Bachelor of Computer Applications

Trimester VII

Academic Year 2021-2022

MINOR PROJECT

School of Computer Science

Faculty of Science

“Machine learning heart disease prediction PROJECT ”

Submitted by

S.No.	PRN	Name
1.	1132190421	Ketki.G
2.	1132190393	Preeti.G
3.	1132190387	Archana.p

Under the guidance of

Prof:madhuri joshi

Asst. Prof., School of Computer Science

Date: 14/05/22

CONTENTS

Sno Topic

1.INTRODUCTION

It is probably worth knowing that machine learning is not a new concept. Machine learning's growing popularity is primarily due to increase in data availability and advancements in technology. Faster machines and smarter algorithms are implemented daily. The amount of data stored in the servers is growing at an exponential rate. This data is valuable and can help us make better decisions in the future.

- **What is Machine Learning?**

Machine learning is a branch of Artificial Intelligence. It is a set of algorithms that learn to discover trends and patterns in data to gain insights. These algorithms then become self-sufficient to make decisions on the data. Machine learning algorithms are now utilized in nearly all sectors – from healthcare to financial organizations top anti-fraud companies to shopping websites. Machine learning is about getting machines to learn data and then make decisions on similar data. Machine learning is about using predictive algorithms to forecast the behaviors of data so that calculated decisions can be taken.

Abstract

Heart disease is among the most serious problems that the world is facing today. In the field of clinical predictive analytics, predicting cardiovascular disease is a major problem. Machine Learning (ML) has shown to be useful in generating choices and forecasts based on vast amounts of data generated by the healthcare industry and hospitals. ML methods have also been utilised in recent advances in many sectors of the Internet of Things (IoT). Various research shows just a sliver of what can be done using machine learning to forecast cardiac disease. In this dissertation there is a narrative approach for identifying important characteristics using machine learning methods, which improves the precision of cardiovascular disease prediction. A prediction model is presented that incorporates a variety of variables and classification methods.

It is the purpose of this section to expand on the work performed by many other academics in the area of Machine Learning in Cardiovascular Disease. It will include an introduction to Machine Learning and an examination of the role played by ML in the detection of diseases in general. Then, this section will delve deeper into Machine Learning models such as Logistic regression, Decision Tree, and others, all of which are used in the prediction of heart disease. Furthermore, a study was conducted on the use of Hybrid Machine Learning methods to detect heart-related issues, and then research gaps were identified as a result of the research carried out.

1.1 Problem Statement

The leading cause of mortality in the world over the last decade has consistently been heart failure or cardiovascular disease (CVD). According to the World Health Organization, approximately 17.9 million people die worldwide each year as a result of cardiovascular illness, with coronary heart disease and vascular stroke accounting for 80% of all deaths (Kononenko, 2001).

Machine learning techniques have the potential to enable everyone to analyse all the essential data that we need. Some results indicates that some patients may have a reaction, which would provide a significant boost to the diagnostic process. More and more applications of machine learning in the NHS have been shown to be effective in easing the rendering of choices and predictions from the enormous amount of data supplied by the healthcare industry .FORINSTANT DIAGNOSIS AND TO SAVE PATIENTS AND DOCTORS EFFORTS WE DECIDED TO GO AHEAD WITH THIS PROJECT

1.2 About the Project

The overall scope of this study is to check for and/or identify if a patient has heart disease (heart disease is present or absent), utilise machine learning algorithms to run a comparative study, and generate a magnificent web interface for the operator to check his/her own likelihood of having heart disease.

Heart disease is among the most serious problems that the world is facing today. In the field of clinical predictive analytics, predicting cardiovascular disease is a major problem. Machine Learning (ML) has shown to be useful in generating choices and forecasts based on vast amounts of data generated by the healthcare industry and hospitals. ML methods have also been utilised in recent advances in many sectors of the Internet of Things (IoT). Various research shows just a sliver of what can be done using machine learning to forecast cardiac disease. In this dissertation there is a narrative approach for identifying important characteristics using machine learning methods, which improves the precision of cardiovascular disease prediction.

A prediction model is presented that incorporates a variety of variables and classification methods.
The report also conducts a comparative analysis among Logistic Regression, Decision Tree, Random Forest, K-nearest Neighbour, and Artificial Neural Network.

The dataset that is used in this report is the Cleaveland UCI dataset of heart disease with actual patient data (no personal information is there). Towards the end of the report, the proposed method for heart disease using Logistic Regression yields an improved efficiency level with 88% accuracy level as compared to other algorithms LR performs the best.

The objective of the project consists of:

- *Investigate and comprehend a literature review.*
- *Predict heart disease using multiple ML algorithms like Logistic Regression, Decision Tree, Random Forest, K-nearest Neighbour, and Artificial Neural Network.*
- *Concluding and finding the results through comparing these algorithms and finding the best one*

Technologies Used & requirements

Speaking of requirement to run machine learning code in a computer these hardware and software requirement are needed as written down below. To run basic python programming, we need Processors: Intel® CoreTM i5 processor 4300M at 2.60 GHz or 2.59 GHz (1 socket, 2 cores, 2 threads per core), 8 GB of DRAM Intel® Xeon® processor E5- 2698 v3 at 2.30 GHz (2 sockets, 16 cores each, 1 thread per core), 64 GB of DRAM Intel® Xeon PhiTM processor 7210 at 1.30 GHz (1 socket, 64 cores, 4 threads per core), 32 GB of DRAM, 16 GB of MCDRAM (flat mode enabled) Disk space: 2 to 3 GB

Talking about software requirement for this specific piece of code

a. Operating systems: Windows® 10, macOS* and Linux*

b. Python IDLE

C.pip

d. NumPy

e. Pandas

f. Matplotlib

g. Seaborn

h. Scikit-learn

i. Joblib or Pickle

j. Flutter SDK (Software Development Kit)

k. Visual Studio Code (VS code)

l. Google collaborated with

LITERATURE REVIEW :-

Author name :	(QIAN wang , CAO LI, &jiang) 2015
Abstract:- Machine learning implemented to medical history, in specific, can be a powerful method both to forecast the survival of any patients experiencing heart failure signs that can contribute to heart problems. Scientists should take advantage of the use of machine learning in medical prediction (note: predictive modelling) as well as feature ranking (note: feature selection). Computational intelligence (CI), particularly in its capacity to forecast, is seen in the health records, as well as in the brain scans of patients.	
The Machine learning model used didn't involve hyper parameter tuning or dataset preprocessing to obtain the optimal working model. No EDA was involved in the research paper.	

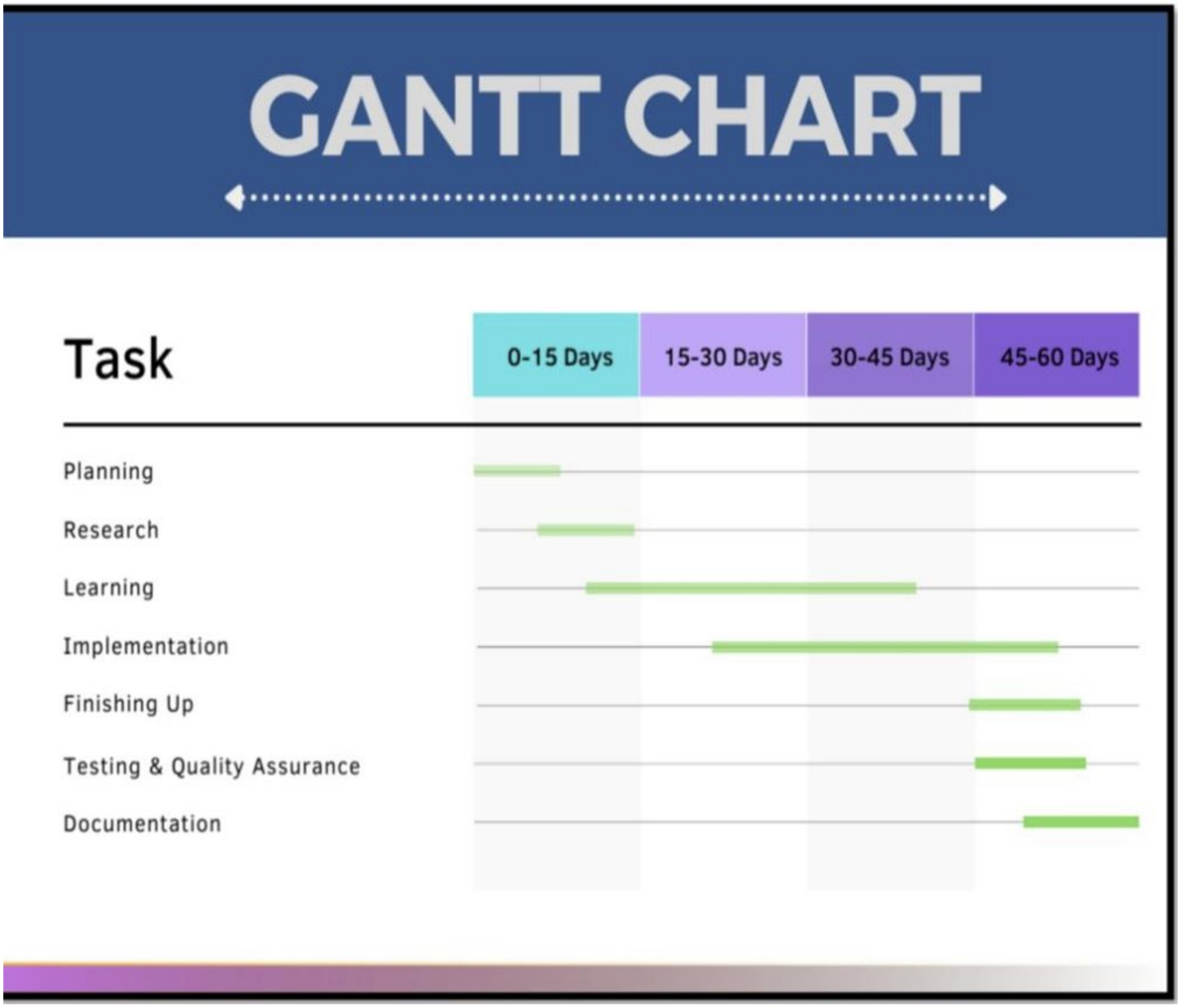
Author name :	*Ahmad, Munir, Bhatti, Aftab, & Raza, 2017)*

performed a research in which they examined the medical records of 299 people who were suffering from heart failure. The findings were published in the journal Heart Failure. The Mayo Clinic in Rochester, Minnesota, sent out mailed lists to 64,000 male patients, which were then analysed. Later, the authors used conventional descriptive stats to calculate heart disease mortality, determine who was at higher risk of dying, and identify variables that influence deaths

In the paper (Mohan, Thirumalai, & Srivastava, 2019)

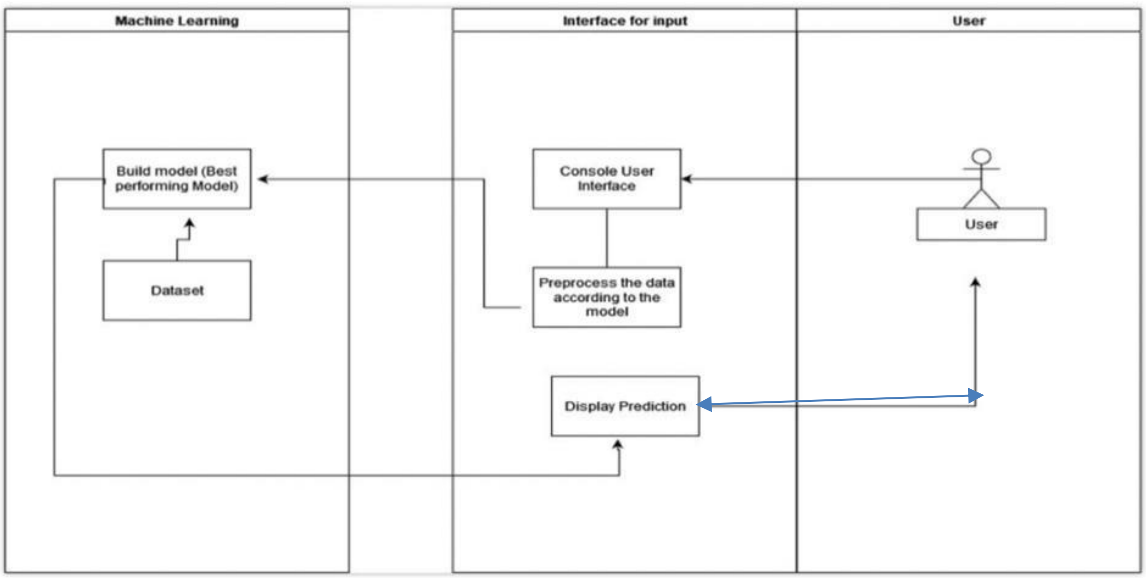
authors have used hybrid Machine Learning Techniques like LR, DT, and ANN to identify heart diseases. With the aid of Rattle, they have done a heart disease identification of the Cleveland UCI server. In relation to the data, the predictive analytics model, and the workers climate, it will offer an easy-to-view visual representation. The pre-processing stage begins first when taking input data. The features are chosen dependent on DT entropy, and the classification output is tested on a workload or an appraisal, and their results are improved.

