

# HEART DISEASES PREDICTION MODEL

## Abstract

Heart related diseases or Cardiovascular Diseases (CVDs) are the main reason for a huge number of deaths in the world over the last few decades and has emerged as the most life-threatening disease, not only in India but in the whole world. So, there is a need of reliable, accurate and feasible system to diagnose such diseases in time for proper treatment. Machine Learning algorithms and techniques have been applied to various medical datasets to automate the analysis of large and complex data. The obtained results have illustrated that the designed diagnostic system can effectively predict the risk level of heart diseases.

## Submitted to:

Mr. Mayur Dev Sewak  
General Manager, Operations  
EI systems Services

Ms. Mallika Srivastava Trainer,  
Programming & Algorithms,  
EI systems Services

Ankit Yadav  
ankiii0071@gmail.com

Serial no.	Title	Page no.
1	Table of contents	1
2	List of figures/flowchart	2
3	Introduction	3
4	Motivation	3
5	Objectives	3
6	Existing System	4
7	Scope	4
8	Limitations	4
9	Methodologies	5
11	Proposed Model	5
12	Data Collection	6
13	Exploratory Data Analysis	7
14	Correlation matrices	14
15	Data Pre-Processing	15
16	Model Selection	17
16.1	Logistic Regression	17
16.2	Applying Logistic Regression	18
17	Splitting of data: Training & Testing	19
18	Evaluation matrices	20
18.1	Confusion matrix	20
18.2	Accuracy	20
19	Evaluating and visualizing model performance	21
20	Testing Technologies & python libraries	21
21	Conclusion	22
22	Project Summary	22
23	References	23

Serial no.	Figure name	Page no.
1	Flow chart of proposed model of project	5
2	plot of how much people suffering with heart diseases and are not.	9
3	plot of different attributes with different ranges of its value to have diseases or not	10
4	plot of exercise-induced ST depression vs. rest looks at heart stress	12
5	Heart disease in function of age and max heart rate	13
6	Correlation Matrix Visualization	14
7	correlation with target value.	15
8	Sigmoid Function	17
9	flow chart of working of logistic regression	18

## INTRODUCTION

Heart is an important organ of the human body. It pumps blood to every part of our anatomy. If it fails to function correctly, then the brain and various other organs will stop working, and within few minutes, the person will die. Change in lifestyle, work related stress and bad food habits contribute to the increase in rate of several heart related diseases. Heart diseases have emerged as one of the most prominent cause of death all around the world.

Cardiovascular diseases (CVDs) are the number 1 cause of death globally, taking an estimated 17.9 million lives each year. CVDs are a group of disorders of the heart and blood vessels and include coronary heart disease, cerebrovascular disease, rheumatic heart disease and other conditions.

Medical organisations, all around the world, collect data on various health related issues. These data can be exploited using various machine learning techniques to gain useful insights. But the data collected is very massive and, many a times, this data can be very noisy. These datasets, which are too overwhelming for human minds to comprehend, can be easily explored using various machine learning techniques. My model based on supervised learning algorithms that is Logistic Regression.

## MOTIVATION

The main objective of this research is to develop a heart prediction system. The system can discover and extract hidden knowledge associated with diseases from a historical heart data set. Heart disease prediction system aims to exploit data mining techniques on medical data set to assist in the prediction of the heart diseases. The objective of this project is to prioritize the diagnostic test and to see some of the health habits that contribute to CVD.

## OBJECTIVES

- ▶ The main objective of developing this project is -
- ▶ To develop machine learning model to predict future possibility of heart disease by implementing Logistic Regression.
- ▶ To determine significant risk factors based on medical dataset which may lead to heart disease.
- ▶ To analyze feature selection methods and understand their working principle.

## Existing System

- ▶ Very few systems use the available clinical data for prediction purposes and even if they do, they are restricted by the large number of association rules that apply.
- ▶ Diagnosis of the condition solely depends upon the Doctor's institution and patient's records.
- ▶ The disadvantages are:

Detection is not possible at the earlier stages.

In the existing system, practical use of various collected data is time consuming.

## Scope

Here the scope of this project is that integration of clinical decision support with computer based patient records could reduce medical errors, enhance patient safety, decrease unwanted practice variation and improve practice outcome. This suggestion is promising as data modelling and analysis tools, e.g., data mining have potential to generate knowledge rich environment which can help significantly improve the quality of clinical decisions.

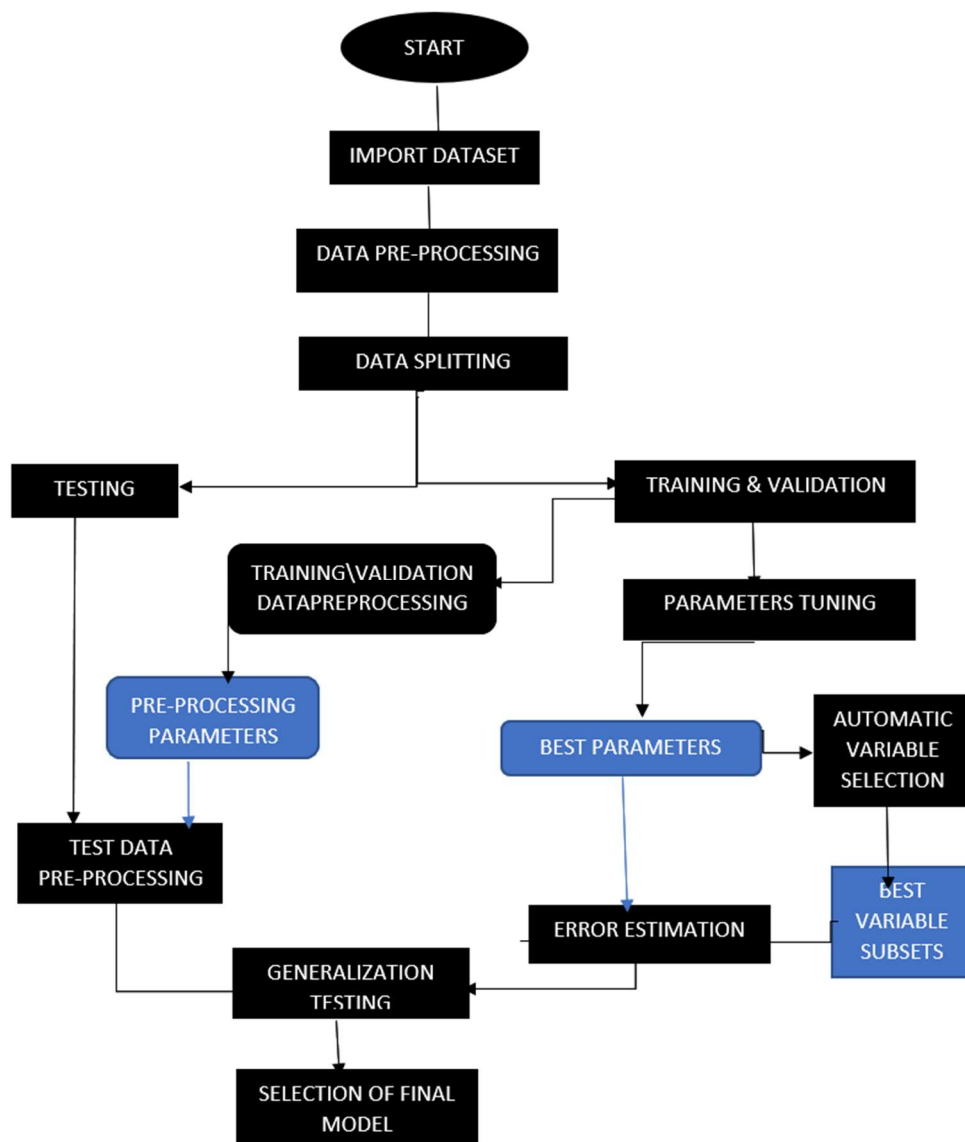
## Limitations:

Medical diagnosis is considered as a significant yet intricate task that needs to be carried out precisely and efficiently., the automation of the same would be highly beneficial. Clinical decisions are often made based doctor's intuitions and experience rather than on the knowledge rich data hidden in the database. This practise leads to the unwanted biases, errors, and excessive medical costs which affects the quality of service provided to patients. Data mining have the enough potential to generate a knowledge rich environment which can help to significantly improve the quality of clinical decisions.

## METHODOLOGIES

The main purpose of this exercise is to forecast cardiac disease in the other data set attributes. This is a problem of machine learning. Jupyter notebook is the app to use. The system proposed was developed to classify people with heart disease and healthy individuals. The efficiency of various predictive models for the diagnosis of cardiac disease have been evaluated on complete and selected apps. The commonly used computer modules generate a detailed report using a powerful predictor algorithm. The main goals of the present framework are to evaluate and test patients with condition results and new patient diseases in order to evaluate the potential for a particular person to develop cardiac disorder. The flow map for the proposed form is seen in the diagram.

Fig no. 1 Flow chart of proposed model of project



## EXPERIMENTS:

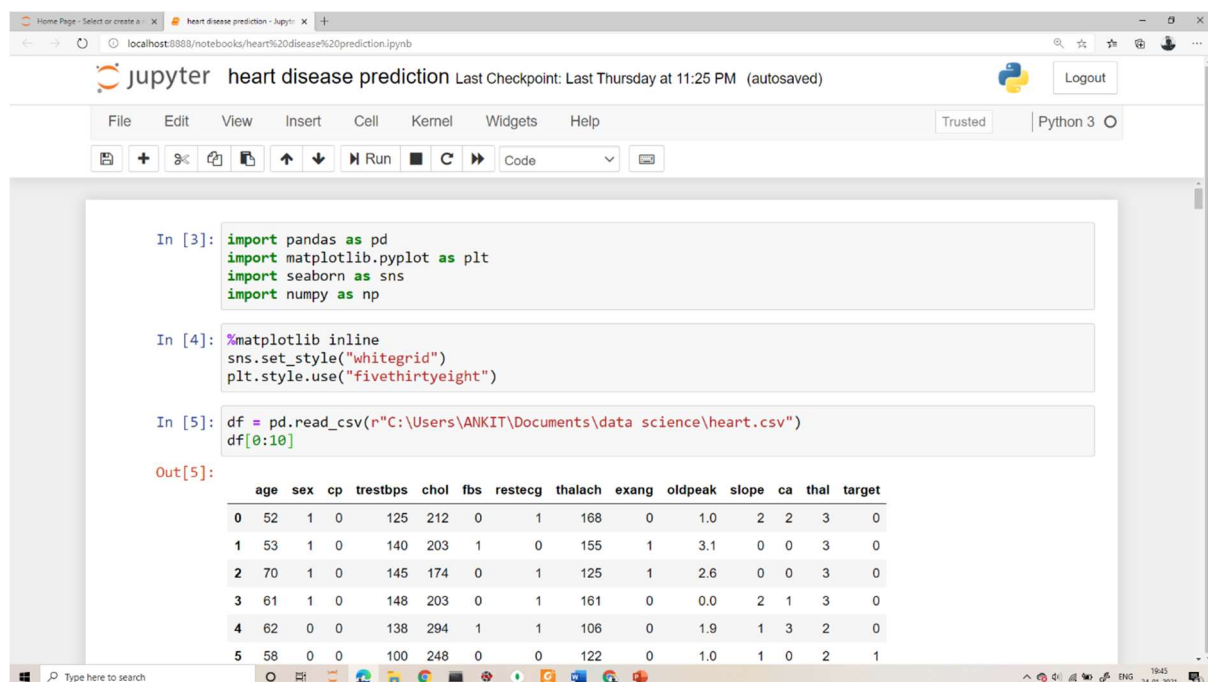
### DATA COLLECTION:

