGRESSION :

```
Accuracy for training set for Logistic Regression = 0.8471074380165289
Accuracy for test set for Logistic Regression = 0.8524590163934426
```

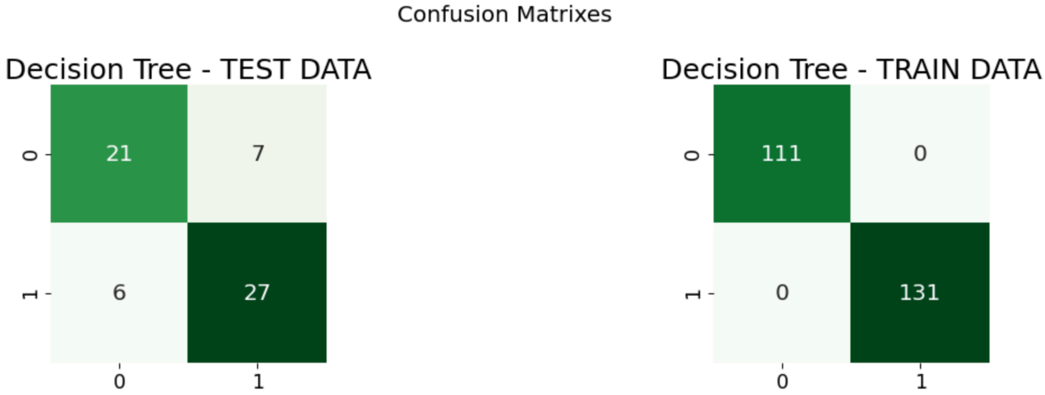
Confusion Matrixes

ACCURACY : TEAST DATA = 52/61 TRAIN DATA = 205/242

= 0.8524 = 0.8471

DECISION TREE :

```
Accuracy for training set for Decision Tree = 1.0
Accuracy for test set for Decision Tree = 0.7868852459016393
```

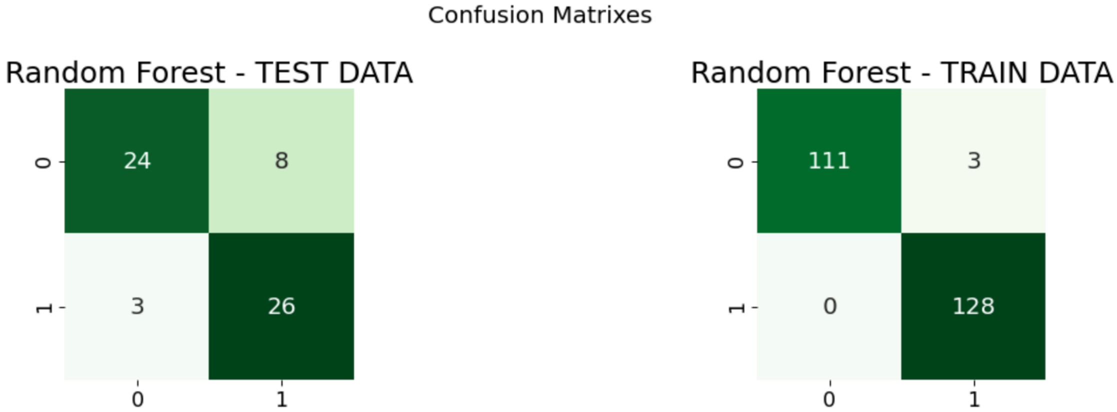


ACCURACY : TEST DATA = 48/61 = 0.7868

TRAIN DATA =242/242 = 1

RANDOM FOREST:

```
Accuracy for training set for Random Forest = 0.987603305785124
Accuracy for test set for Random Forest = 0.819672131147541
```

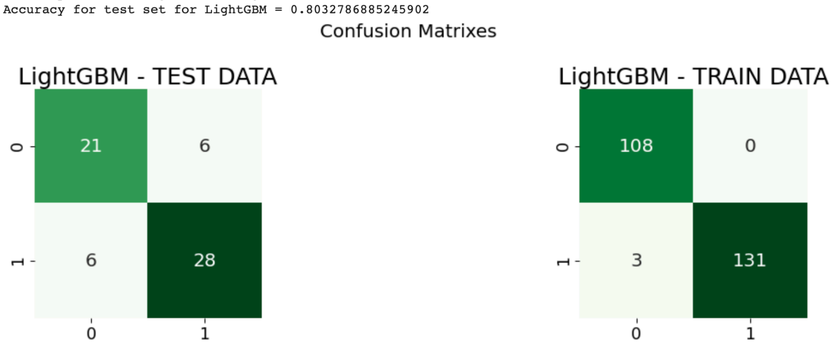


ACCURACY : TEST DATA = 50/61 =0.81967

TRAIN DATA = 239/242 = 0.98760

XGBOOST:

```
Accuracy for training set for LightGBM = 0.987603305785124
Accuracy for test set for LightGBM = 0.8032786885245902
```



TEST DATA = 49/61 = 0.80327

TRAIN DATA= 239/242 = 0.9876

DISCUSSION AND CONCLUSION :

After observing each model SVM gave the best accuracy of 86.88 with test data and Decision tree gave the best accuracy of 100 with Tain data.

Heart Disease is one of the major concerns for society today.

it is difficult to manually determine the odds of getting heart disease based on risk factors. However, machine learning techniques are useful to predict the output from existing data.

REFERENCE :

towardsdatascience.com

docs.python.org

colab.research.google.com