PLAN OF WORK

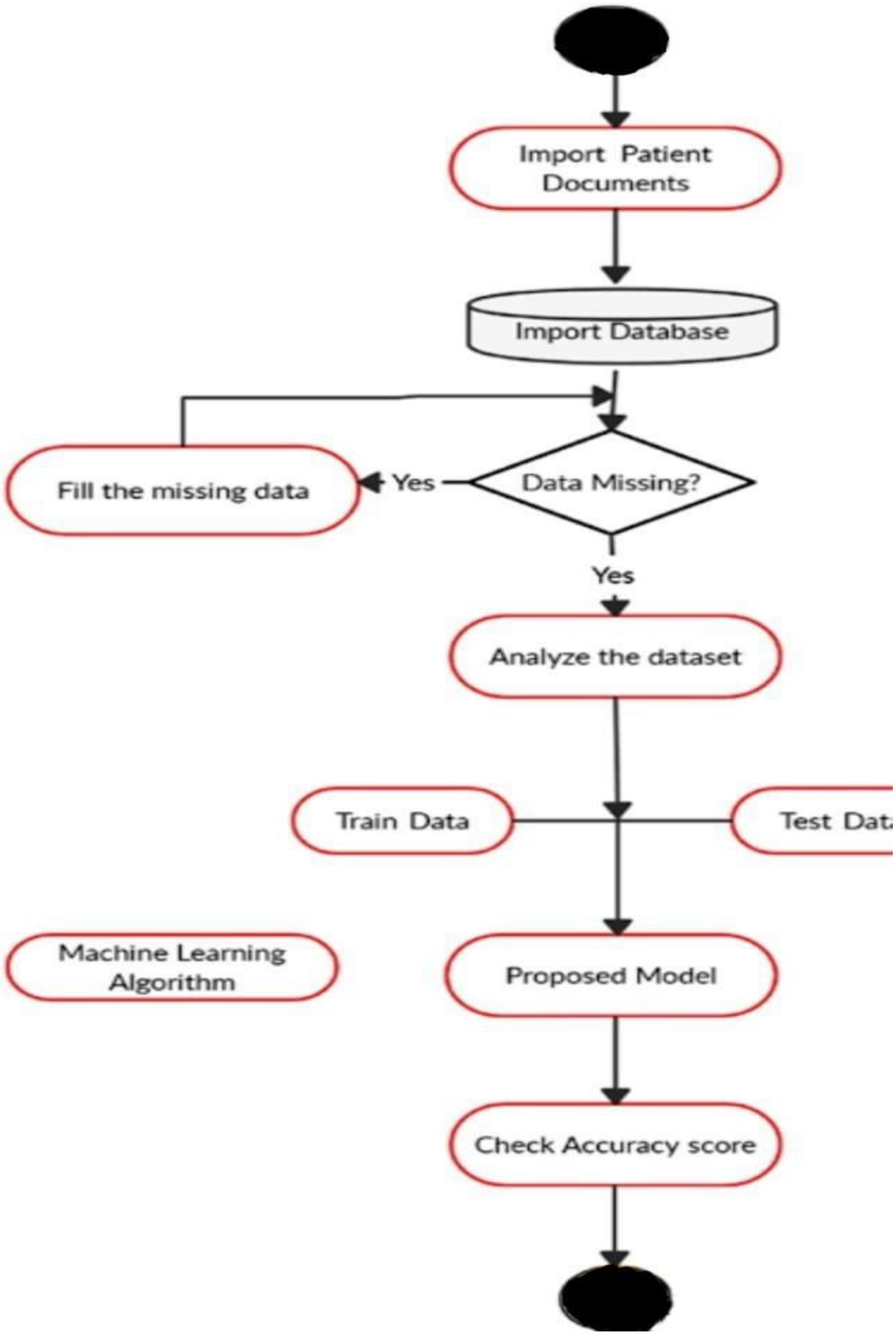

GANTT CHART
Task
0-15 Days
15-30 Days
30-45 Days
45-60 Days
Planning
Research
Learning
Implementation
Finishing Up
Testing & Quality Assurance
Documentation