The dataset is publicly available on the Kaggle Website at [4] which is from an ongoing cardiovascular study on residents of the town of Framingham, Massachusetts. It provides patient information which includes over 4000 records and 14 attributes. The attributes include - age, sex, chest pain type, resting blood pressure, serum cholesterol, fasting sugar blood, resting electrocardiographic results, maximum heart rate, exercise induced angina, ST depression induced by exercise, slope of the peak exercise, number of major vessels, and target ranging from 0 to 2, where 0 is absence of heart disease. The data set is in csv (Comma Separated Value) format which is further prepared to data frame as supported by Pandas library in python.

The education data is irrelevant to the heart disease of an individual, so it is dropped. Further with this dataset pre-processing and experiments are then carried out. Now in this section, I will take you through the task of Heart Disease Prediction using machine learning by using the Logistic regression algorithm.

As I am going to use the Python programming language for this task of heart disease prediction.

so let's start by importing some necessary libraries:



The screenshot shows a Jupyter Notebook titled "heart disease prediction" with a last checkpoint from Thursday at 11:25 PM. The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, running, and code execution. The notebook contains three input cells and one output cell. The first cell imports pandas, matplotlib, seaborn, and numpy. The second cell sets the matplotlib style to "whitegrid" and uses "fivethirtyeight". The third cell reads a CSV file from the local path "C:\Users\ANKIT\Documents\data science\heart.csv" and displays the first 10 rows. The output shows a table with 14 columns: age, sex, cp, trestbps, chol, fbs, restecg, thalach, exang, oldpeak, slope, ca, thal, and target.

```
In [3]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

In [4]: %matplotlib inline
sns.set_style("whitegrid")
plt.style.use("fivethirtyeight")

In [5]: df = pd.read_csv(r"C:\Users\ANKIT\Documents\data science\heart.csv")
df[0:10]
```

Out[5]:

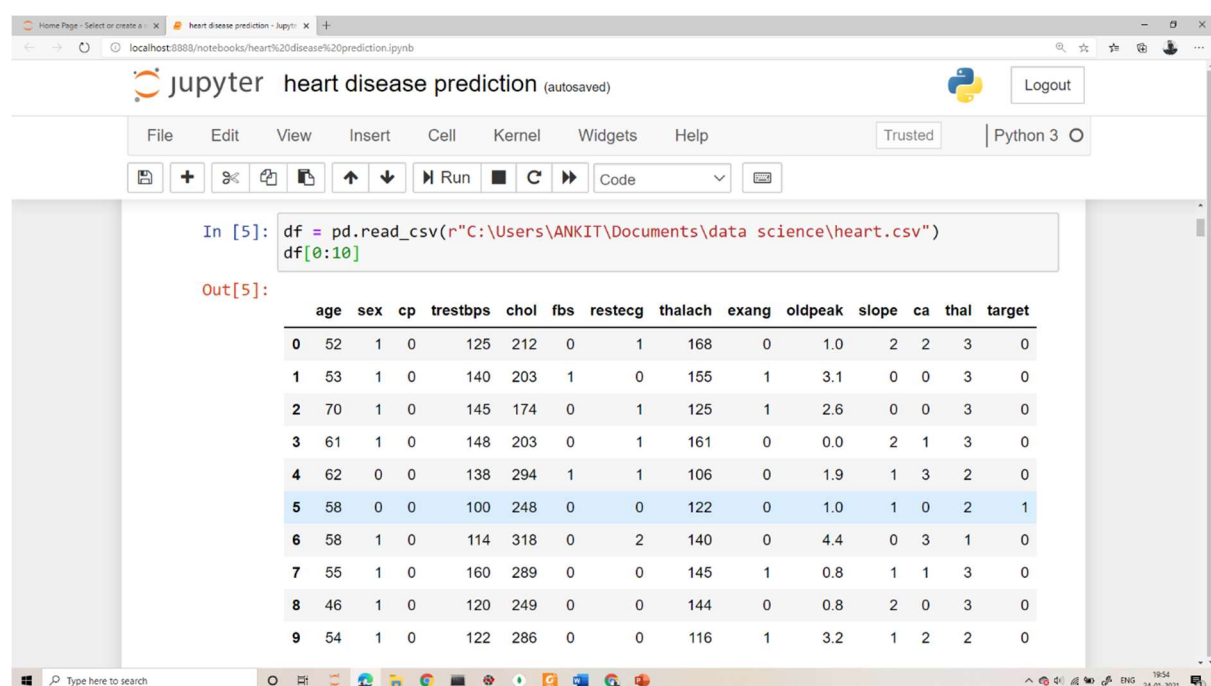
	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0
5	58	0	0	100	248	0	0	122	0	1.0	1	0	2	1

# Exploratory Data Analysis:

Before training the logistic regression we need to observe and analyse the data to see what we are going to work with. The goal here is to learn more about the data and become a topic expert on the dataset you are working with.

EDA helps us find answers to some important questions such as: What question (s) are you trying to solve? What kind of data do we have and how do we handle the different types? What is missing in the data and how do you deal with it? Where are the outliers and why should you care? How can you add, change, or remove features to get the most out of your data?

Now let's start with exploratory data analysis:



The screenshot shows a Jupyter Notebook titled "heart disease prediction (autosaved)". The code cell "In [5]:" contains the following Python code:

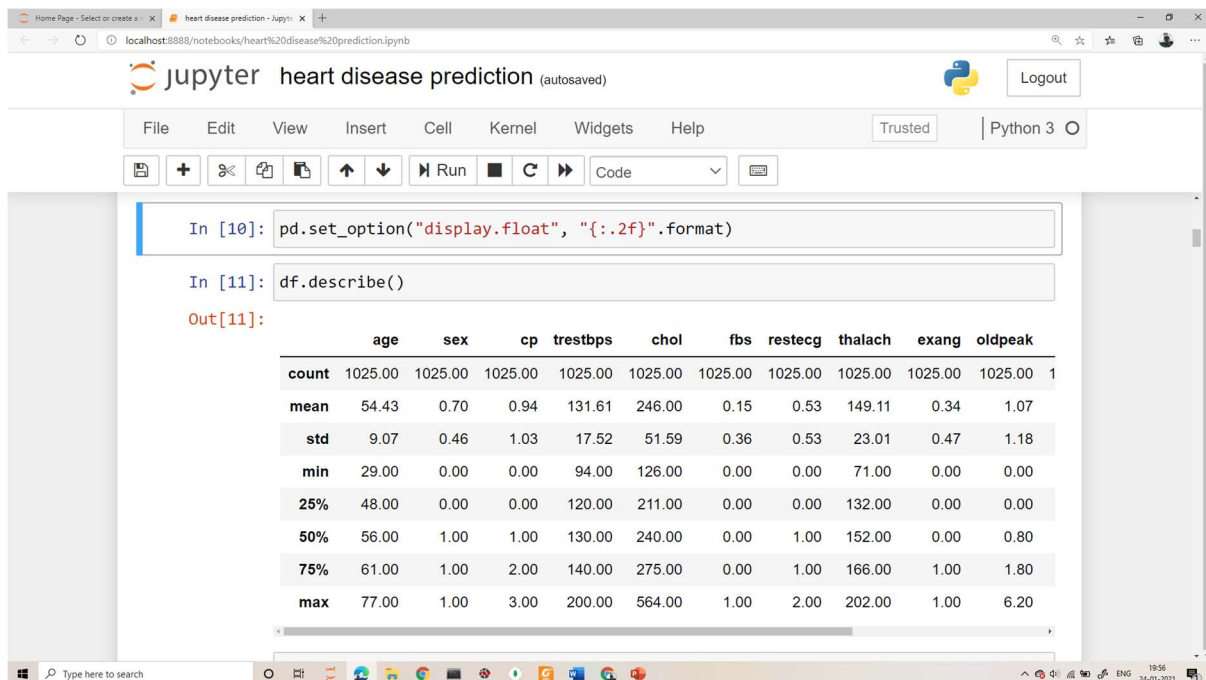
```
df = pd.read_csv(r"C:\Users\ANKIT\Documents\data science\heart.csv")
df[0:10]
```

The output cell "Out[5]:" displays the first 10 rows of the CSV file as a table with 15 columns: age, sex, cp, trestbps, chol, fbs, restecg, thalach, exang, oldpeak, slope, ca, thal, and target. The table is as follows:

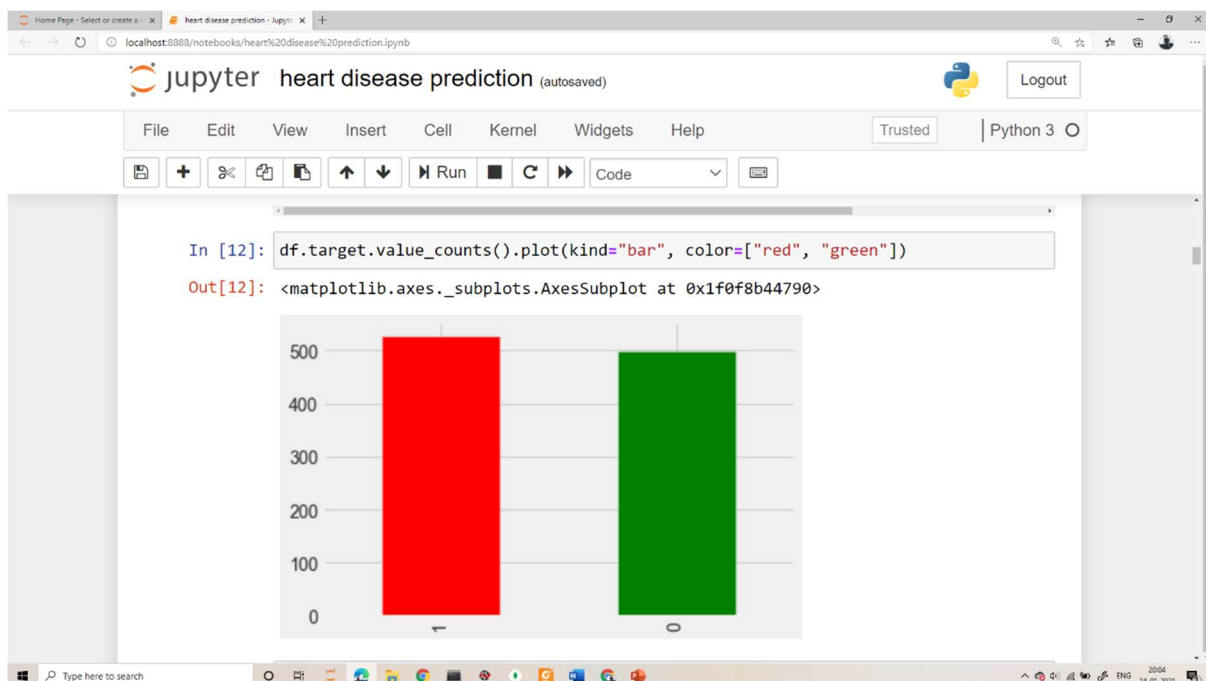
	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0
5	58	0	0	100	248	0	0	122	0	1.0	1	0	2	1
6	58	1	0	114	318	0	2	140	0	4.4	0	3	1	0
7	55	1	0	160	289	0	0	145	1	0.8	1	1	3	0
8	46	1	0	120	249	0	0	144	0	0.8	2	0	3	0
9	54	1	0	122	286	0	0	116	1	3.2	1	2	2	0



We first run a few lines of code to understand what data type each column is and also the number of entries in each of these columns.



We have plotted a basic bar chart using matplotlib to understand how data is split between the different output classes. If we are not satisfied with the representational data, now is the time to get more data to be used for training and testing.



We have 500+ people with heart disease and 490+ people without heart disease, so our problem is balanced.

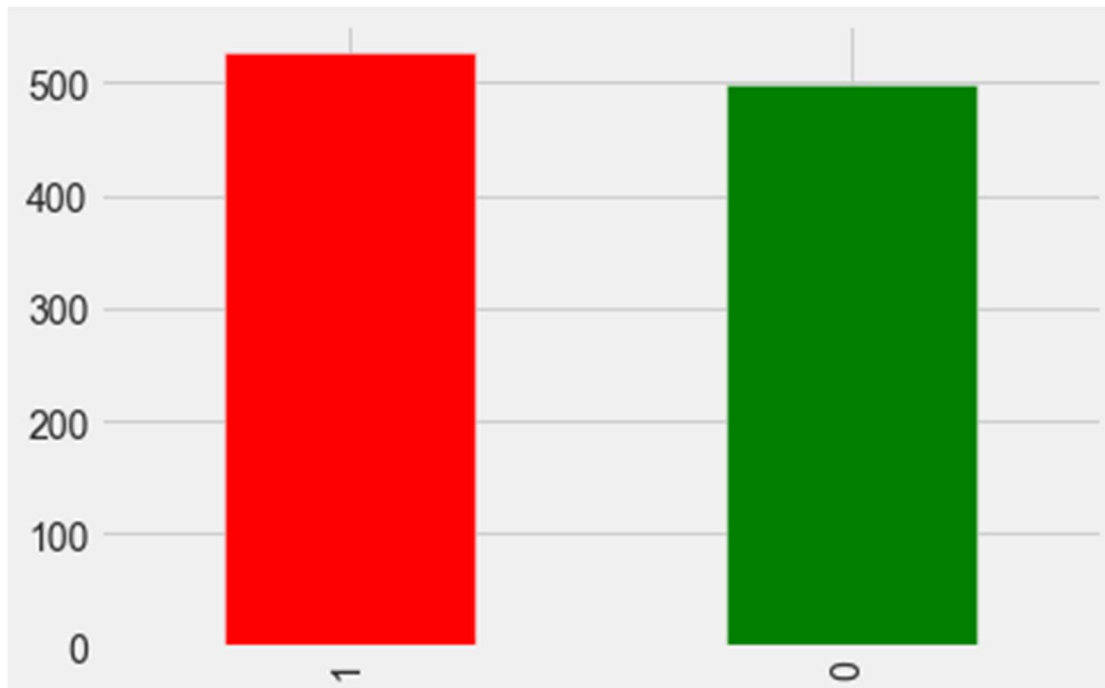
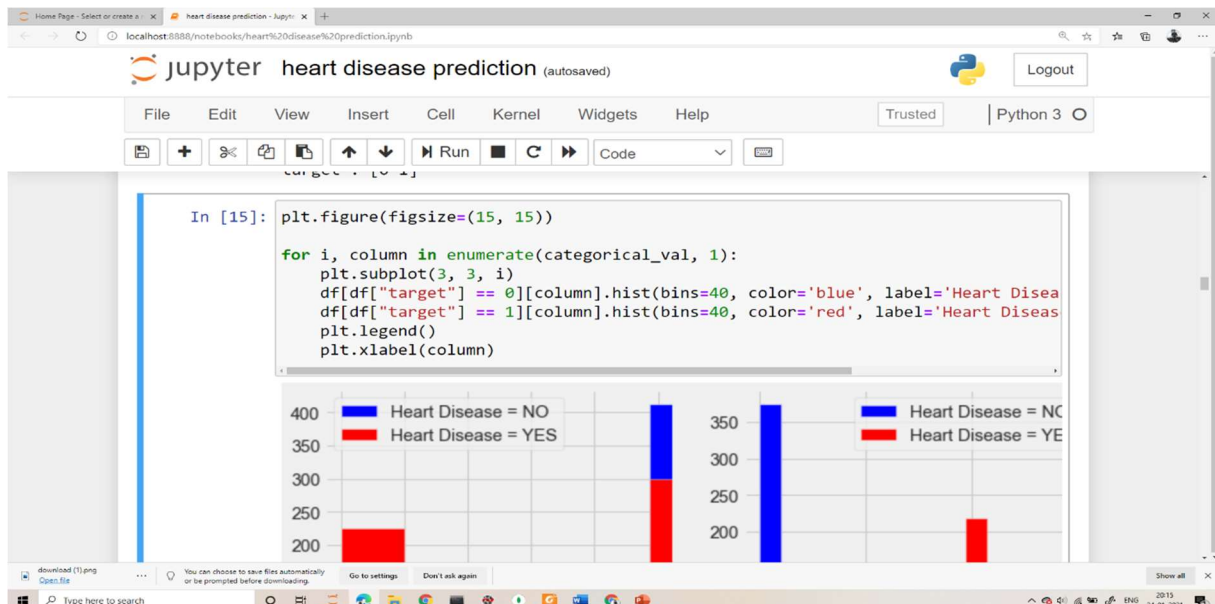


Fig:2 plot of how much people suffering with heart diseases and are not.

Now, we have to plot different types of factors to observe with plot:



Observations from the plot:

1. cp {Chest pain}: People with cp 1, 2, 3 are more likely to have heart disease than people with cp 0.
2. restecg {resting ECG results}: People with a value of 1 (reporting an abnormal heart rhythm, which can range from mild symptoms to severe problems) are more likely to have heart disease.

3. exang {exercise-induced angina}: people with a value of 0 (No ==> angina induced by exercise) have more heart disease than people with a value of 1 (Yes ==> angina induced by exercise)
4. slope {the slope of the ST segment of peak exercise}: People with a slope value of 2 (Downsloping: signs of an unhealthy heart) are more likely to have heart disease than people with a slope value of 2 slope is 0 (Upsloping: best heart rate with exercise) or 1 (Flatsloping: minimal change (typical healthy heart)).
5. ca {number of major vessels (0-3) stained by fluoroscopy}: the more blood movement the better, so people with ca equal to 0 are more likely to have heart disease.
6. Thal {thallium stress result}: People with a thal value of 2 (defect corrected: once was a defect but ok now) are more likely to have heart disease.