2. DIAGRAMS

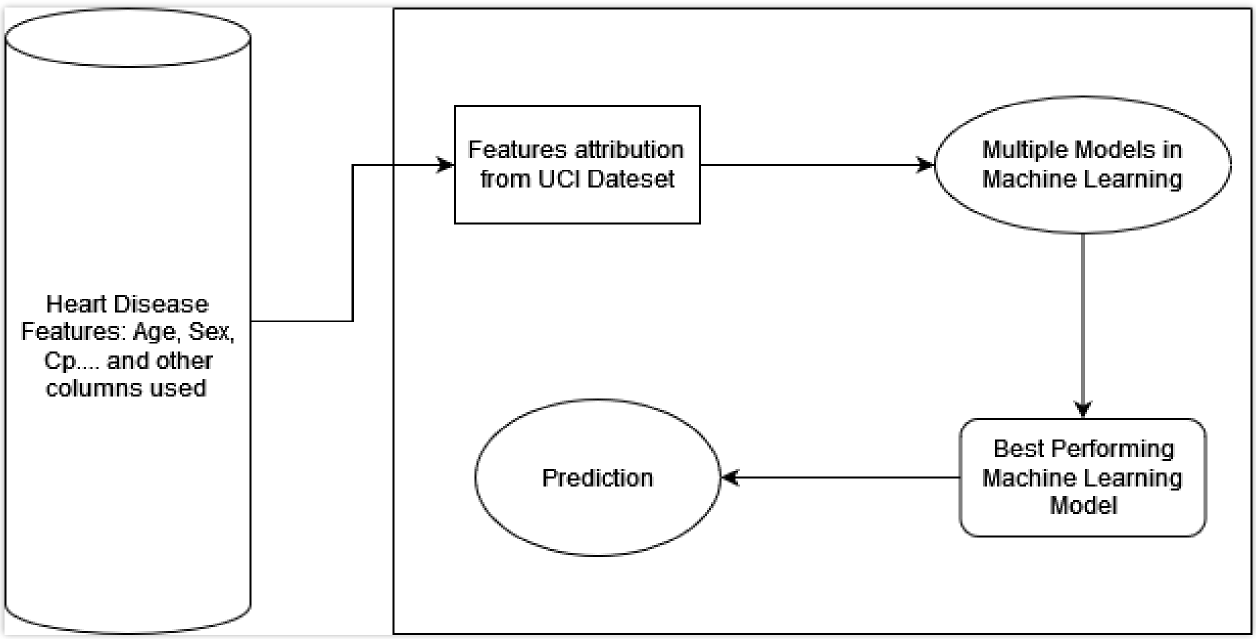
- **Use case diagram**
-



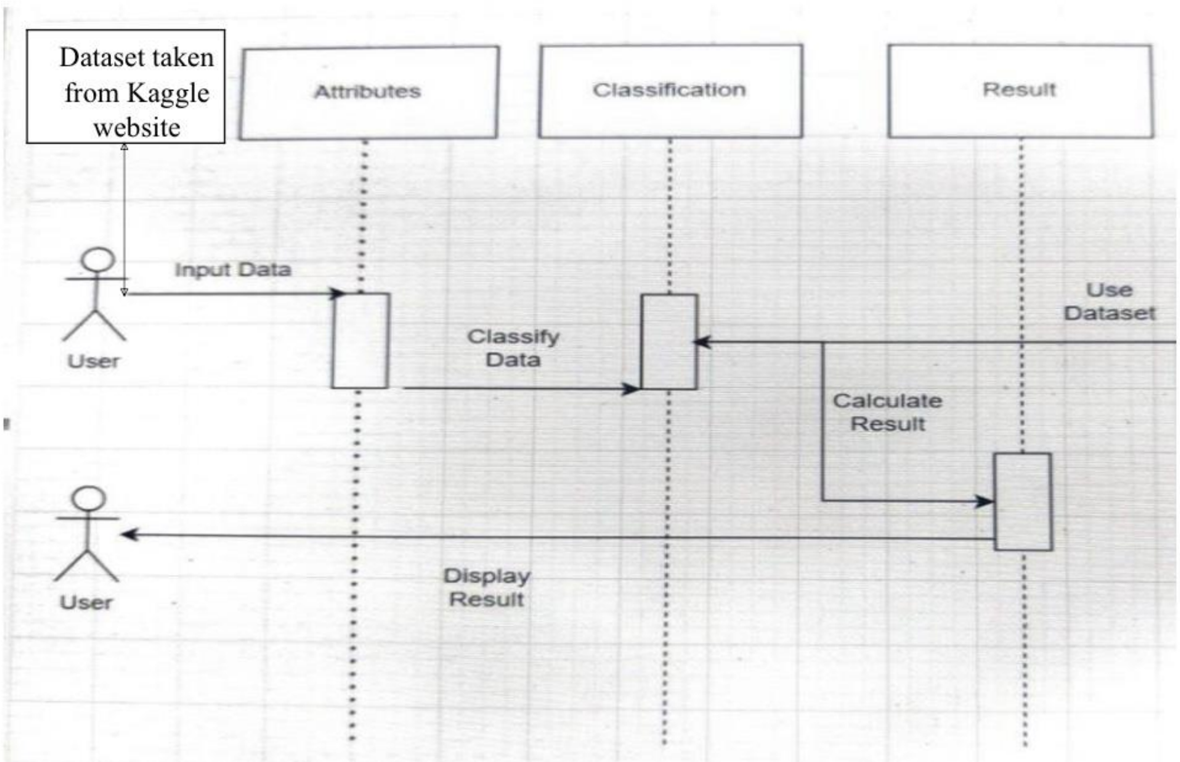
- **Activity DIAGRAM**



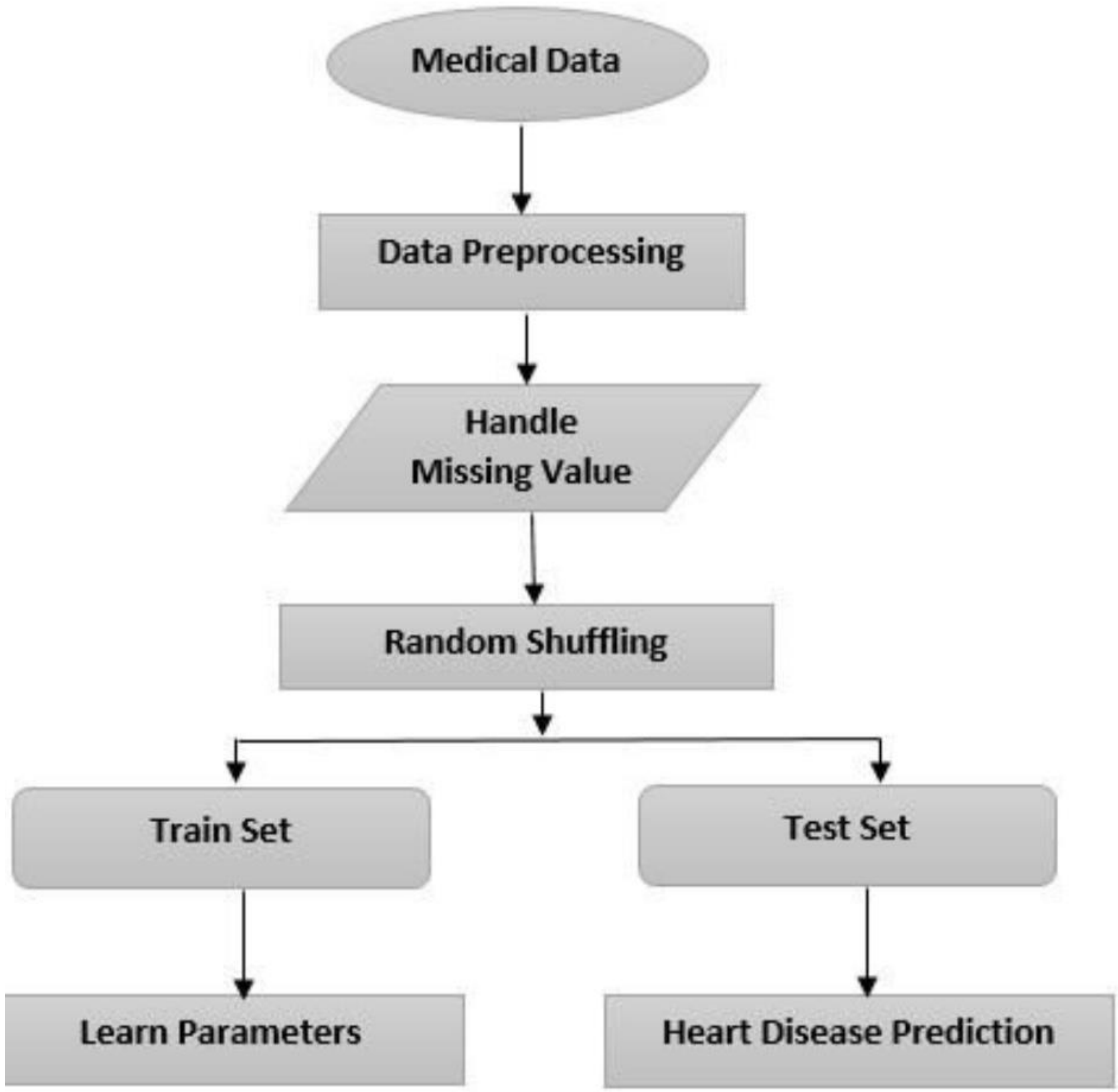
- **<u>Workflow diagram</u>**



- **SEQUENCE DIAGRAM**

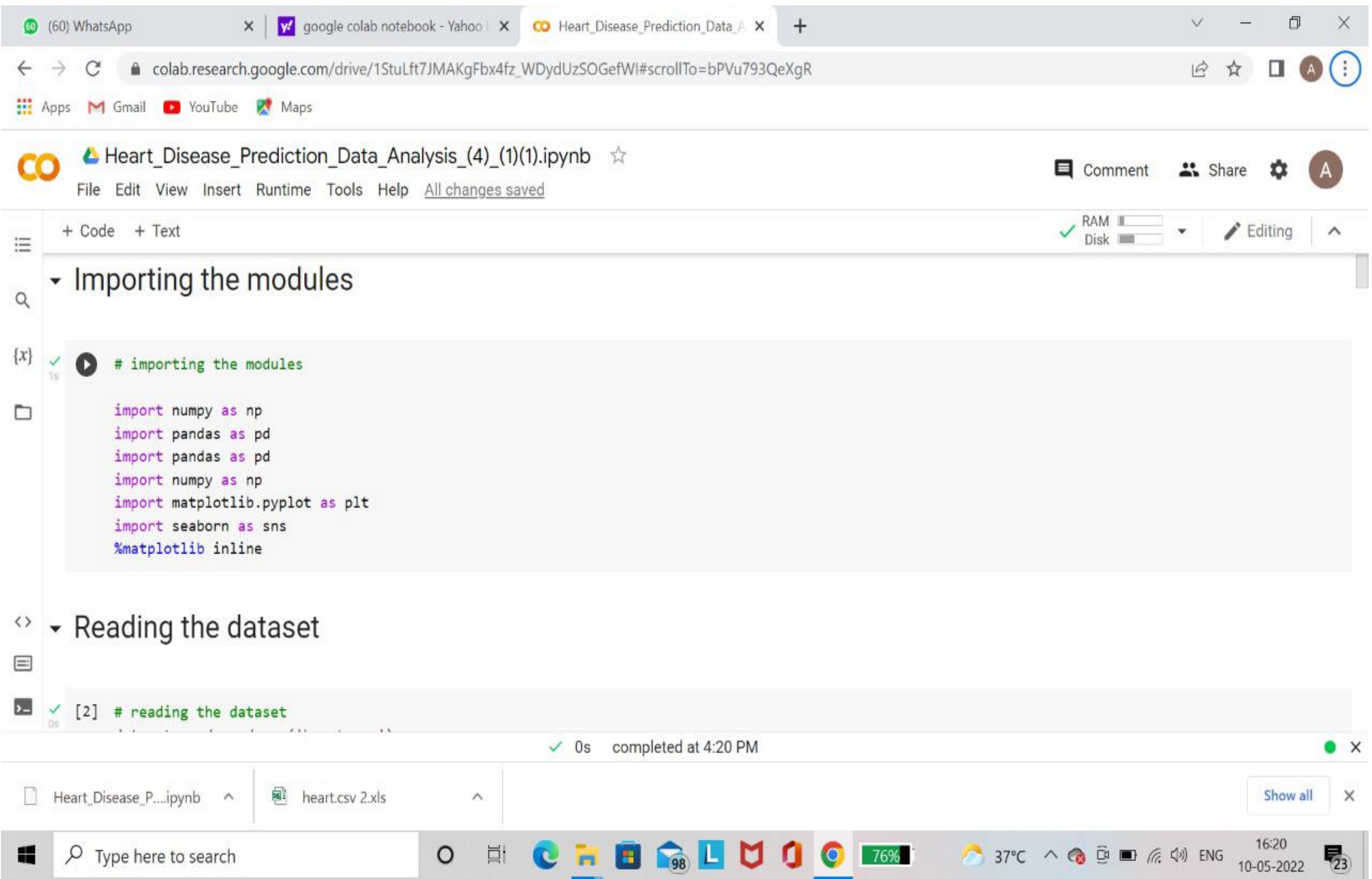


- **<u>STATE DIAGRAM</u>**

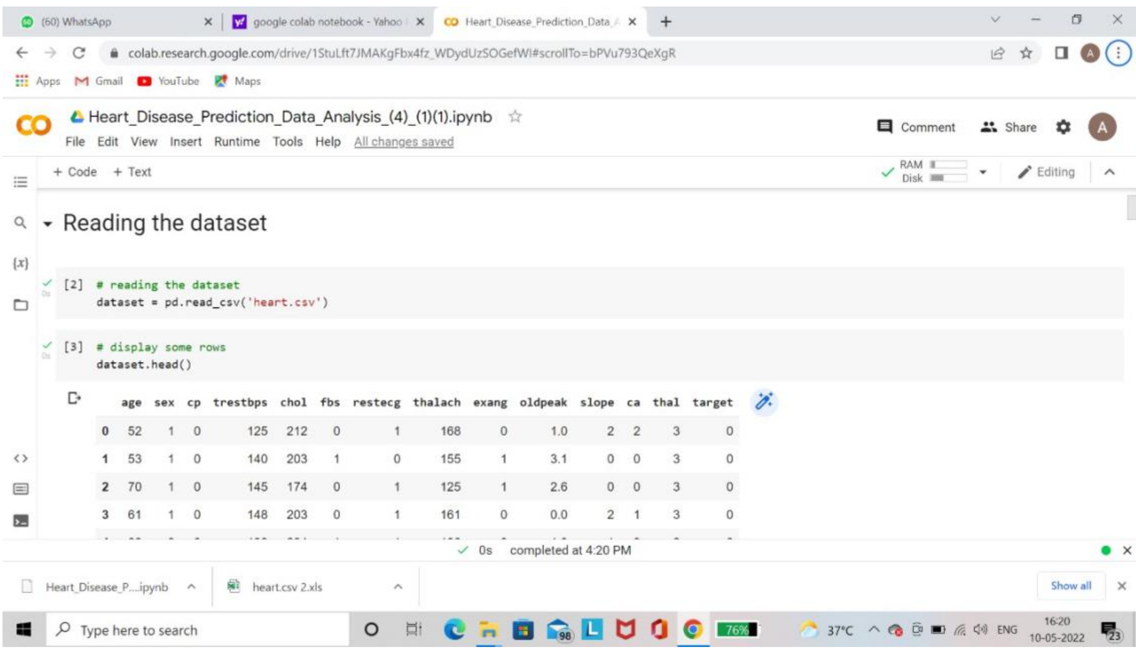


3. PROJECT SCREENSHOTS

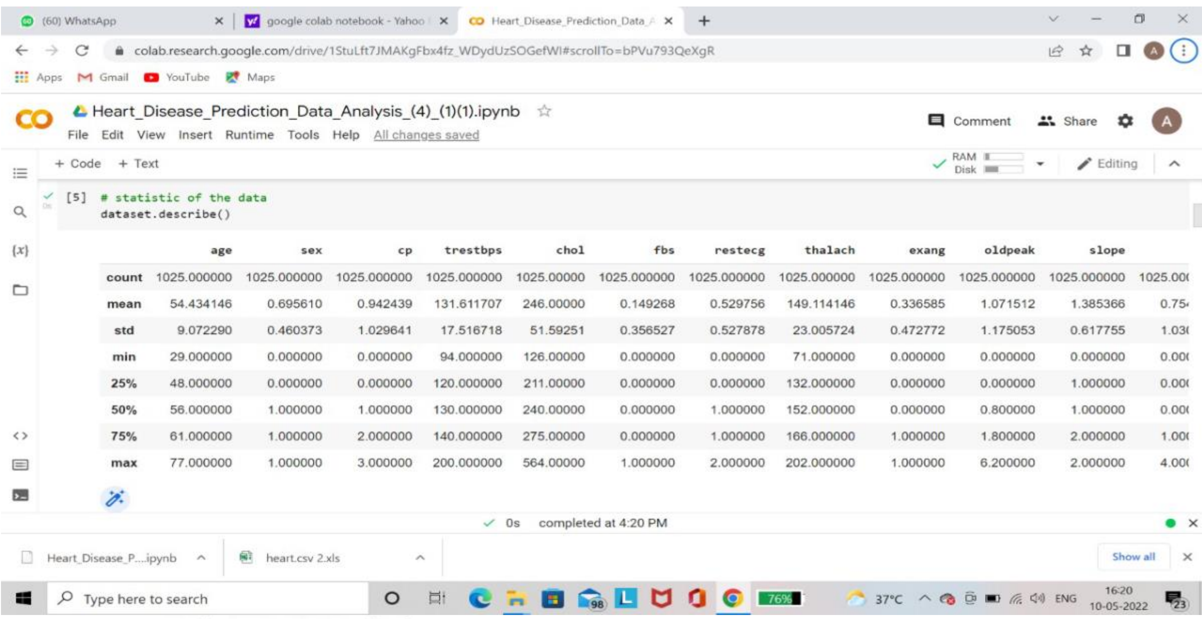
Importing the modules



<u>*Reading dataset through csv file*</u>

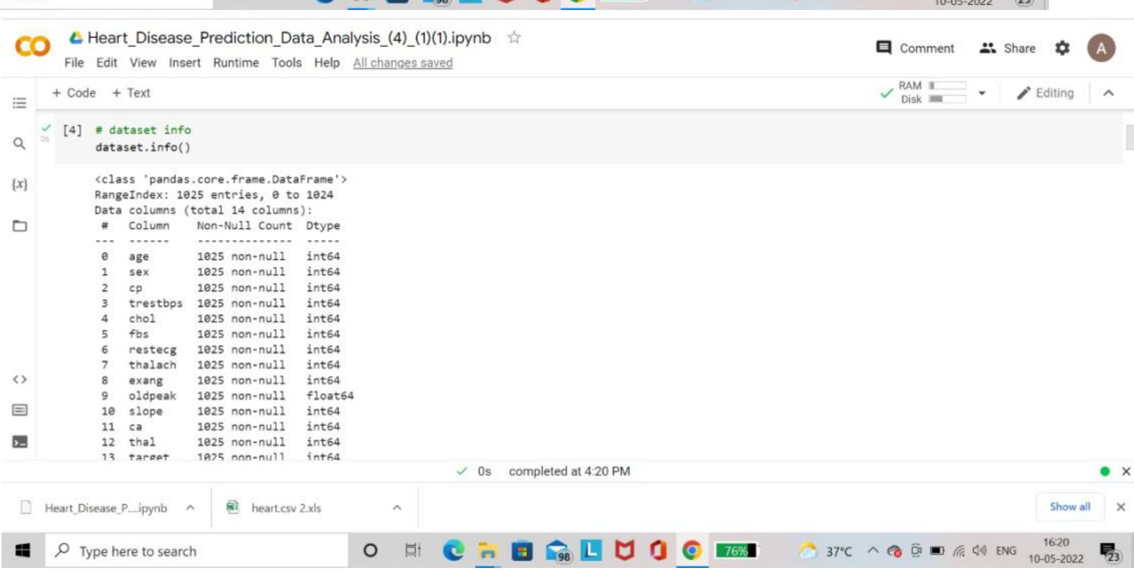


Dataset info is displayed shows null values or not



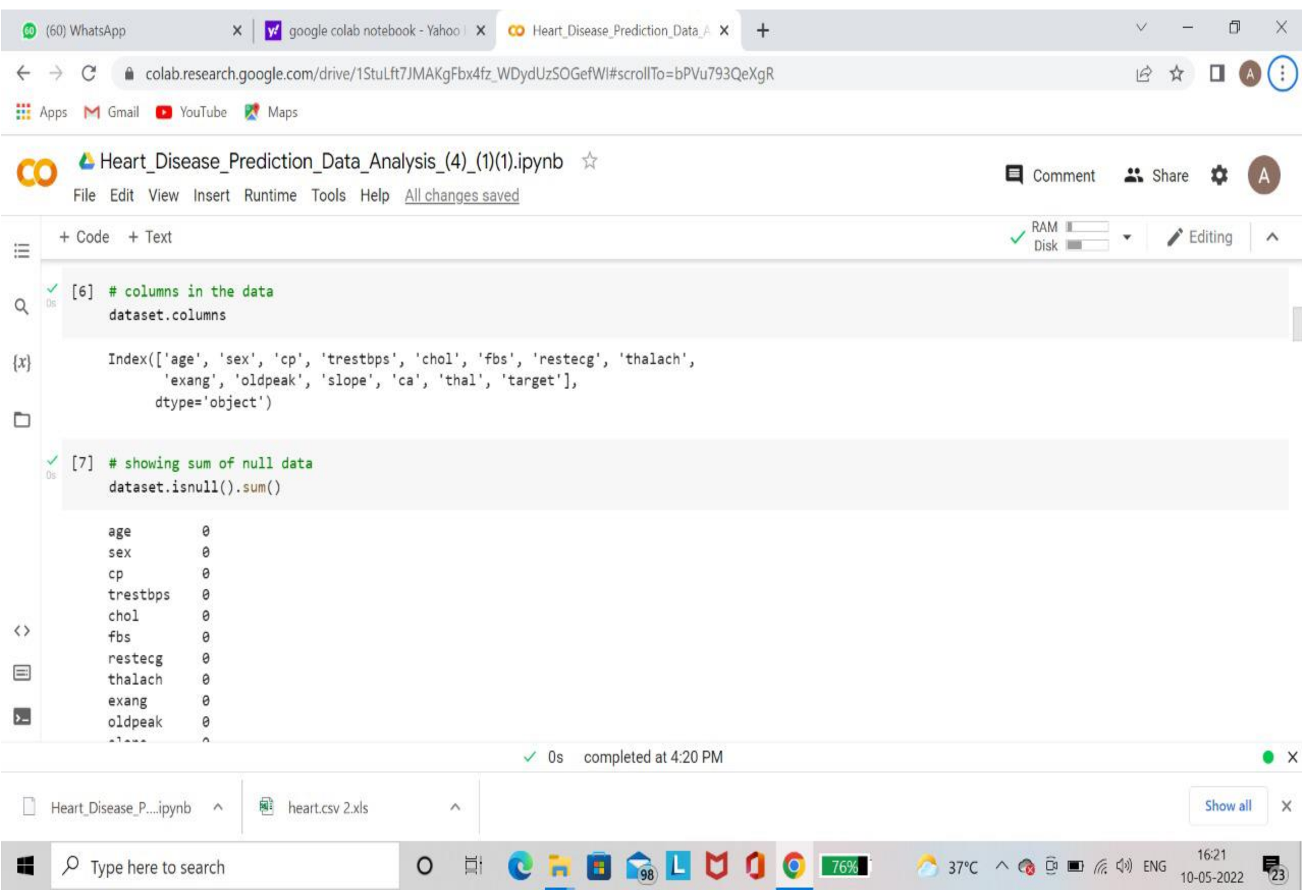
```
[5] # statistic of the data
dataset.describe()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	
count	1025.000000	1025.000000	1025.000000	1025.000000	1025.00000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.00
mean	54.434146	0.695610	0.942439	131.611707	246.00000	0.149268	0.529756	149.114146	0.336585	1.071512	1.385366	0.75
std	9.072290	0.460373	1.029641	17.516718	51.59251	0.356527	0.527878	23.005724	0.472772	1.175053	0.617755	1.03
min	29.000000	0.000000	0.000000	94.000000	126.00000	0.000000	0.000000	71.000000	0.000000	0.000000	0.000000	0.00
25%	48.000000	0.000000	0.000000	120.000000	211.00000	0.000000	0.000000	132.000000	0.000000	0.000000	1.000000	0.00
50%	56.000000	1.000000	1.000000	130.000000	240.00000	0.000000	1.000000	152.000000	0.000000	0.800000	1.000000	0.00
75%	61.000000	1.000000	2.000000	140.000000	275.00000	0.000000	1.000000	166.000000	1.000000	1.800000	2.000000	1.00