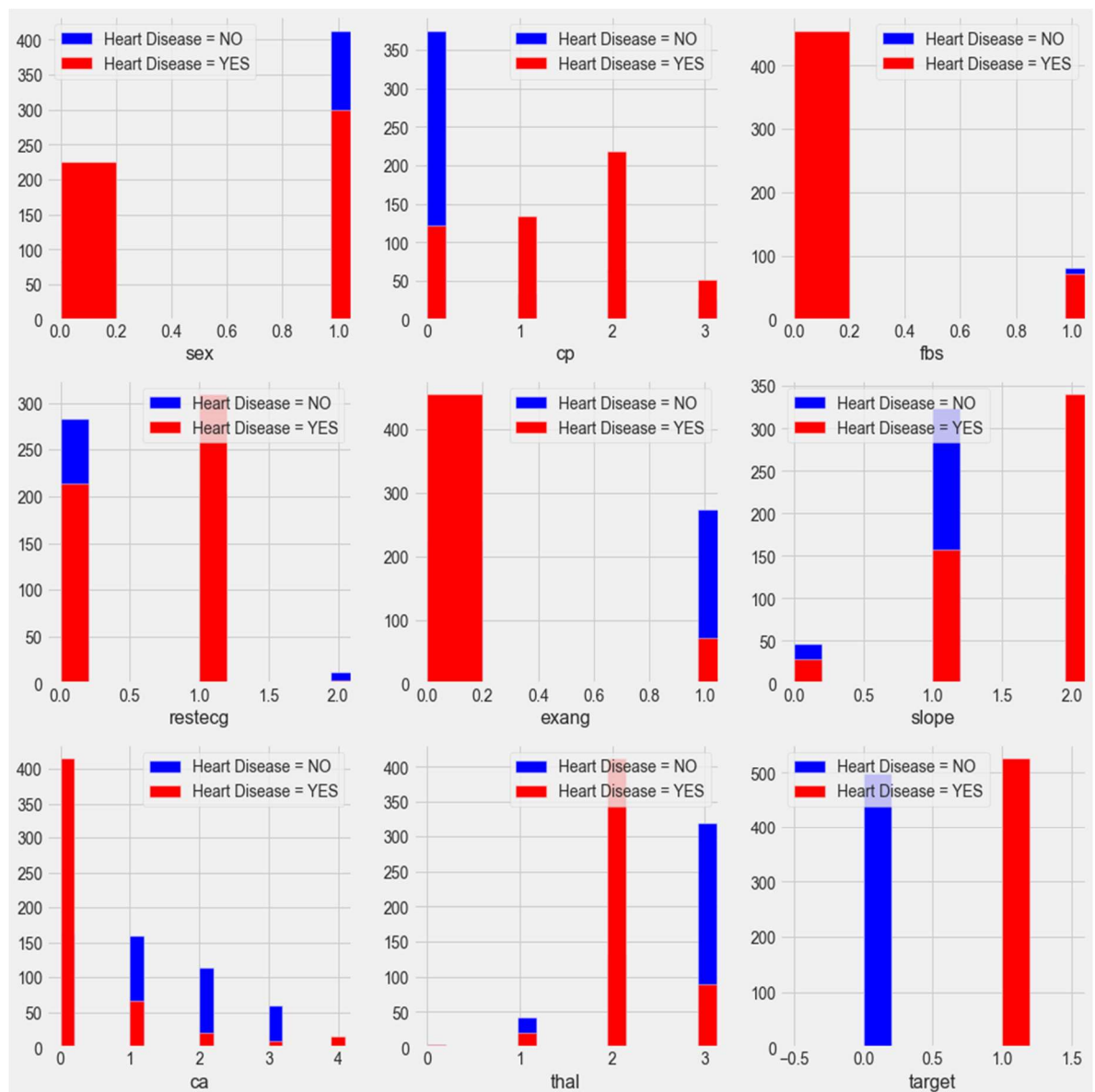
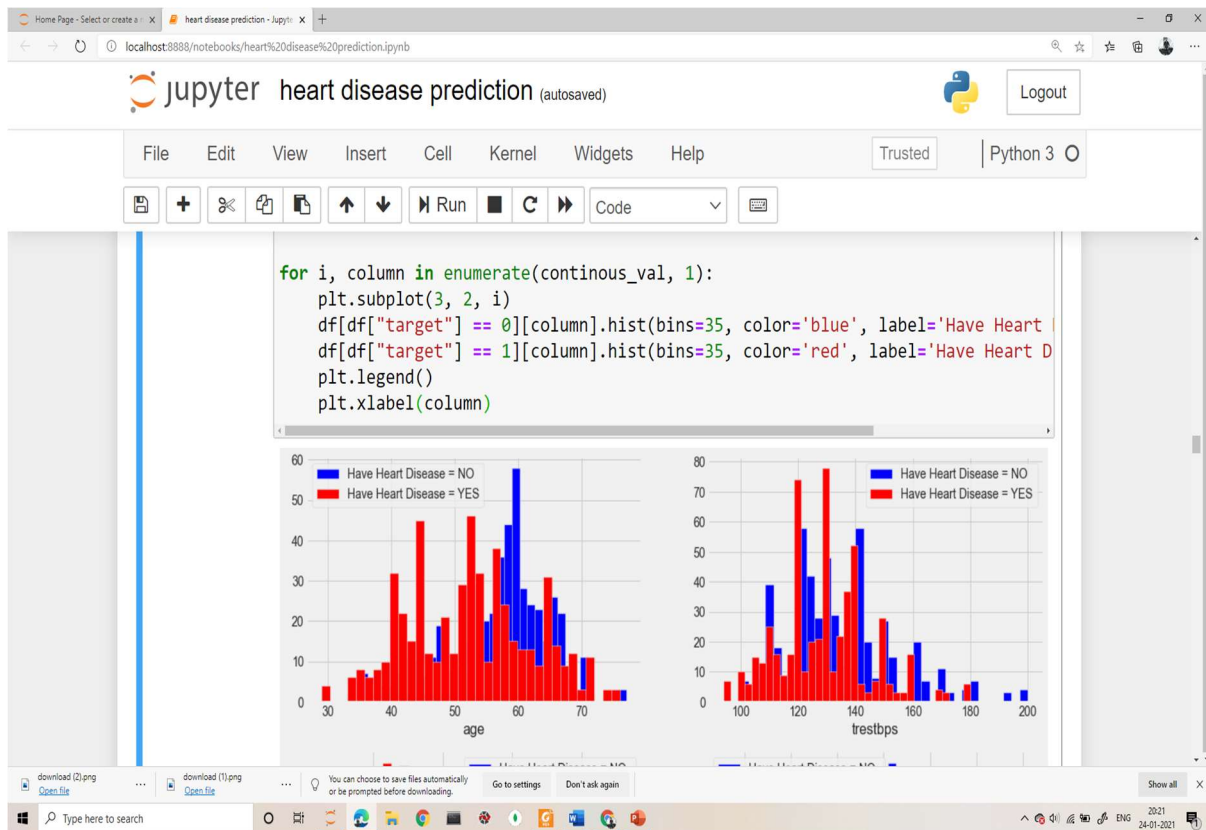


Fig no. :3 plot of different attributes with different ranges of its value to have diseases or not

So we go to the next plot to explore more about our data using exploratory data analysis



Observations from the plot:

1. trestbps: resting blood pressure anything above 130-140 is generally of concern
2. chol: greater than 200 is of concern.
3. thalach: People with a maximum of over 140 are more likely to have heart disease.
4. the old peak of exercise-induced ST depression vs. rest looks at heart stress during exercise an unhealthy heart will stress more.

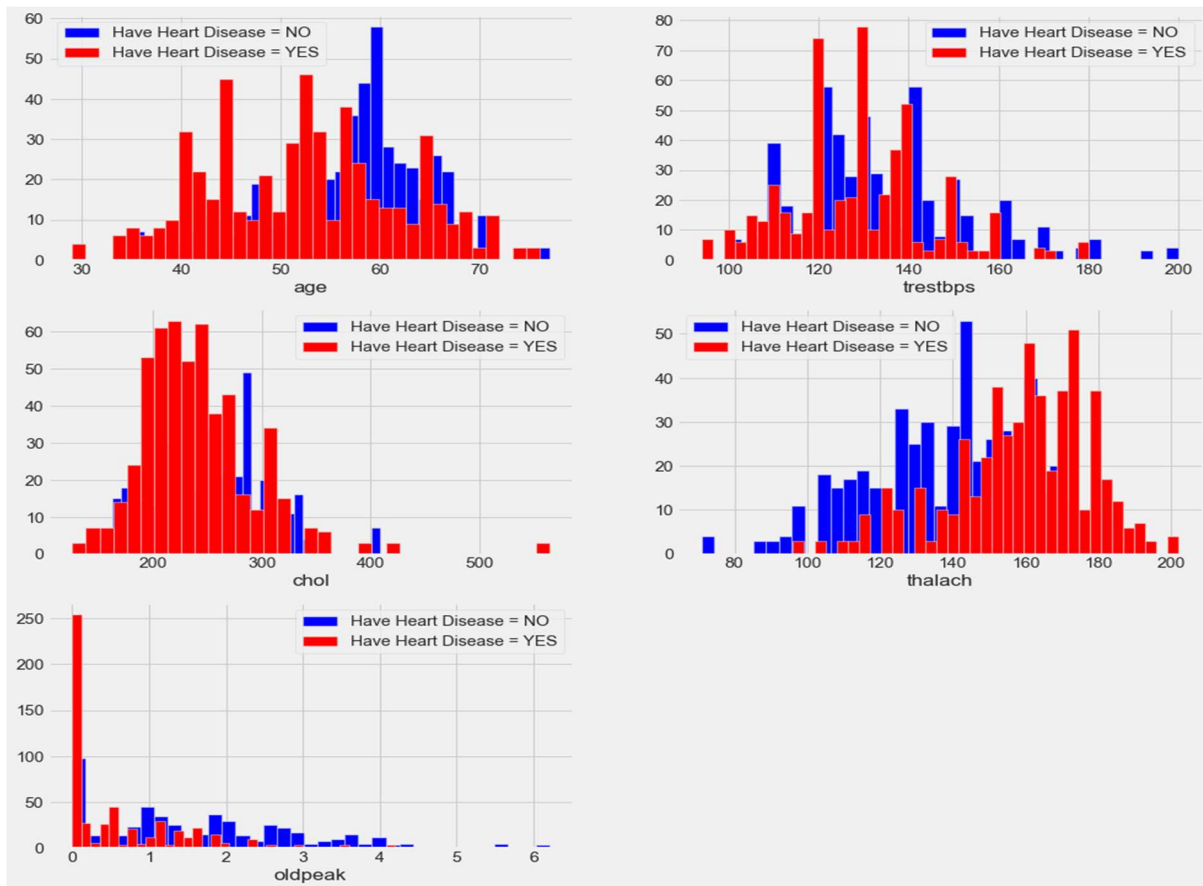
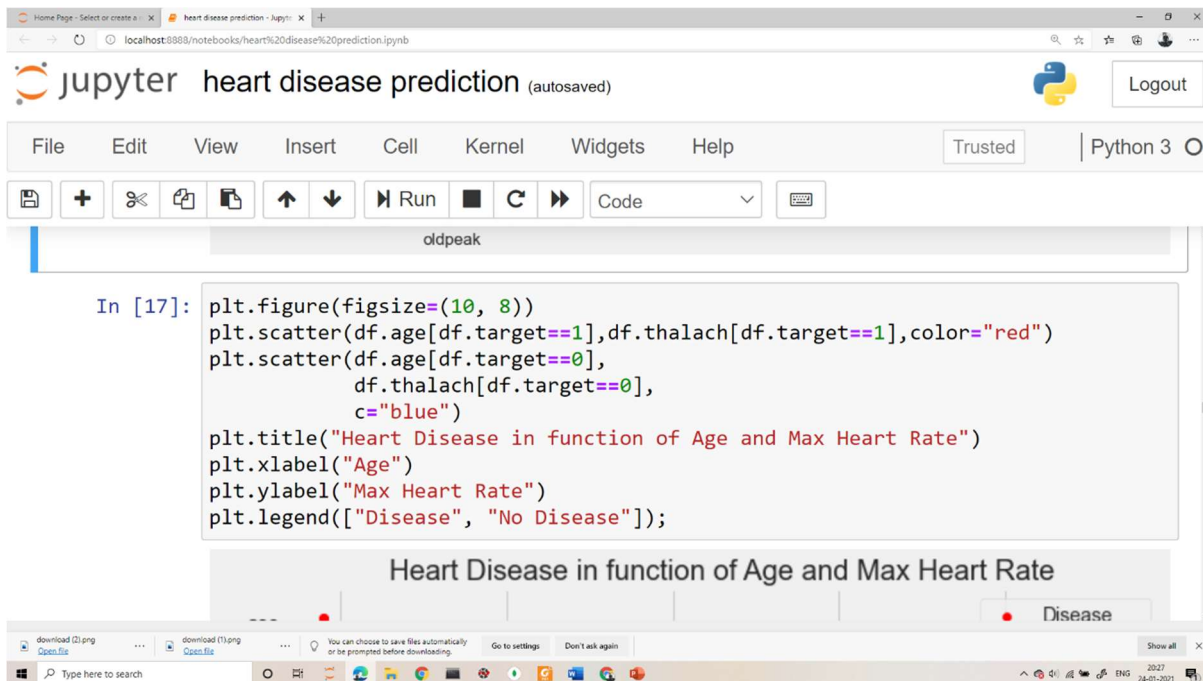


Fig:4 plot of exercise-induced ST depression vs. rest looks at heart stress



Heart disease in function of age and max heart rate whether it is suffering from disease or not.

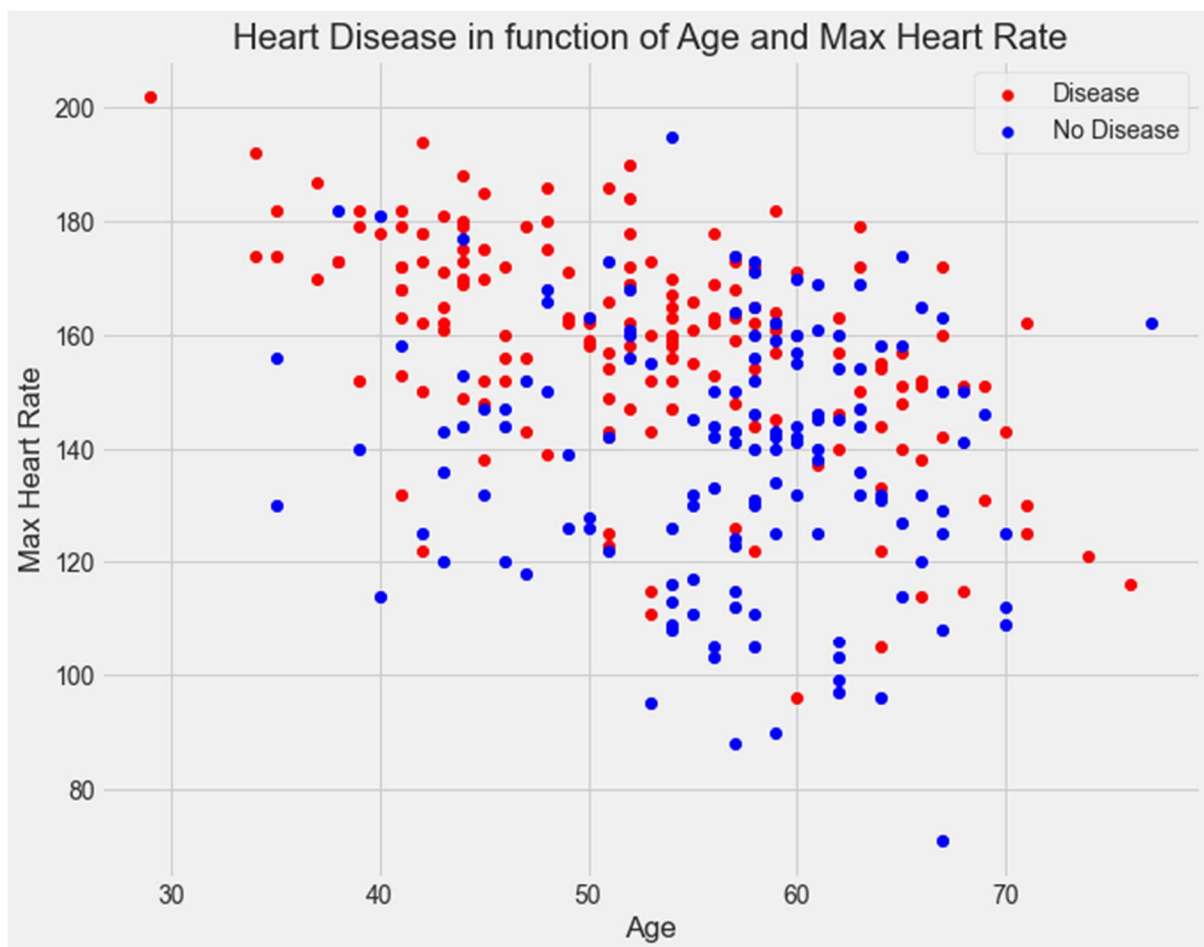
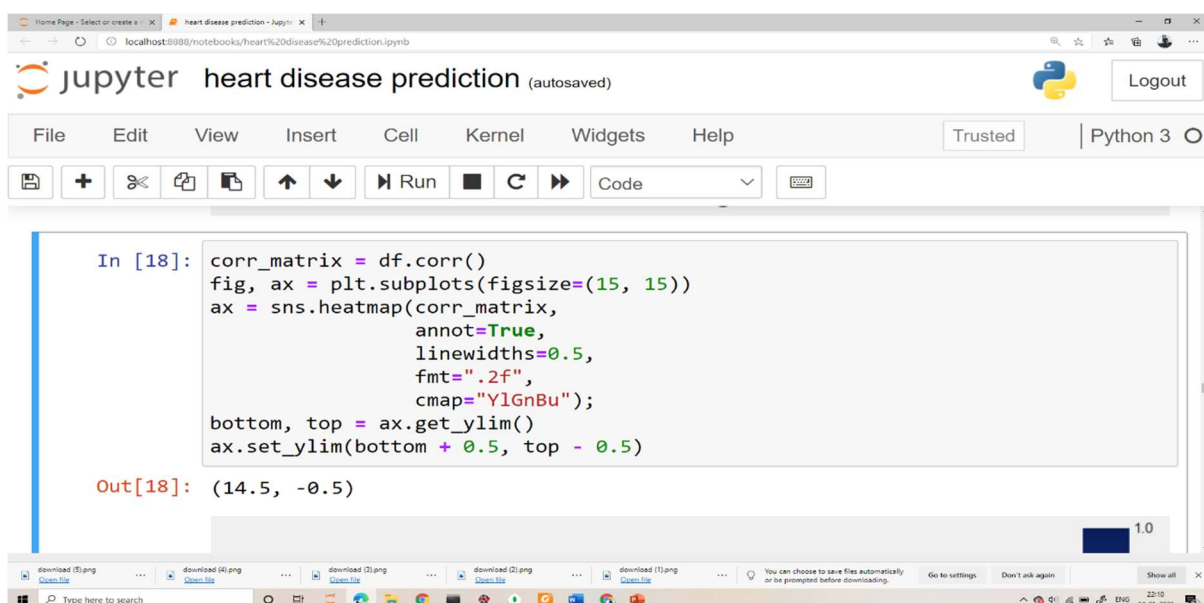


Fig:5 Heart disease in function of age and max heart rate

## Correlation Matrix





Correlation Matrix visualization Before Feature Selection shows

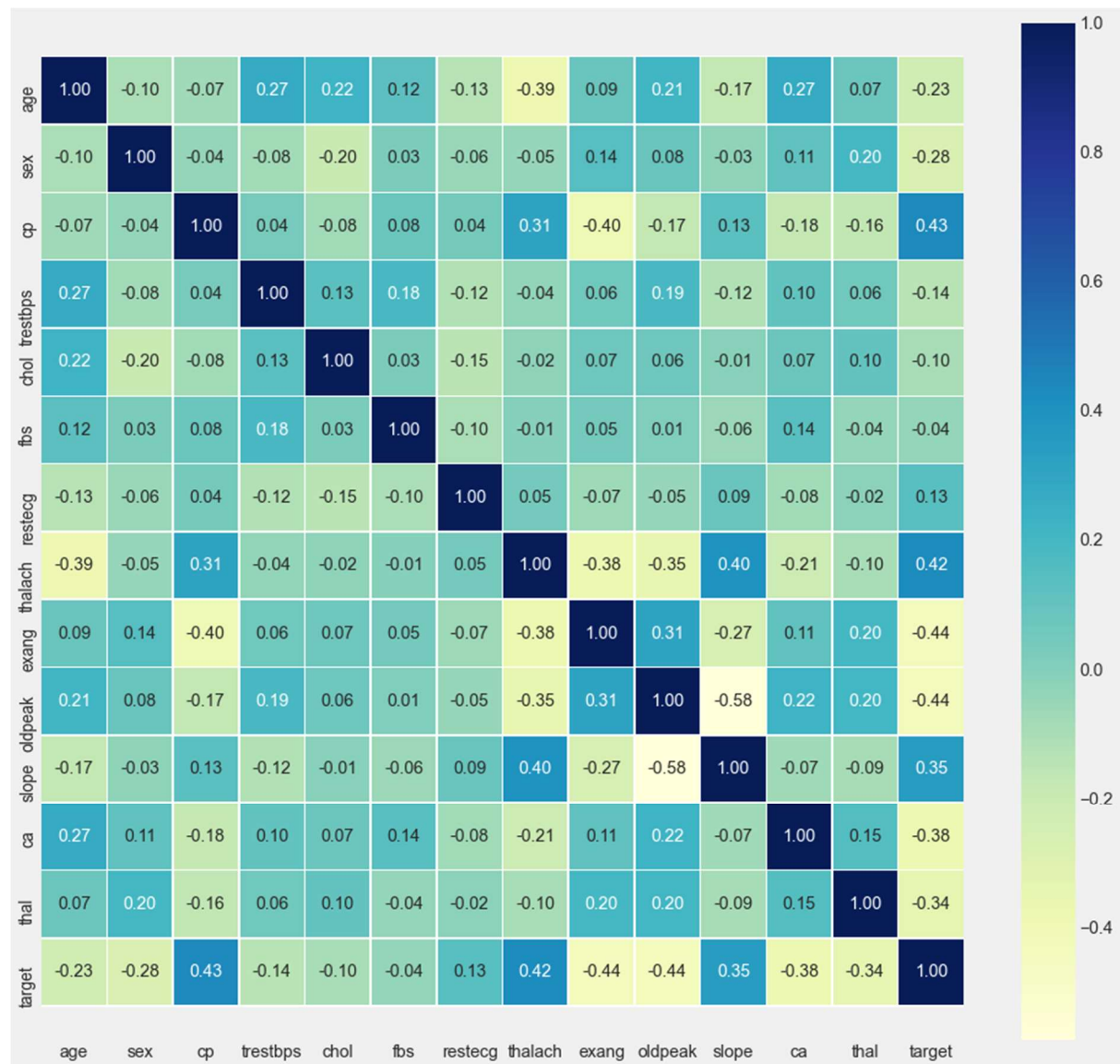


Fig no.6: Correlation Matrix Visualization

It shows 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. The data was also visualized through plots and bar graphs.

```
Home Page - Select or create a... heart disease prediction - Jupyter
localhost:8888/notebooks/heart%20disease%20prediction.ipynb

jupyter heart disease prediction (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

In [19]: df.drop('target', axis=1).corrwith(df.target).plot(kind='bar',
grid=True, figsize=(10, 8),
title="Correlation with target")
Out[19]: <matplotlib.axes._subplots.AxesSubplot at 0x1f0fa183a30>
```