max	77.000000	1.000000	3.000000	200.000000	564.00000	1.000000	2.000000	202.000000	1.000000	6.200000	2.000000	4.00

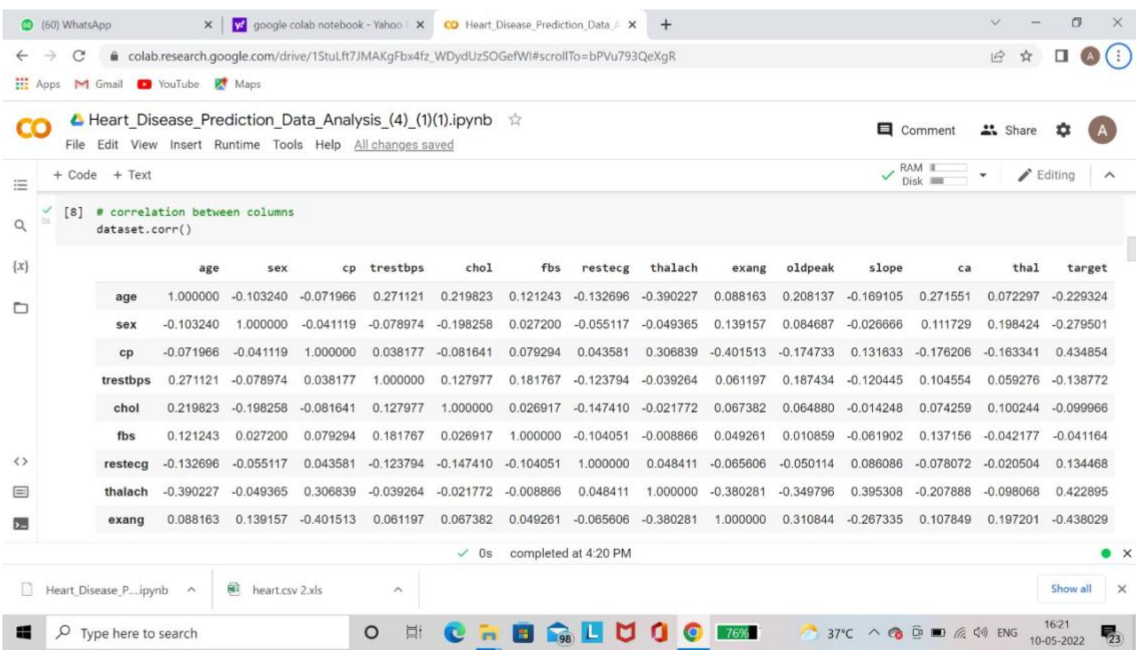


```
[4] # dataset info
dataset.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1025 entries, 0 to 1024
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       1025 non-null   int64
 1   sex       1025 non-null   int64
 2   cp        1025 non-null   int64
 3   trestbps  1025 non-null   int64
 4   chol      1025 non-null   int64
 5   fbs       1025 non-null   int64
 6   restecg   1025 non-null   int64
 7   thalach   1025 non-null   int64
 8   exang     1025 non-null   int64
 9   oldpeak   1025 non-null   float64
 10  slope     1025 non-null   int64
 11  ca        1025 non-null   int64
 12  thal      1025 non-null   int64
 13  target    1025 non-null   int64
```


Heart_Disease_Prediction_Data_Analysis_(4)_(1)(1).ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
[6] # columns in the data
dataset.columns
Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',
'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'],
dtype='object')
[7] # showing sum of null data
dataset.isnull().sum()
age 0
sex 0
cp 0
trestbps 0
chol 0
fbs 0
restecg 0
thalach 0
exang 0
oldpeak 0
0s completed at 4:20 PM

The below code represents the count, mean, std, etc. of each column, for example, count is 303.000, mean is different for different column, and so on Correlation for matrix



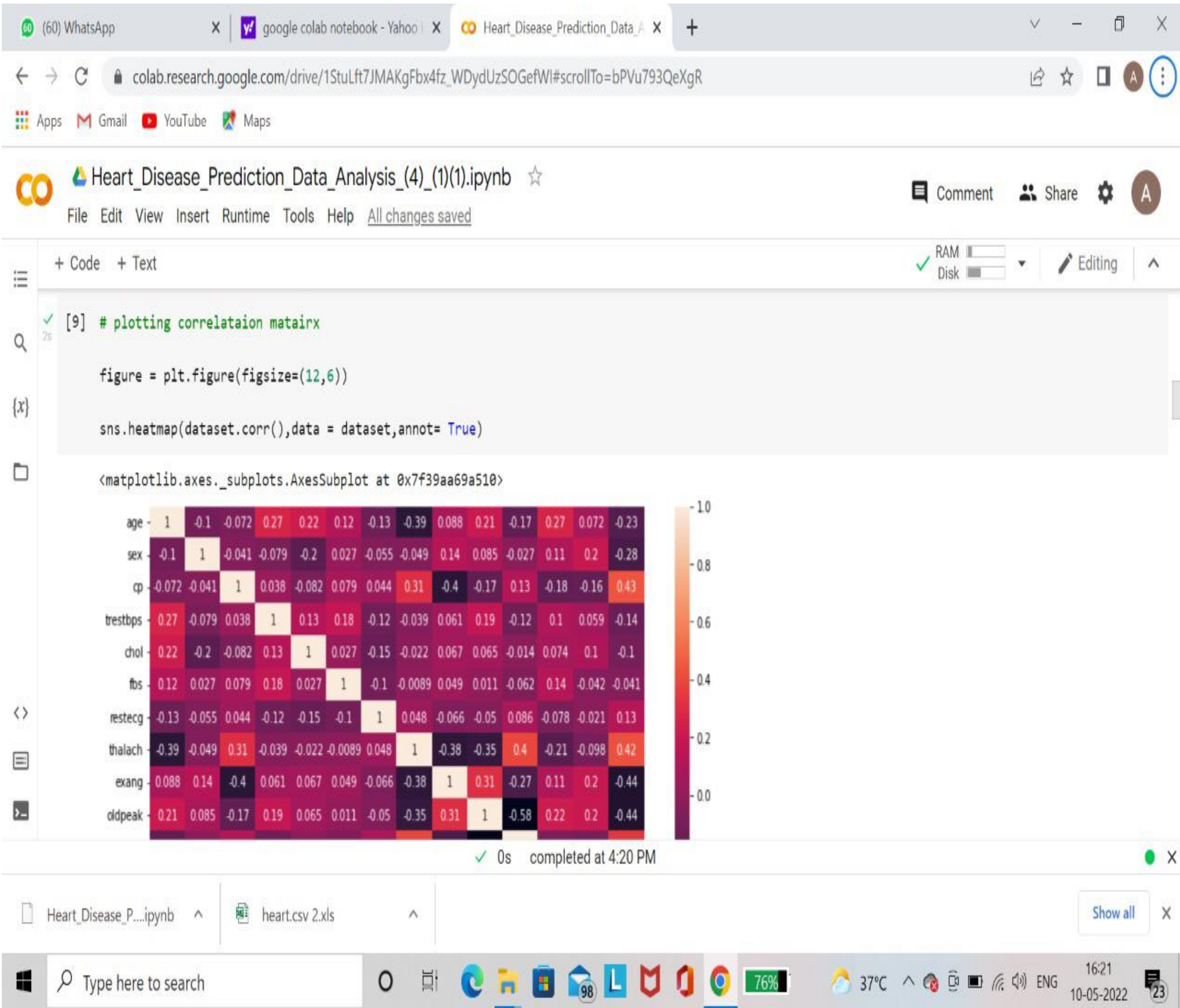
Heart_Disease_Prediction_Data_Analysis_(4)_(1)(1).ipynb