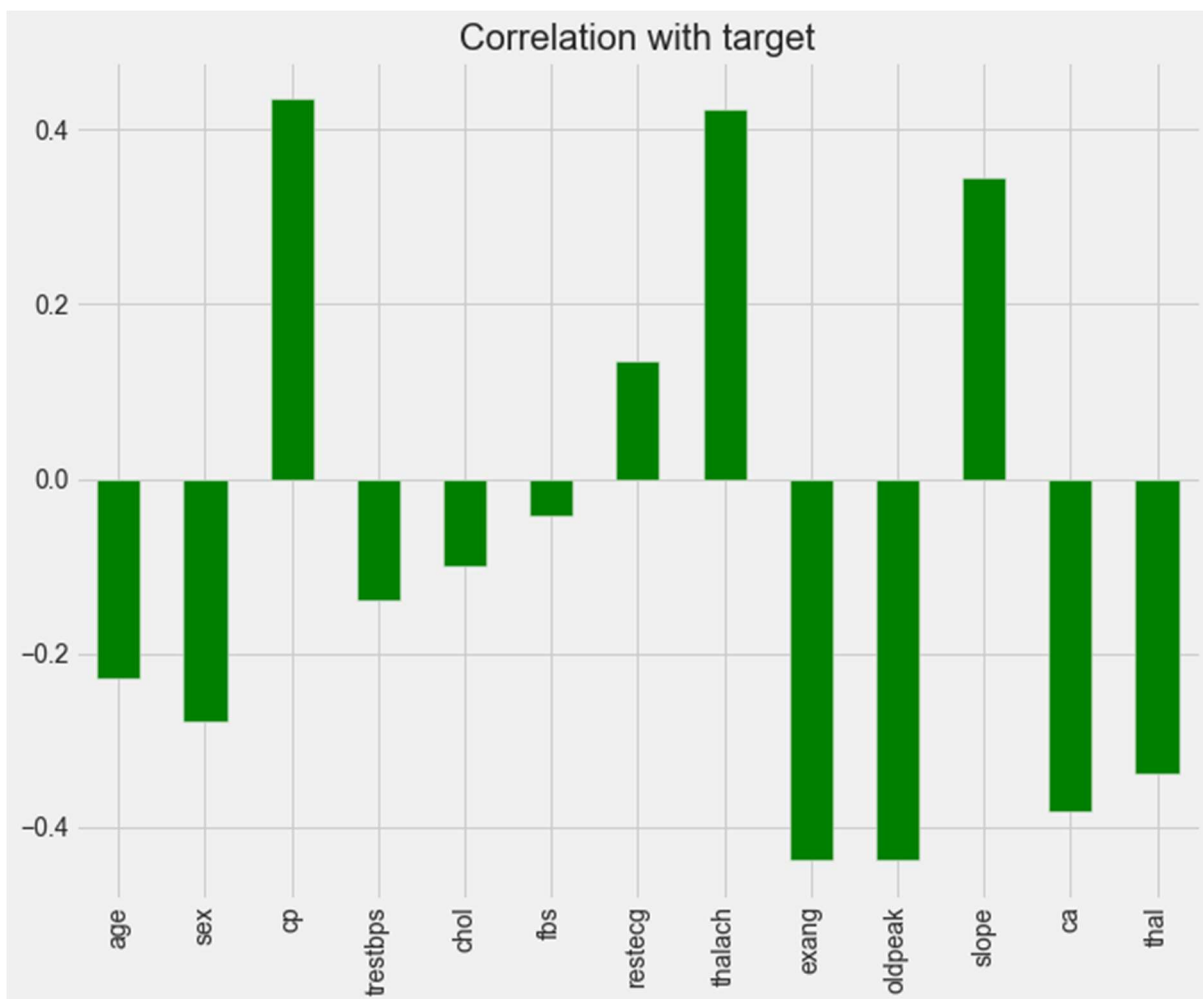


Fig no.7: correlation with target value.

Observations from correlation:

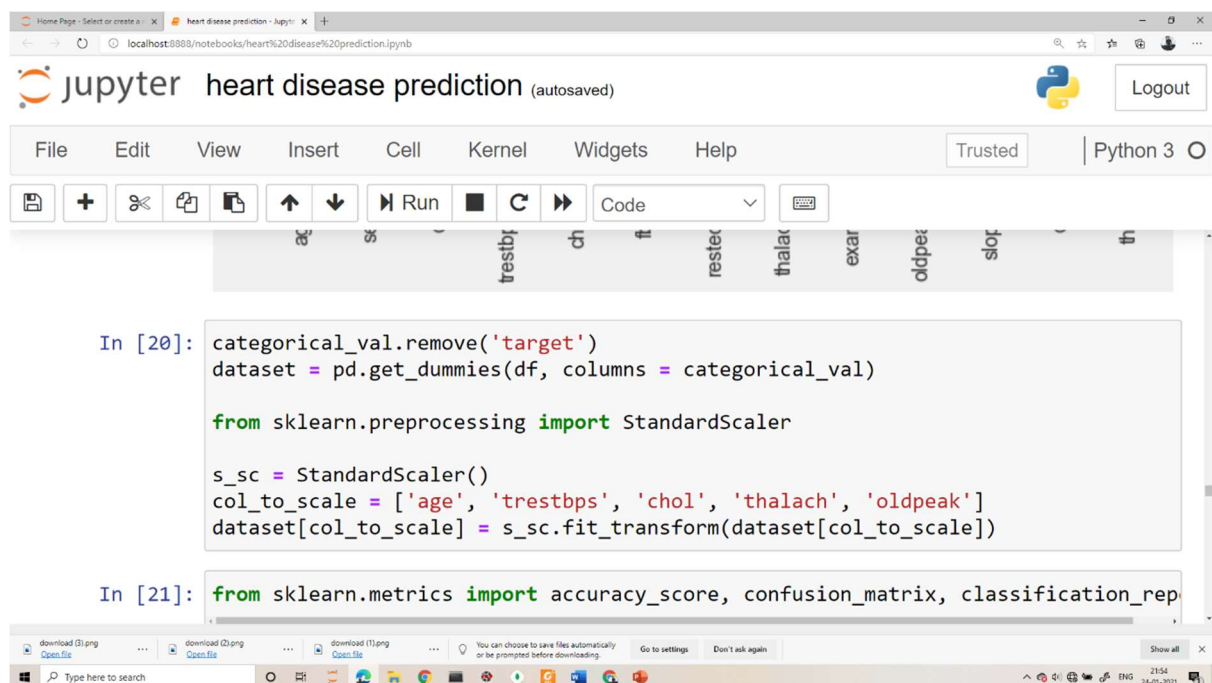
1. fbs and chol are the least correlated with the target variable.
2. All other variables have a significant correlation with the target variable.



## Data Pre-processing

Data pre-processing is an important step in the machine learning model building process because the model can perform well only when the data it is trained on is good and well prepared. Therefore, when building models this step consumes a large amount of time.

After exploring the dataset, we can observe that we need to convert some categorical variables to dummy variables and scale all values before training the machine learning models. So, for this task, I'll use the get dummies method to create dummy columns for categorical variables:



The screenshot shows a Jupyter Notebook titled "heart disease prediction (autosaved)" running on a local host. The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, running cells, and code execution. The code is written in Python and uses pandas, sklearn.preprocessing, and sklearn.metrics. It includes comments and variable names that correspond to the heart disease dataset features.

```
In [20]: categorical_val.remove('target')
dataset = pd.get_dummies(df, columns = categorical_val)

from sklearn.preprocessing import StandardScaler

s_sc = StandardScaler()
col_to_scale = ['age', 'trestbps', 'chol', 'thalach', 'oldpeak']
dataset[col_to_scale] = s_sc.fit_transform(dataset[col_to_scale])

In [21]: from sklearn.metrics import accuracy_score, confusion_matrix, classification_rep
```

- Data has been collected from Kaggle. Data Collection is the process of gathering and measuring information from countless different sources.

Kaggle:

- Kaggle is an online community of data scientists and machine learners, owned by Google LLC. Kaggle allow users to find and publish data sets. Explore and build models in a web-based data-science environment.

The main purpose of designing this system is to predict the ten-year risk of future heart disease. we have used Logistic regression as a machine-learning algorithm to train our system.

## LOGISTIC REGRESSION:

- Logistic Regression is a supervised classification algorithm. It is a predictive analysis algorithm based on the concept of probability. It measures the relationship between the dependent variable and the one or more independent variables (risk factors) by estimating probabilities using underlying logistic function (sigmoid function). Sigmoid function is used as a cost function to limit the hypothesis of logistic regression between 0 and 1 (squashing) i.e.  $0 \leq h\theta(x) \leq 1$ .

- In logistic regression cost function is defined as:

$$\text{Cost}(h\theta(x), Y(\text{actual})) = -\log(h\theta(x)) \text{ if } y=1$$

$$-\log(1 - h\theta(x)) \text{ if } y=0$$

- Logistic Regression relies highly on the proper presentation of data. So, to make the model more powerful, important features from the available data set are selected using Backward elimination and recursive elimination techniques.

- model

Output = 0 or 1

Hypothesis  $\Rightarrow Z = WX + B$

$h\theta(x) = \text{sigmoid}(Z)$

### Sigmoid Function

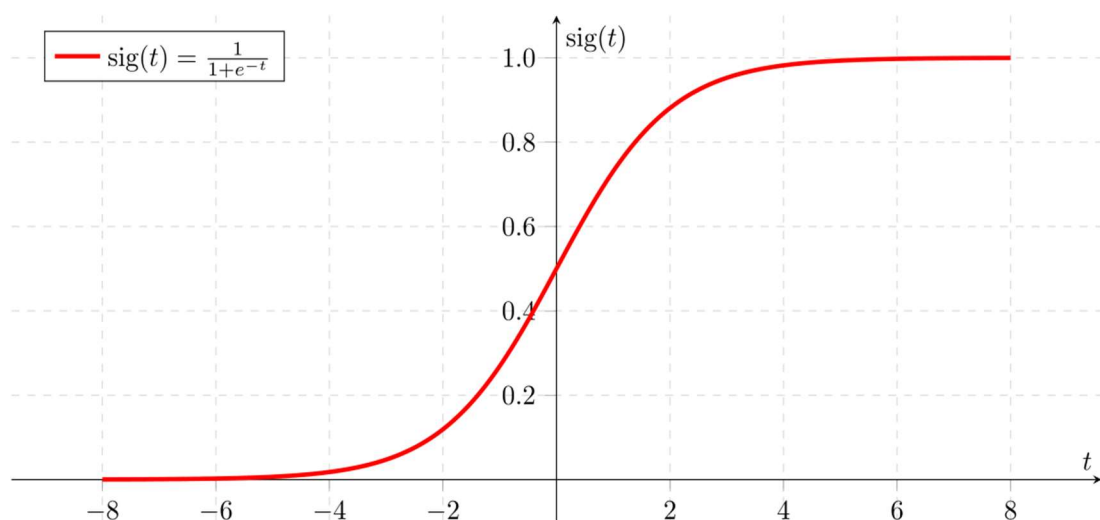


Fig no.8: sigmoid function

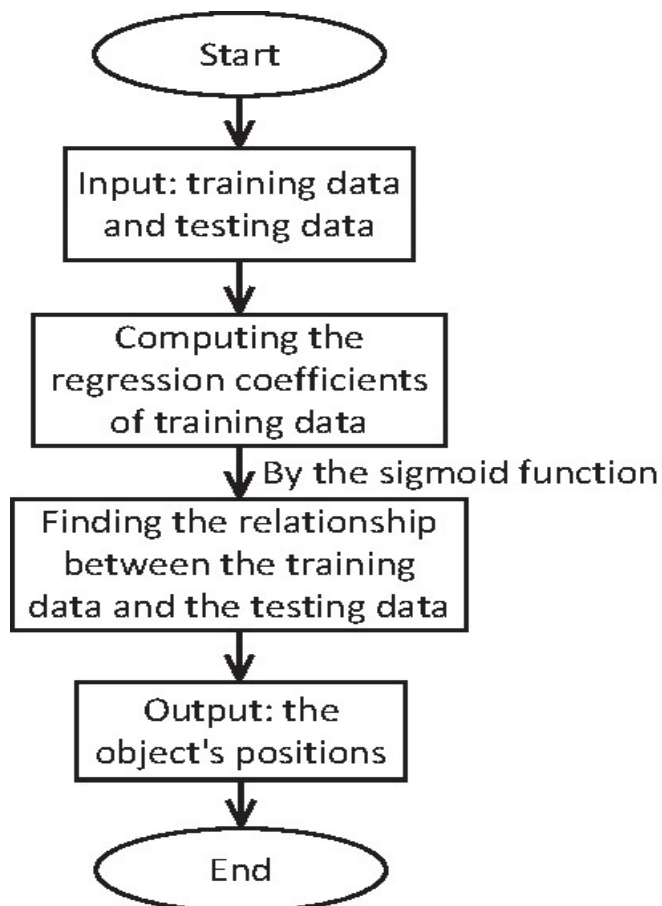
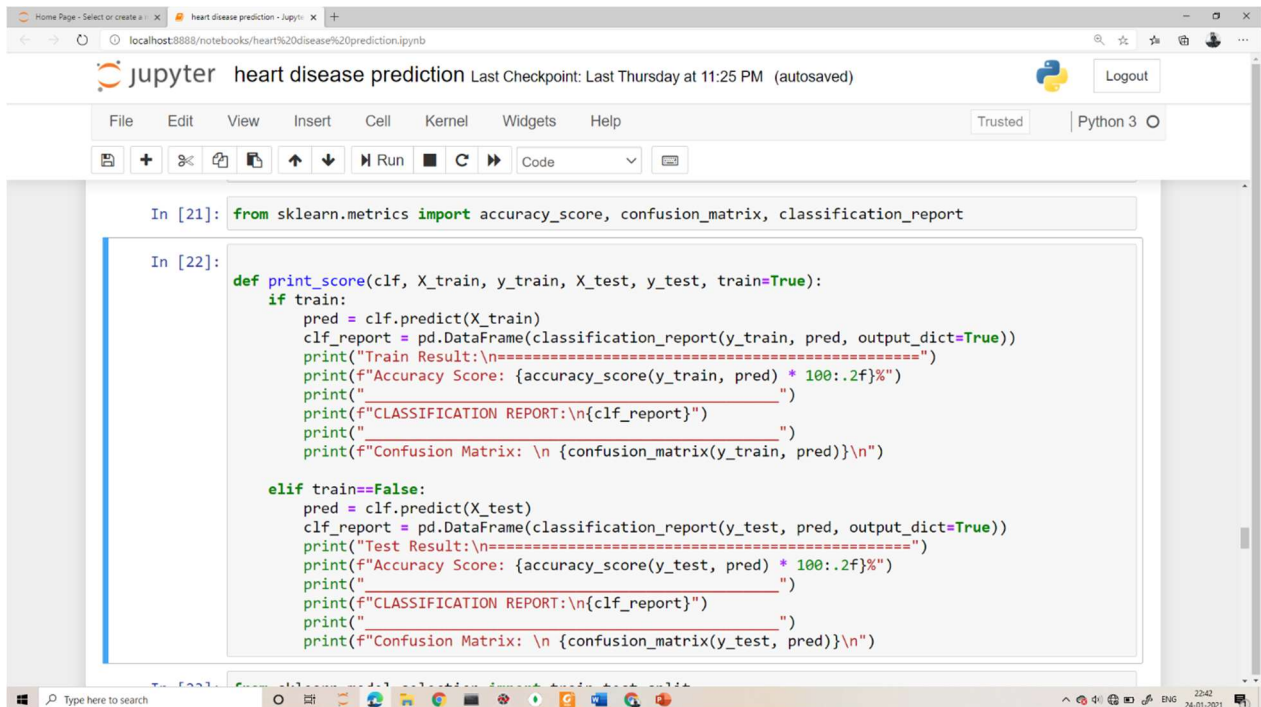


fig no.9 flow chart of working of logistic regression

## Applying Logistic Regression

Now, I will train a machine learning model for the task of heart disease prediction. I will use the logistic regression algorithm as I mentioned at the beginning of the article.

But before training the model I will first define a helper function for printing the classification report of the performance of the machine learning model:



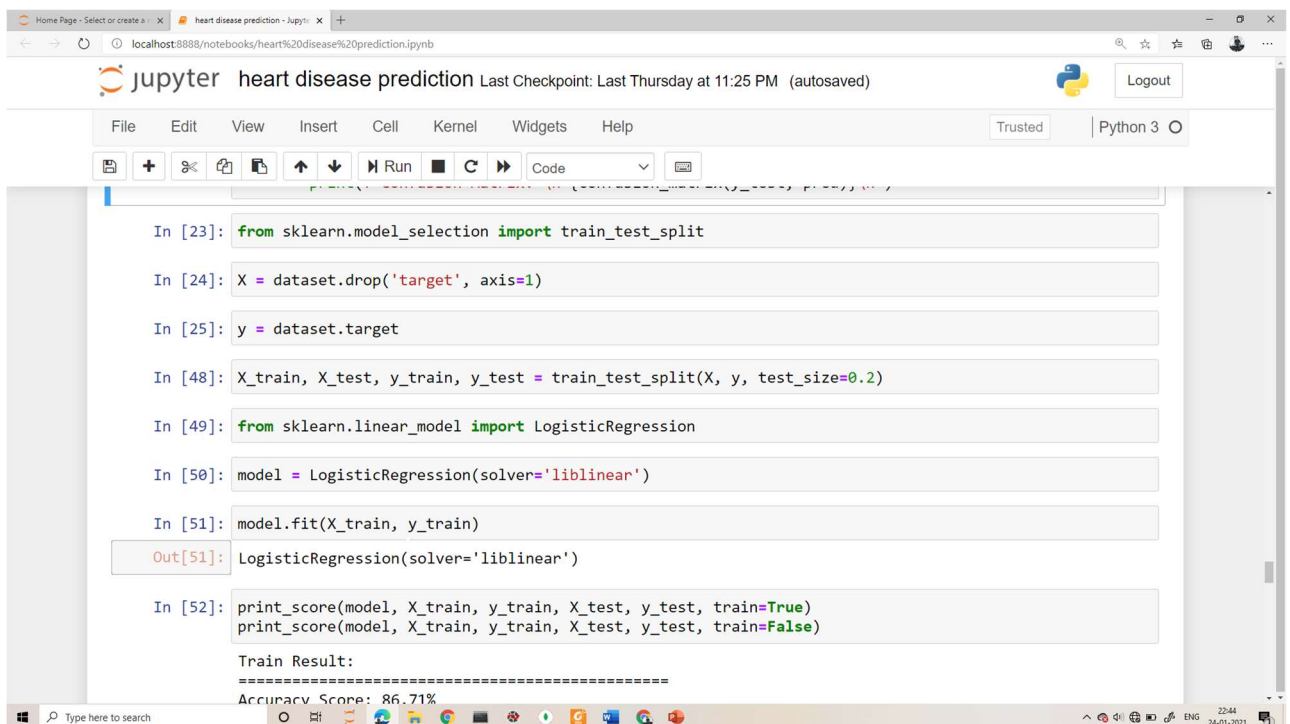
```
In [21]: from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

In [22]: def print_score(clf, X_train, y_train, X_test, y_test, train=True):
    if train:
        pred = clf.predict(X_train)
        clf_report = pd.DataFrame(classification_report(y_train, pred, output_dict=True))
        print("Train Result:\n=====")
        print(f"Accuracy Score: {accuracy_score(y_train, pred) * 100:.2f}%")
        print("=====")
        print(f"CLASSIFICATION REPORT:\n{clf_report}")
        print("=====")
        print(f"Confusion Matrix: \n {confusion_matrix(y_train, pred)}\n")

    elif train==False:
        pred = clf.predict(X_test)
        clf_report = pd.DataFrame(classification_report(y_test, pred, output_dict=True))
        print("Test Result:\n=====")
        print(f"Accuracy Score: {accuracy_score(y_test, pred) * 100:.2f}%")
        print("=====")
        print(f"CLASSIFICATION REPORT:\n{clf_report}")
        print("=====")
        print(f"Confusion Matrix: \n {confusion_matrix(y_test, pred)}\n")
```