```
[8] # correlation between columns
dataset.corr()
```

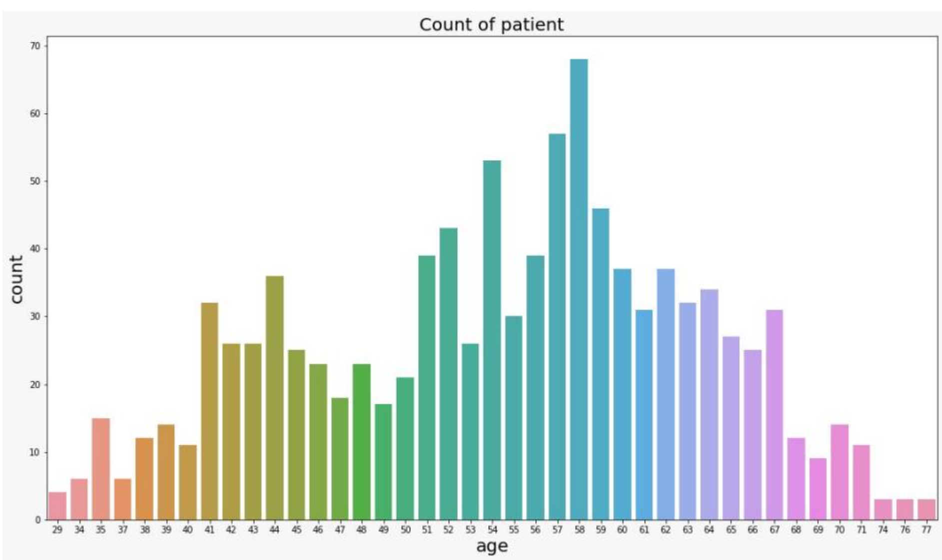
	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
age	1.000000	-0.103240	-0.071966	0.271121	0.219823	0.121243	-0.132696	-0.390227	0.088163	0.208137	-0.169105	0.271551	0.072297	-0.229324
sex	-0.103240	1.000000	-0.041119	-0.078974	-0.198258	0.027200	-0.055117	-0.049365	0.139157	0.084687	-0.026666	0.111729	0.198424	-0.279501
cp	-0.071966	-0.041119	1.000000	0.038177	-0.081641	0.079294	0.043581	0.306839	-0.401513	-0.174733	0.131633	-0.176206	-0.163341	0.434854
trestbps	0.271121	-0.078974	0.038177	1.000000	0.127977	0.181767	-0.123794	-0.039264	0.061197	0.187434	-0.120445	0.104554	0.059276	-0.138772
chol	0.219823	-0.198258	-0.081641	0.127977	1.000000	0.026917	-0.147410	-0.021772	0.067382	0.064880	-0.014248	0.074259	0.100244	-0.099966
fbs	0.121243	0.027200	0.079294	0.181767	0.026917	1.000000	-0.104051	-0.008866	0.049261	0.010859	-0.061902	0.137156	-0.042177	-0.041164
restecg	-0.132696	-0.055117	0.043581	-0.123794	-0.147410	-0.104051	1.000000	0.048411	-0.065606	-0.050114	0.086086	-0.078072	-0.020504	0.134468
thalach	-0.390227	-0.049365	0.306839	-0.039264	-0.021772	-0.008866	0.048411	1.000000	-0.380281	-0.349796	0.395308	-0.207888	-0.098068	0.422895
exang	0.088163	0.139157	-0.401513	0.061197	0.067382	0.049261	-0.065606	-0.380281	1.000000	0.310844	-0.267335	0.107849	0.197201	-0.438029

✓ 0s completed at 4:20 PM

Heat map


(60) WhatsApp
google colab notebook - Yahoo
Heart_Disease_Prediction_Data_A
colab.research.google.com/drive/1StuLft7JMAKgFbx4fz_WDydUzSOGefWI#scrollTo=bPVu793QeXgR
Apps
Gmail
YouTube
Maps
Heart_Disease_Prediction_Data_Analysis_(4)_(1)(1).ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
[9] # plotting correlataion matairx
figure = plt.figure(figsize=(12,6))
sns.heatmap(dataset.corr(),data = dataset,annot= True)
<matplotlib.axes._subplots.AxesSubplot at 0x7f39aa69a510>
age
sex
cp
trestbps
chol
fbs
restecg
thalach
exang
oldpeak
1.0
0.8
0.6
0.4
0.2
0.0
0s completed at 4:20 PM
Heart_Disease_P....ipynb
heart.csv 2.xls
Show all
Type here to search
37°C
ENG
16:21
10-05-2022

some graphs have been developed to give deeper insights into the dataset and understand what the data contains.



This graph show distribution of the patient age and count age ranging from 50-60 mostly .

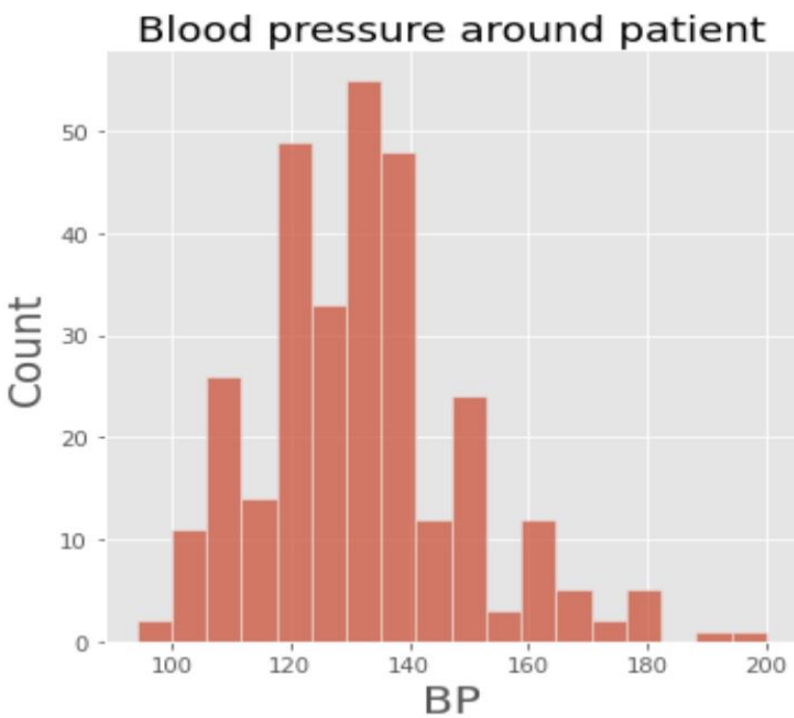
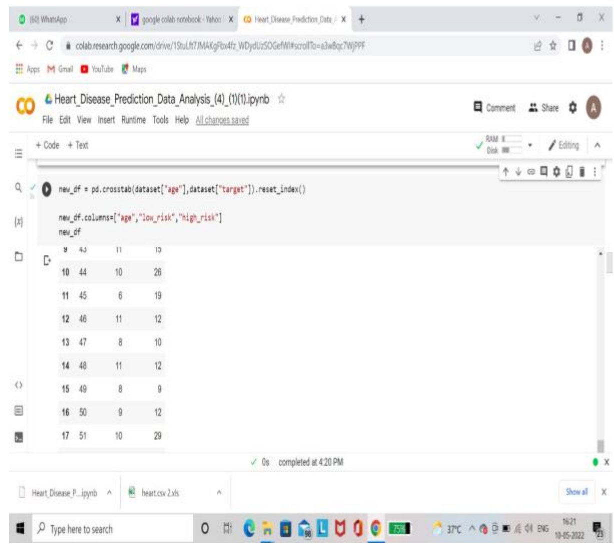


Figure 8: Count of Patient with respect to BP



Count of patients with respect to blood pressure data has huge number of patients count from 50-60

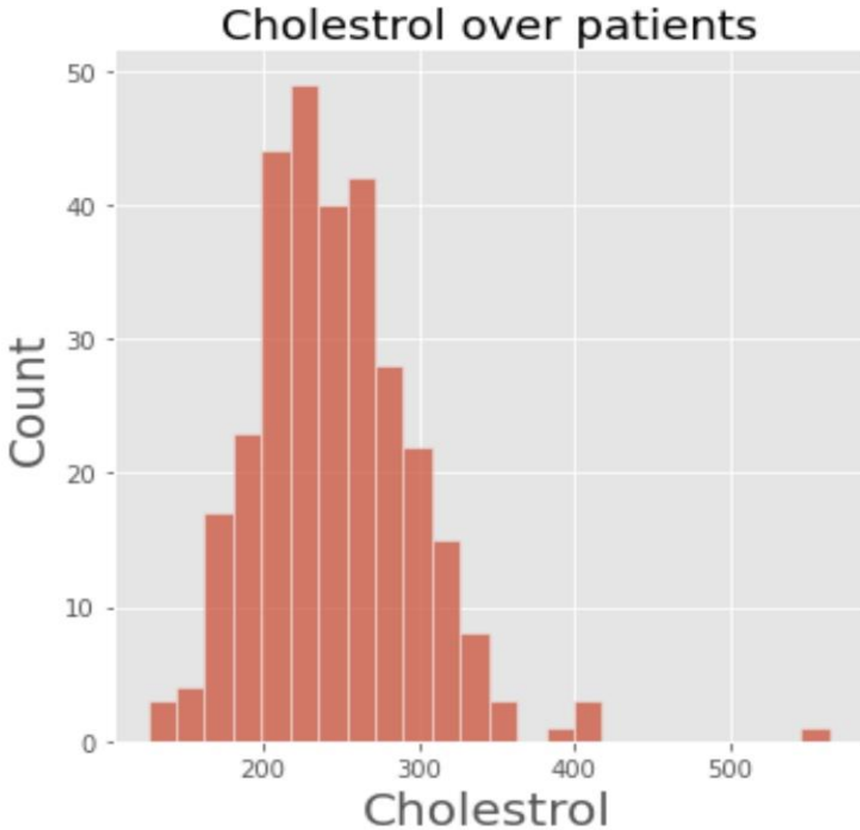


Figure 9: Count of Patient with respect to Cholesterol

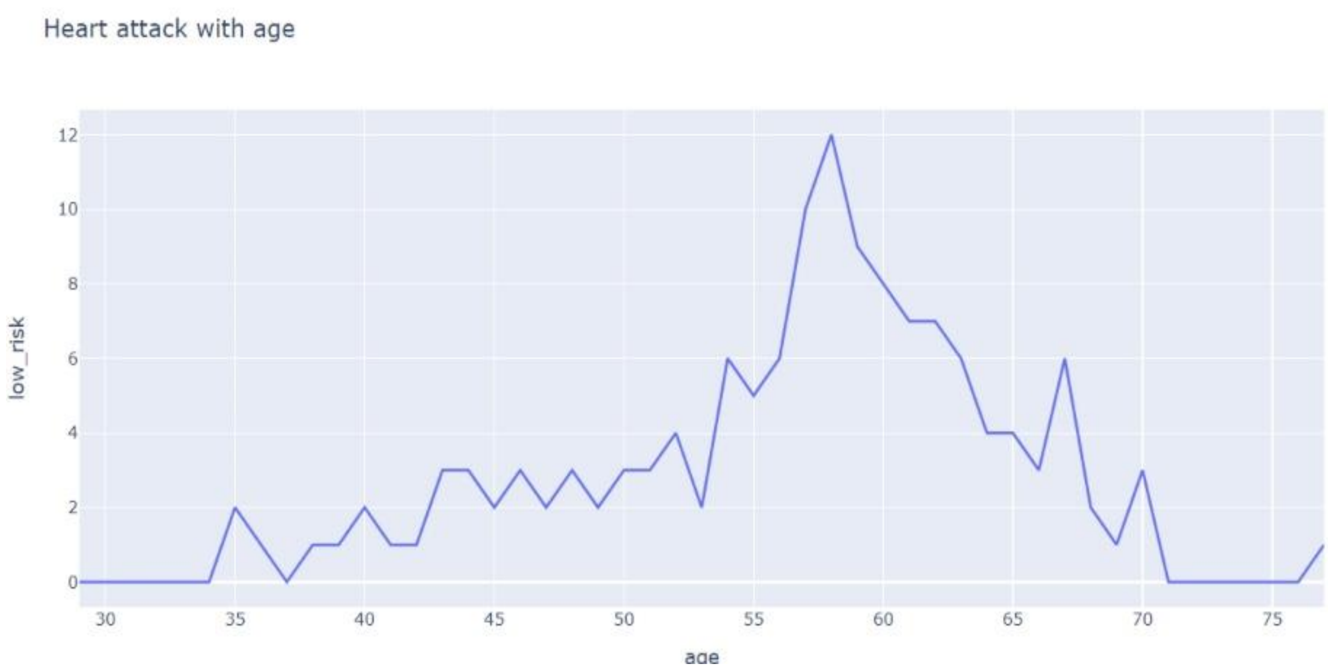
Count of patient with cholesterol data has huge count from 200-400

S. No.	age	low_risk	high_risk
0	29	0	1
1	34	0	2
2	35	2	2
3	37	0	2
4	38	1	2
5	39	1	3
6	40	2	1
7	41	1	9
8	42	1	7
9	43	3	5
10	44	3	8
11	45	2	6
12	46	3	4
13	47	2	3
14	48	3	4
15	49	2	3
16	50	3	4
17	51	3	9
18	52	4	9
19	53	2	6
20	54	6	10
21	55	5	3
22	56	6	5
23	57	10	7
24	58	12	7
25	59	9	5
26	60	8	3
27	61	7	1
28	62	7	4
29	63	6	3
30	64	4	6
31	65	4	4
32	66	3	4
33	67	6	3
34	68	2	2
35	69	1	2
36	70	3	1
37	71	0	3
38	74	0	1
39	76	0	1
40	77	1	0

The table above above shows that if the patient is high risk or low risk based on the age

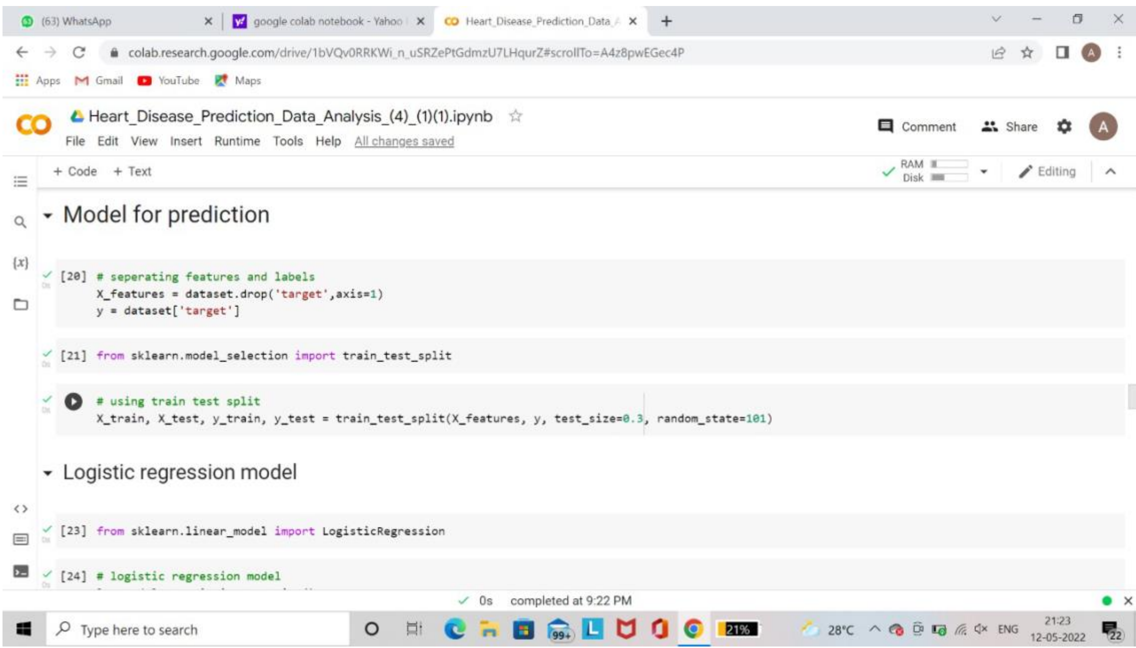
Risk table