## Splitting of data: Training and testing

Finally, this resulting data split into 80% train and 20% test data, which was further passed to the Logistic Regression model to fit, predict and score the model



```
In [23]: from sklearn.model_selection import train_test_split

In [24]: X = dataset.drop('target', axis=1)

In [25]: y = dataset.target

In [48]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

In [49]: from sklearn.linear_model import LogisticRegression

In [50]: model = LogisticRegression(solver='liblinear')

In [51]: model.fit(X_train, y_train)

Out[51]: LogisticRegression(solver='liblinear')

In [52]: print_score(model, X_train, y_train, X_test, y_test, train=True)
print_score(model, X_train, y_train, X_test, y_test, train=False)

Train Result:
=====
Accuracy Score: 86.71%
```

## EVALUATION METRICS

For the evaluation of our output from our training the data, the accuracy was analysed “Confusion matrix”.

### Confusion Matrix

A confusion matrix, also known as an error matrix, is a table that is often used to describe the performance of a classification model (or “classifier”) on a set of test data for which the true values are known. It allows the visualization of the performance of an algorithm. It allows easy identification of confusion between classes e.g. one class is commonly mislabeled as the other. The key to the confusion matrix is the number of correct and incorrect predictions are summarized with count values and broken down by each class not just the number of errors made.

Output of previous code:

Train Result:

=====  
Accuracy Score: 86.71%

CLASSIFICATION REPORT:

	0	1	accuracy	macro avg	weighted avg
precision	0.89	0.85	0.87	0.87	0.87
recall	0.82	0.91	0.87	0.87	0.87
f1-score	0.86	0.88	0.87	0.87	0.87
support	394.00	426.00	0.87	820.00	820.00

Confusion Matrix:

```
[[324  70]
 [ 39 387]]
```

Test Result:

=====  
Accuracy Score: 86.83%

CLASSIFICATION REPORT:

	0	1	accuracy	macro avg	weighted avg
precision	0.89	0.85	0.87	0.87	0.87
recall	0.85	0.89	0.87	0.87	0.87
f1-score	0.87	0.87	0.87	0.87	0.87
support	105.00	100.00	0.87	205.00	205.00

Confusion Matrix:

```
[[89 16]
 [11 89]]
```

### Accuracy

The accuracy is calculated as:

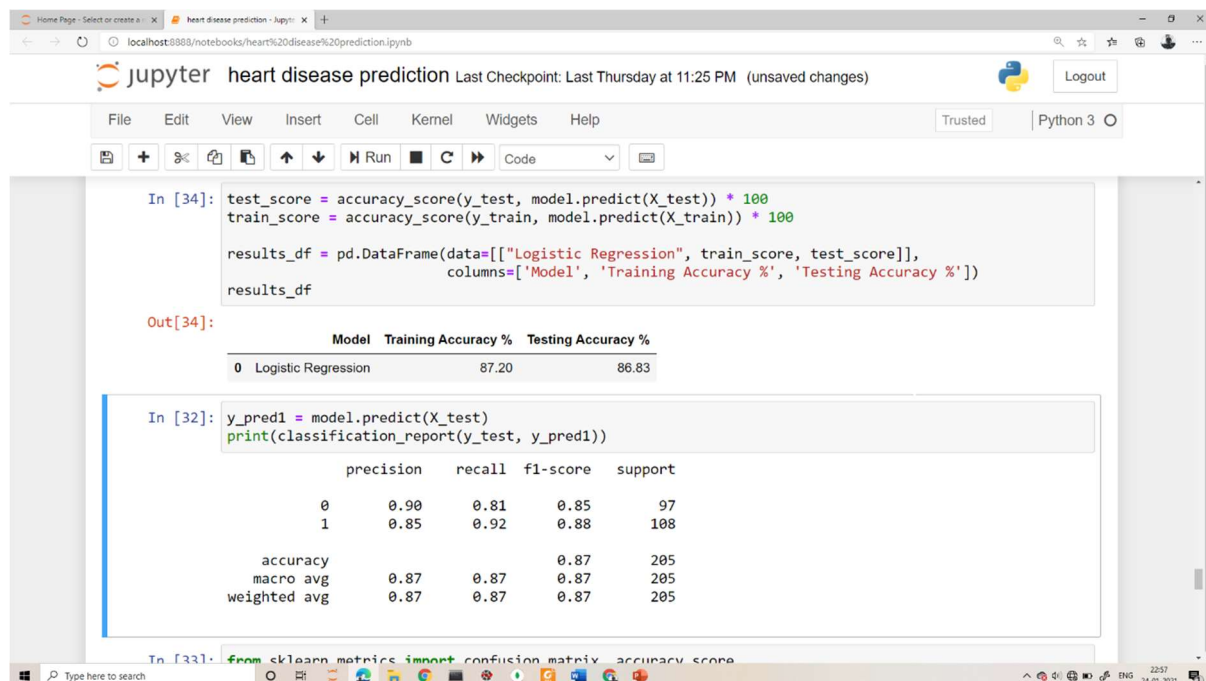
$$\text{Accuracy} = (TP+TN) \setminus (TP+TN+FP+FN)$$

Where,

- True Positive (TP) = Observation is positive and is predicted to be positive.
- False Negative (FN) = Observation is positive but is predicted negative.
- True Negative (TN) = Observation is negative and is predicted to be negative.
- False Positive (FP) = Observation is negative but is predicted positive.

## Evaluating and visualizing model performance

The prediction results acquired in the previous step are compared using what the actual results should have been. Several evaluation metrics are generated to calculate the performance of the model.



```
In [34]: test_score = accuracy_score(y_test, model.predict(X_test)) * 100
train_score = accuracy_score(y_train, model.predict(X_train)) * 100

results_df = pd.DataFrame(data=[["Logistic Regression", train_score, test_score]],
                           columns=['Model', 'Training Accuracy %', 'Testing Accuracy %'])
results_df

Out[34]:
```

	Model	Training Accuracy %	Testing Accuracy %
0	Logistic Regression	87.20	86.83

```
In [32]: y_pred1 = model.predict(X_test)
print(classification_report(y_test, y_pred1))
```

```

              precision    recall  f1-score   support

     0       0.90      0.81      0.85         97
     1       0.85      0.92      0.88        108

 accuracy          0.87
 macro avg          0.87
 weighted avg       0.87
```

```
In [33]: from sklearn.metrics import confusion_matrix, accuracy_score
```

## Testing Technologies & python libraries:

The coding portion were carried out to prepare the data, visualize it, pre-process it, building the model and then evaluating it. The code has been written in Python programming language using Jupyter Notebook as IDE. The experiments and all the models building are done based on python libraries.

- ▶ Anaconda(python): Anaconda is a free and open-source distribution of the python and R programming languages for scientific computing and many more.
- ▶ Jupyter Notebook: The Jupyter Notebook is an open-source web-application that allow you to create and share documents that contain live code, equations, visualizations and narrative text.
- ▶ NumPy
- ▶ SciPy
- ▶ Matplotlib (pyplot)
- ▶ Statsmodels
- ▶ Pandas
- ▶ Sklearn

## CONCLUSION

The number of heart diseases can go beyond the control line and reach the peak. Heart disorders are difficult, and number of people suffer each year from this condition. While utilizing these methods one of the key drawbacks of these works is relying exclusively on the specification, with all this researching different data cleaning and mining technologies, of classifications strategies and algorithms to forecast heart disease. So that I can use this machine learning algorithms by predicting whether not a patient has heart disease in various machine learning algorithms. Any non-medical personnel may use this process to forecast heart failure to reduce doctors' time-complexity. It shows the efficiency of the proposed procedure for classifying Dataset with correct results.

## PROJECT SUMMARY

Predicting and diagnosing heart disease is the biggest challenge in the medical industry and relies on factors such as the physical examination, symptoms and signs of the patient. Factor that influences heart disease are body cholesterol levels, smoking habit and obesity, family history of illnesses, blood pressure, and work environment. Machine learning algorithms play an essential and precise role in the prediction of heart disease.

Advances in technology allow machine language to combine with Big Data tools to manage unstructured and exponentially growing data. Heart disease is seen as the world's deadliest disease of human life. Particularly, in this type of disease, the heart is not able to push the required amount of blood to the remaining organs of the human body to perform regular functions.

The health industry creates huge volumes of data every day. But most of it is not used effectively. Efficient methods to obtain information from such repositories are not widespread for clinical disease diagnosis or other purposes. This project aims at comparing specific approaches for forecasting cardiac diseases using data mining techniques.

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.

## REFERENCES

- [1] A. L. Bui, T. B. Horwich, and G. C. Fonarow, —Epidemiology and risk profile of heart failure, *Nature Reviews Cardiology*, vol. 8, no. 1, pp. 30–41, 2011
- [2] Vikas Chaurasia, Saurabh Pal, —Early Prediction of Heart disease using Data mining Techniques, *Caribbean journal of Science and Technology*, 2013
- [3] Chaitrali S. Dangare et al, —Improved Study of Heart Disease Prediction System using Data Mining Classification Techniques, *International Journal of Computer Application* Vol.47, No.10, pg.no:44 – 48, 2012.
- [4] Poornima Singh et al, —Effective heart disease prediction system using data mining techniques, *International Journal of Nano medicine*, pg.no:121- 124, 2018.
- [5] W.J. Frawley and G. Piatetsky-Shapiro, —Knowledge Discovery in Databases: An Overview, *AI Magazine*, Vol.
- [6] J. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Scholkopf, C. Burges, and A. Smola, editors *Advances in kernel methods: support vector learning*. MIT Press, 1998.
- [7] Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121{167, 1998. URL [citeseer.nj.nec.com/burges98tutorial.html](http://citeseer.nj.nec.com/burges98tutorial.html).