The relation of age and high risk is represented in the graph above, this is done to find the relation between heart attack and age. The graph describes the high risk of heart attack over the age, the patient age between 40 to 55 has the high chance of heart attack. As the count of patient had heart attack is more

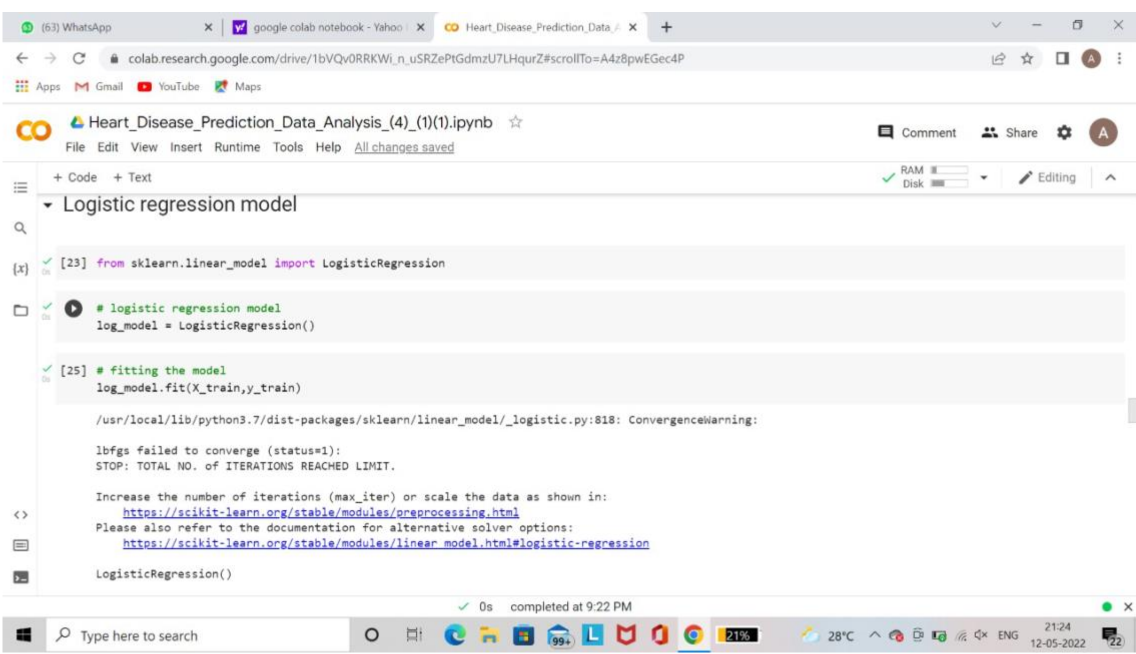


The graph describes the low risk of heart attack over the age, the patient age between 55 to 65 has the low risk of heart attack.

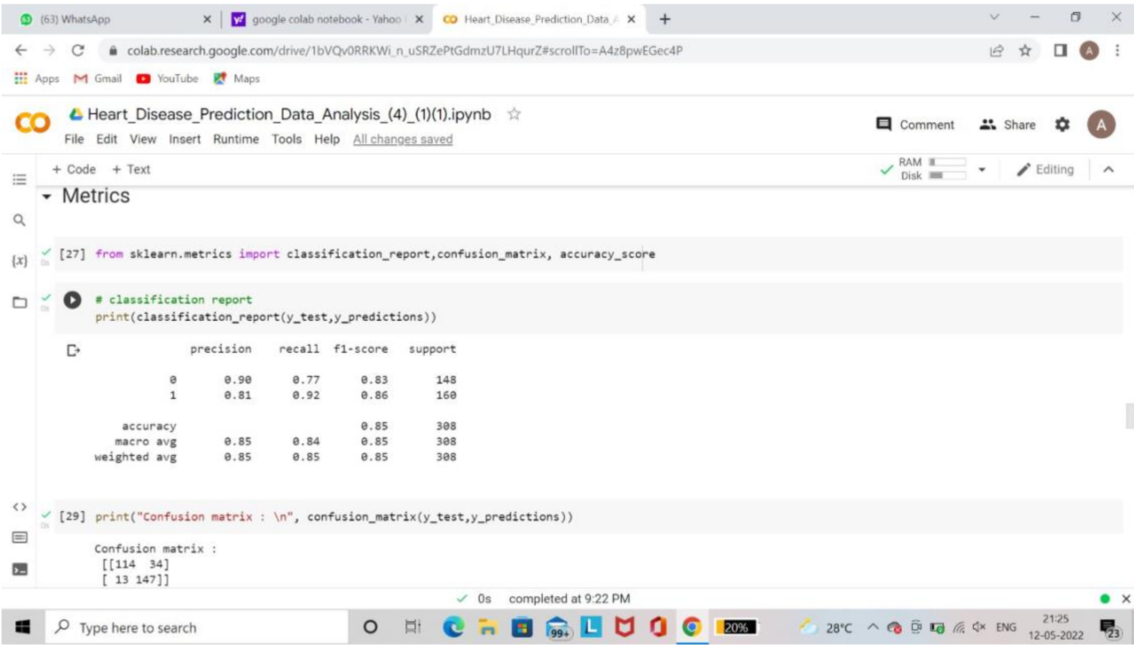
Model prediction



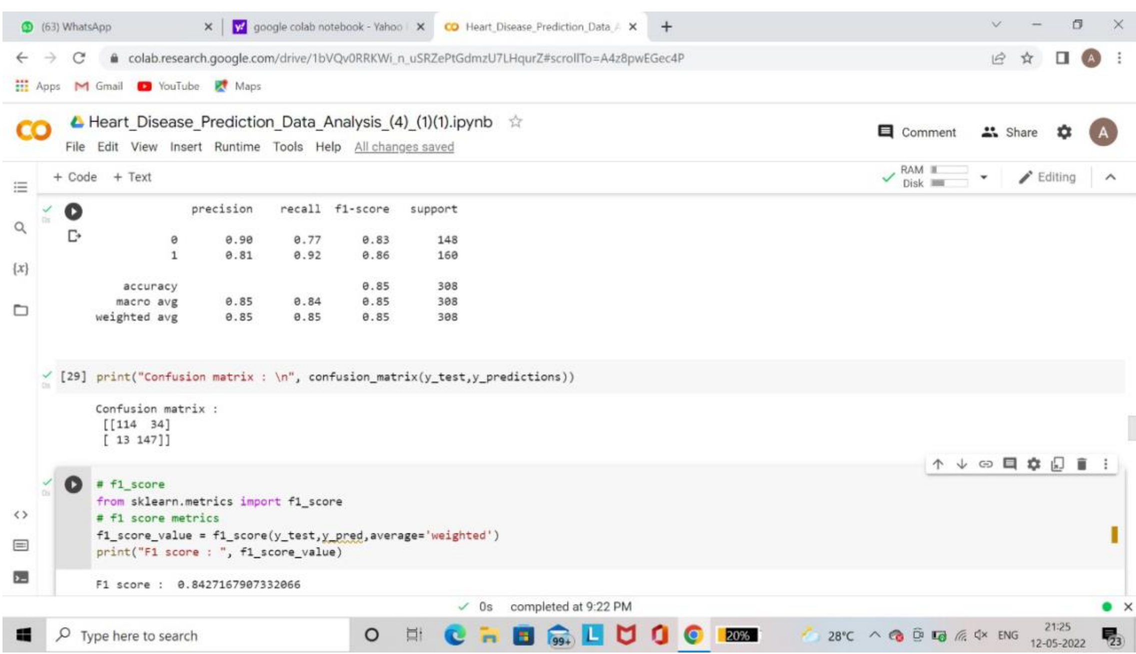
Fitting logistic regression model



Metrics for logistic regression

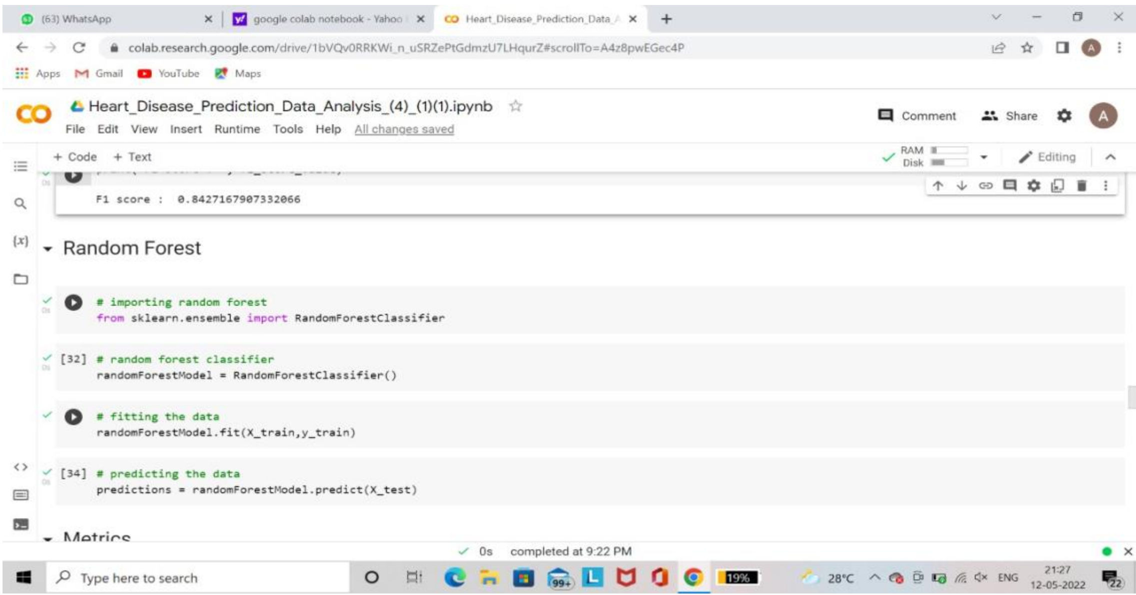


F1 score metrics for logistic regression

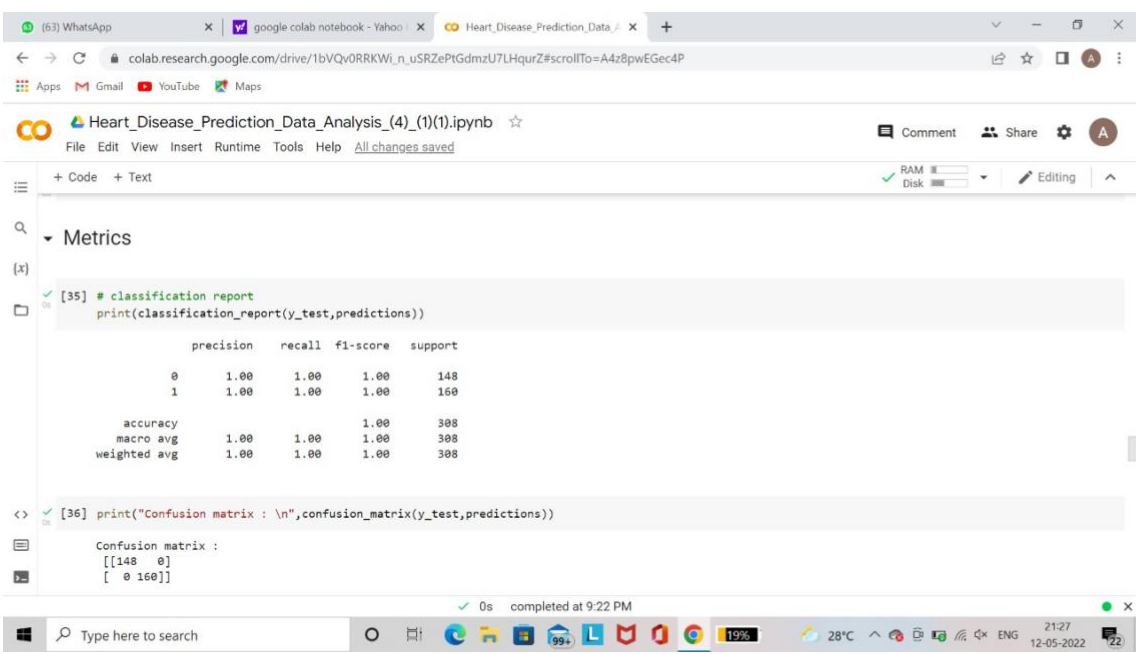


Random forest

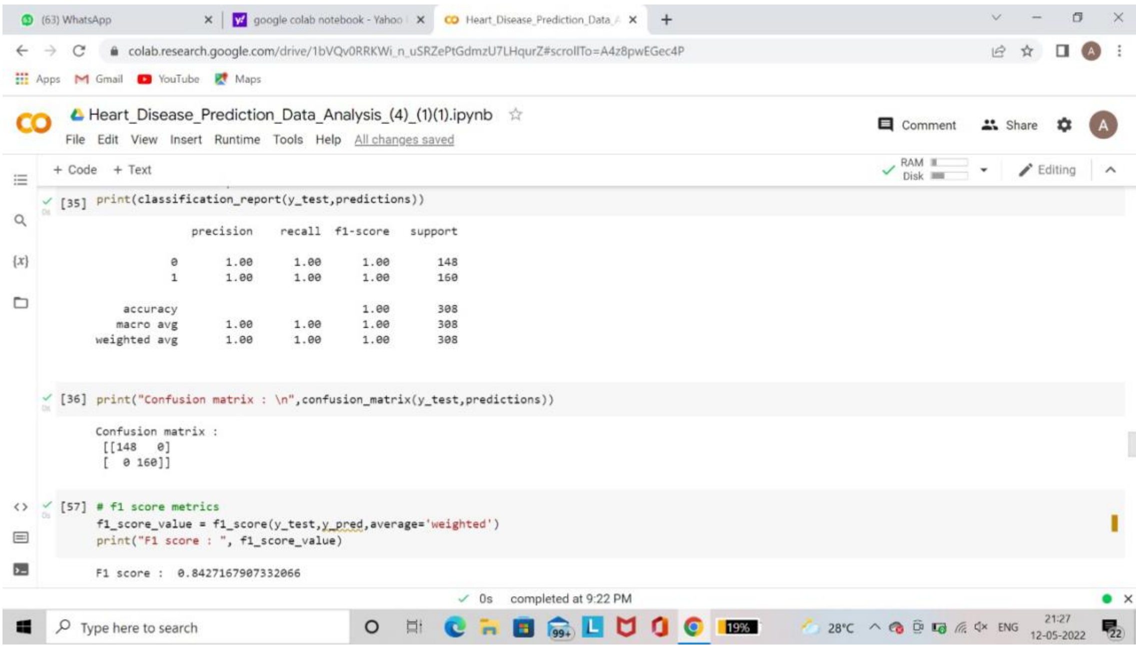
Importing random forest ,fitting data .



Metrics

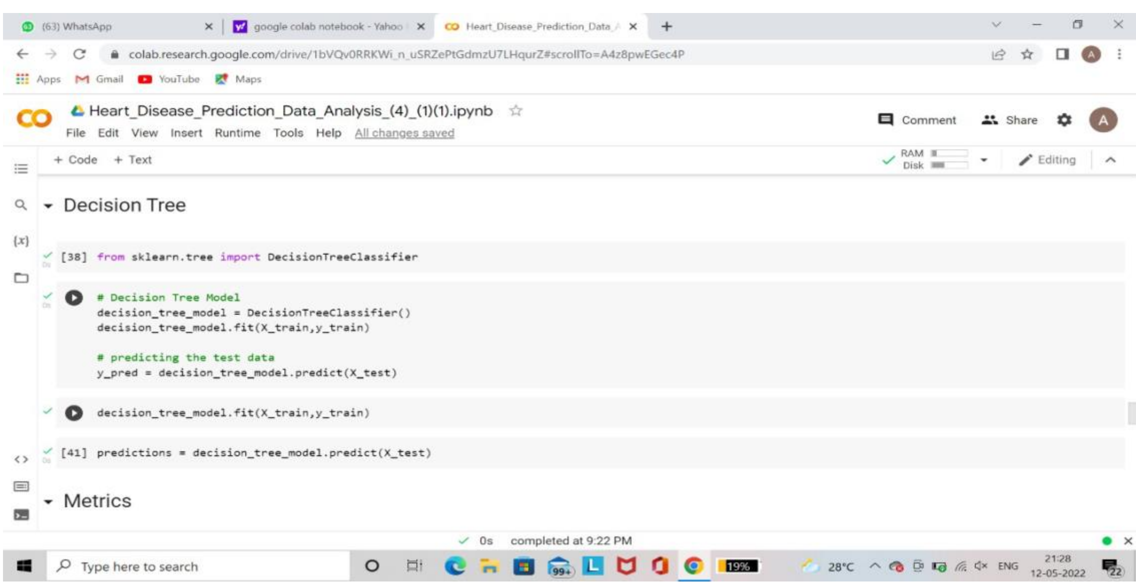


F1 score metrics for random forest

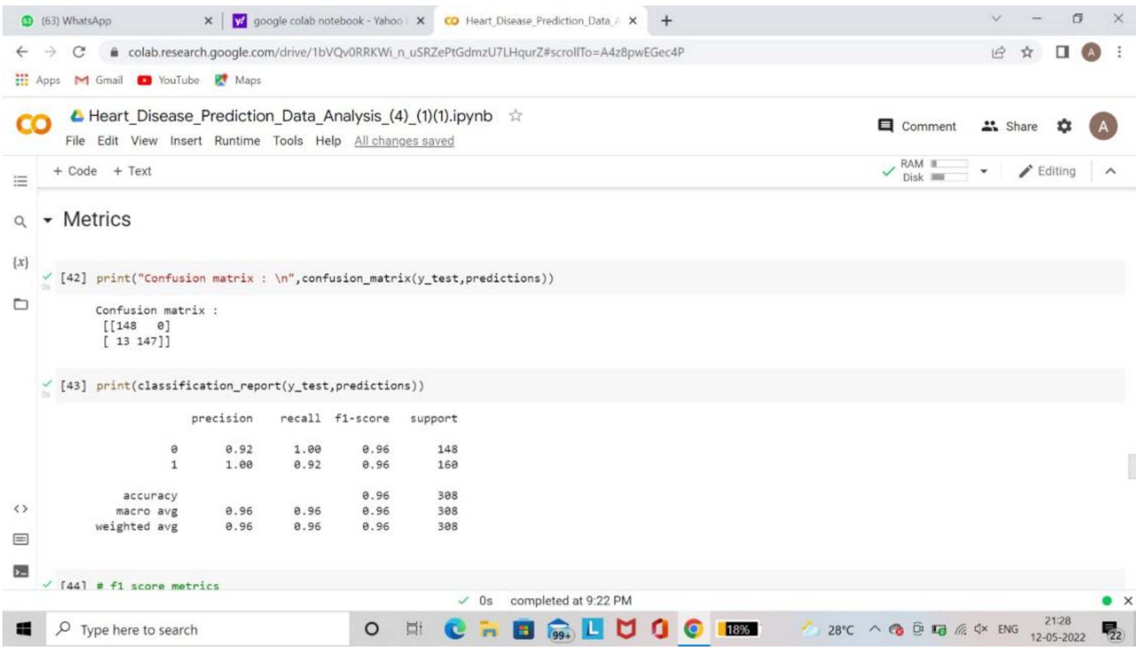


Decision tree

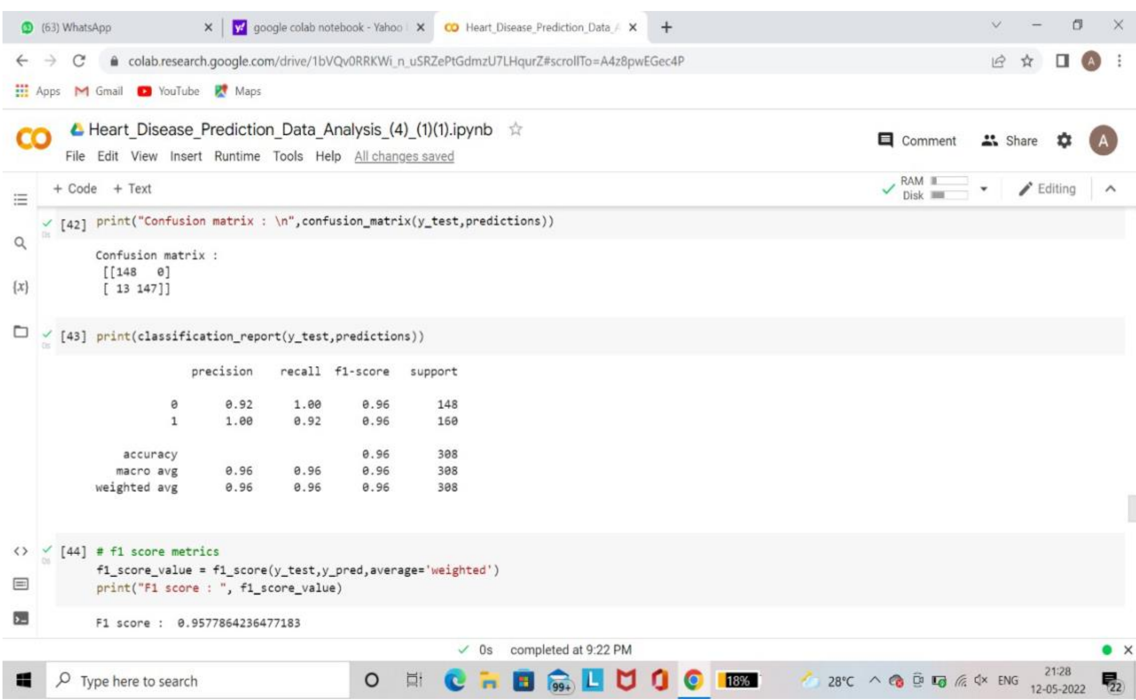
Importing decision tree



Metrics

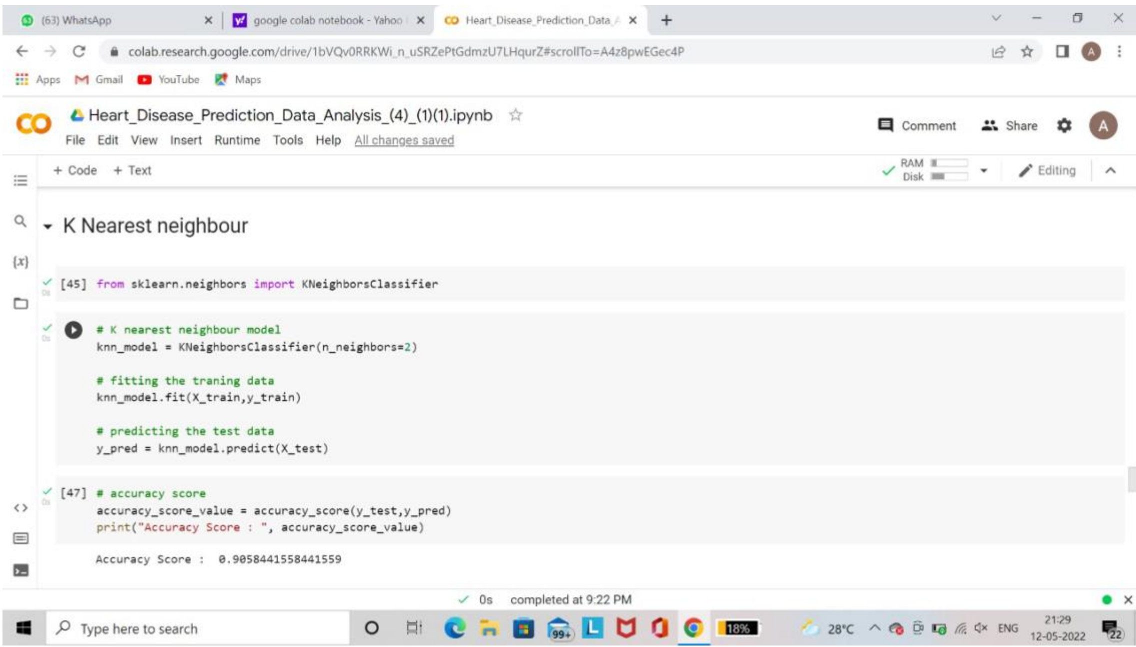


F1 score metrics

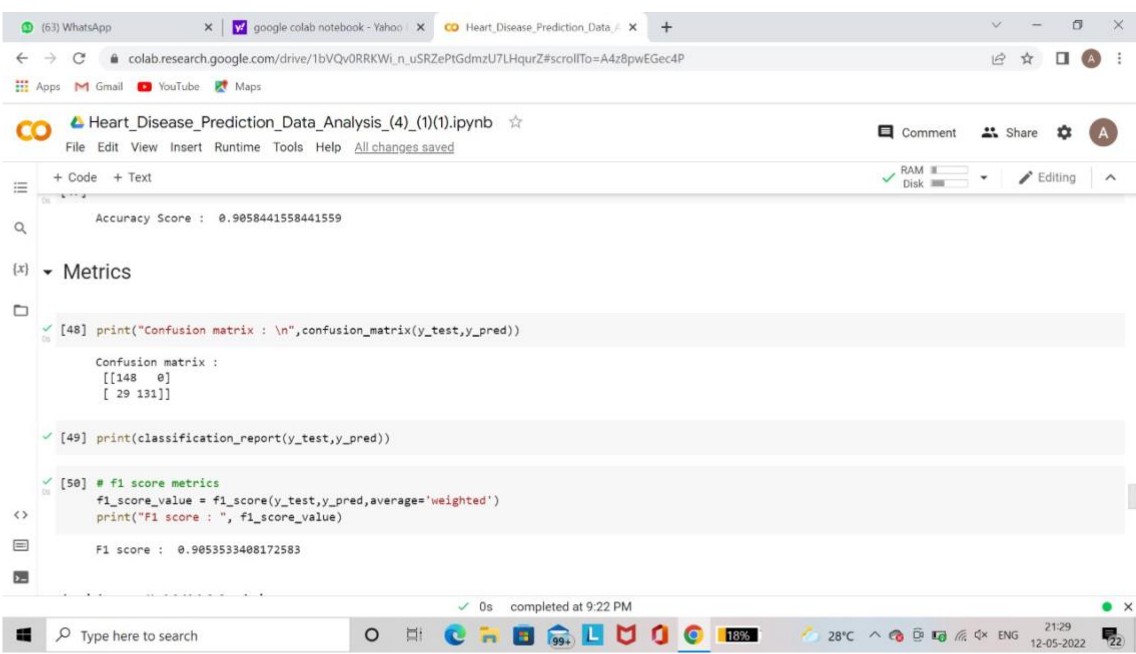


Knn

Importing knn

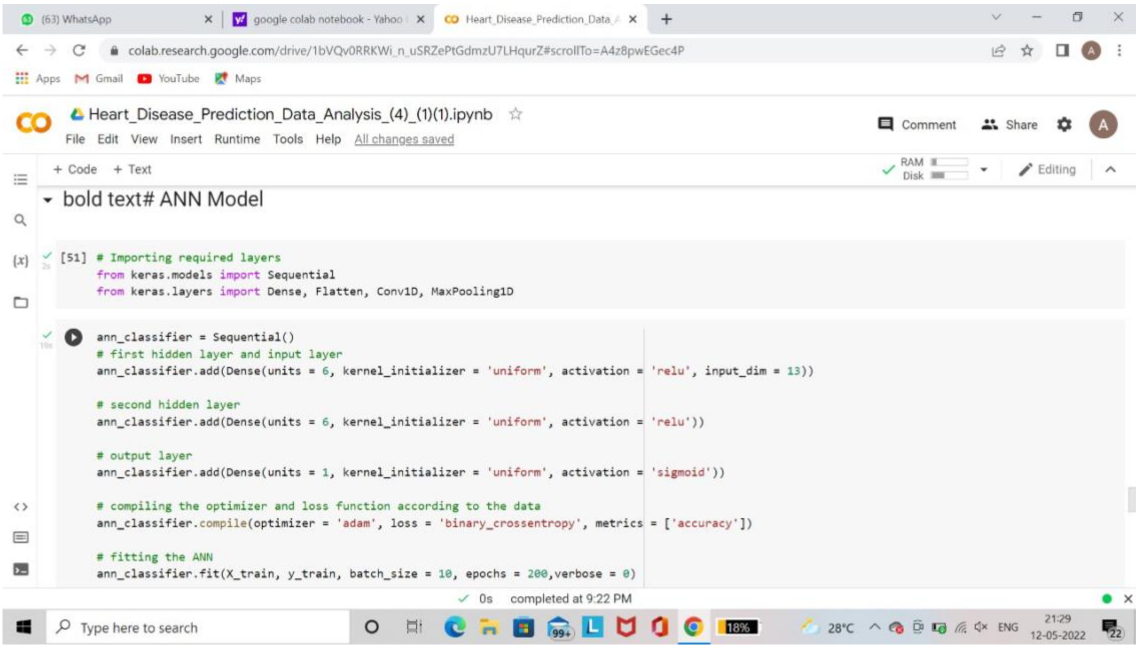


F1 score Metrics

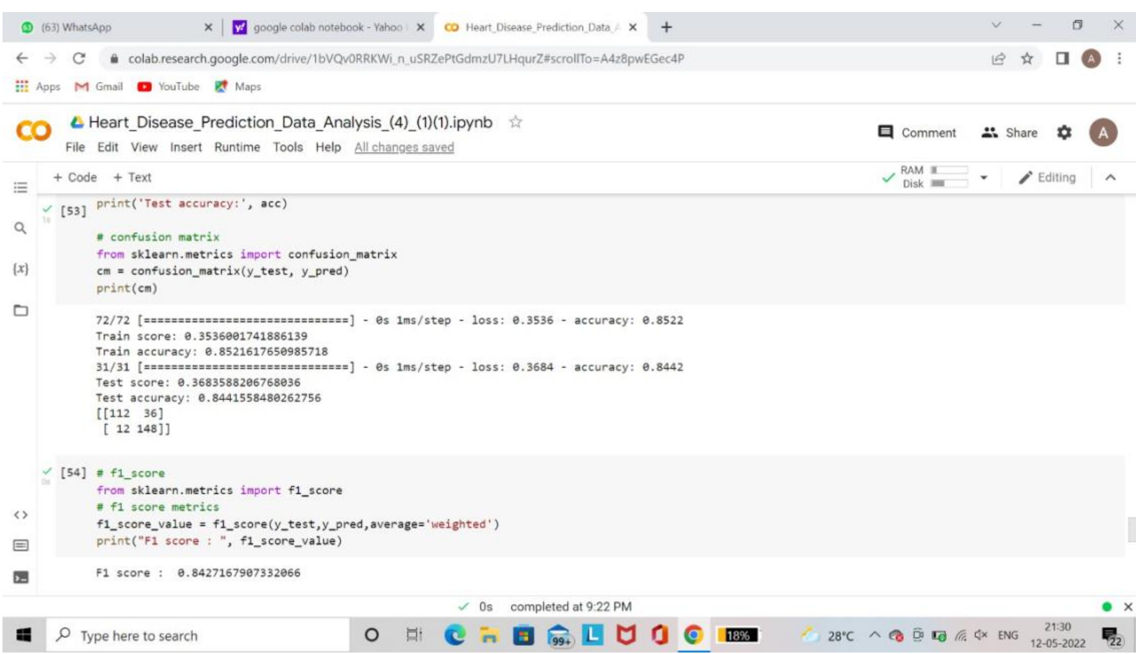


Ann

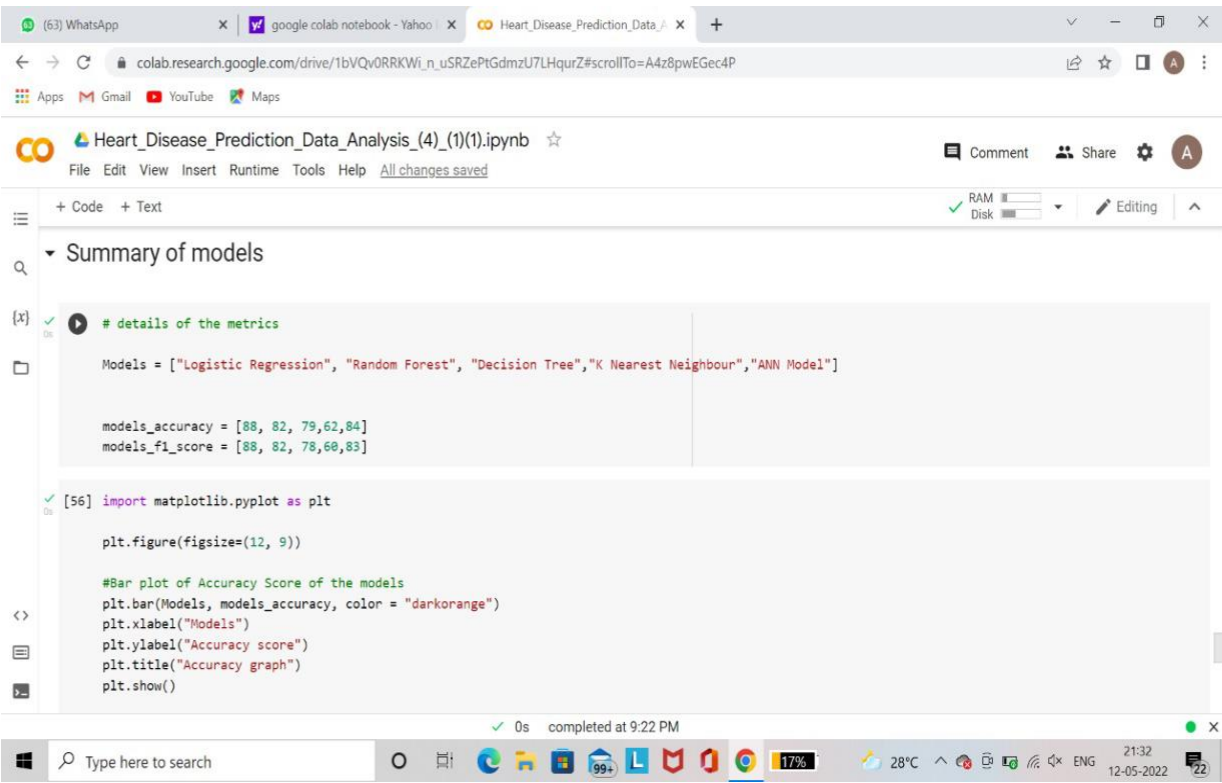
Importing Ann



F1 score metrics

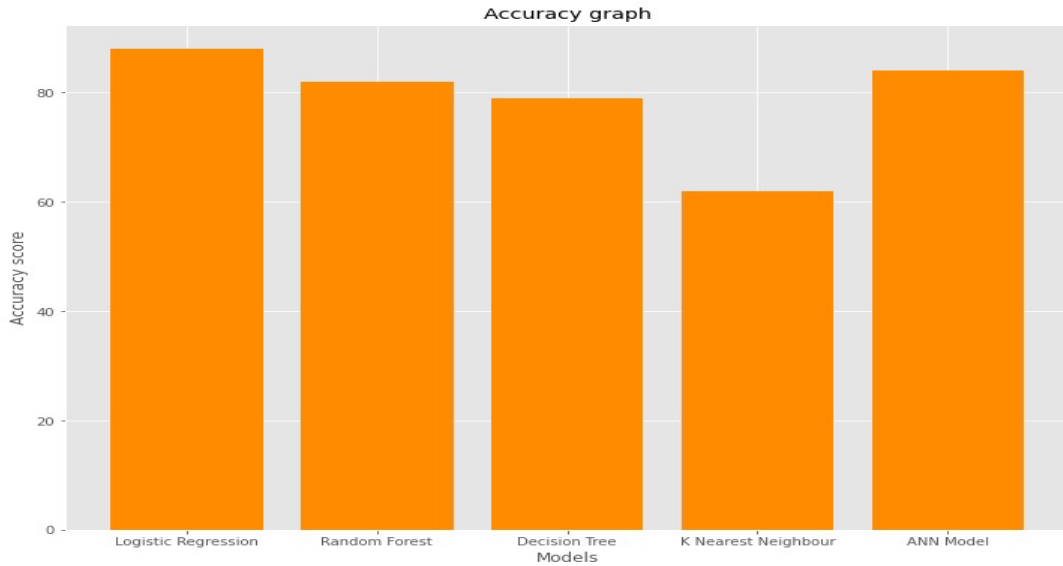


Summary of all models and graph plotting based on F1 score of each graph

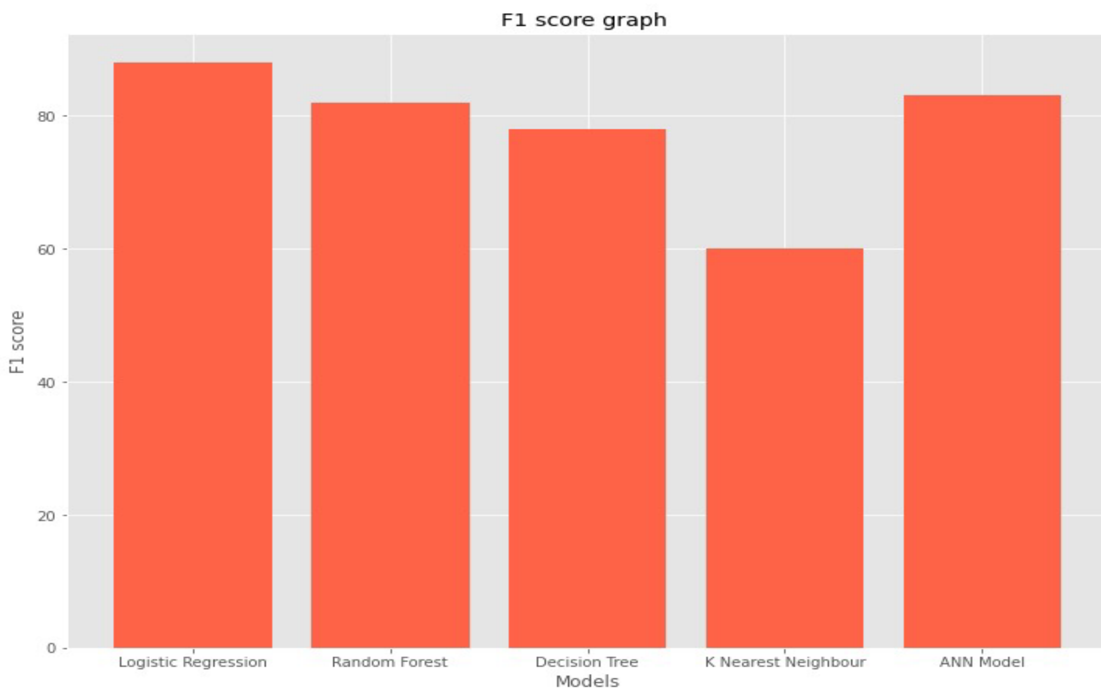


Plotting final graph based on all accuracy score of all algorithms used . Final graph consist accuracy and F1 score .

Accuracy graph



F1 score graph



4.TESTING

LOGISITC REGRESSION :-

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

[ ] # classification report
    print(classification_report(y_test,y_predictions))

                  precision    recall  f1-score   support

               0       0.92      0.82      0.87        44
               1       0.85      0.94      0.89        47

        accuracy                           0.88        91
       macro avg       0.88      0.88      0.88        91
    weighted avg       0.88      0.88      0.88        91

[ ] print("Confusion matrix : \n", confusion_matrix(y_test,y_predictions))

    Confusion matrix :
     [[36  8]
     [ 3 44]]

[ ] # f1_score
    from sklearn.metrics import f1_score
    # f1 score metrics
    f1_score_value = f1_score(y_test,y_pred,average='weighted')
    print("F1 score : ", f1_score_value)

    F1 score :  0.8780865876575448
```

Accuracy score – 0.88

Classification report:

Precision for class 0 – 0.92

Precision for class 1 – 0.85

Recall for class 0 – 0.82

Recall for class 1 – 0.94

F1 score – 0.87

In the output, there are 36 and 44 actual/correct predictions, with 8 and 4 wrong predictions, respectively.

Random forest

```
[ ]  # classification report
     print(classification_report(y_test,predictions))
```

```
              precision    recall  f1-score   support

           0       0.87      0.77      0.82        44
           1       0.81      0.89      0.85        47

    accuracy                           0.84        91
   macro avg       0.84      0.83      0.83        91
weighted avg       0.84      0.84      0.83        91
```

```
[ ]  print("Confusion matrix : \n",confusion_matrix(y_test,predictions))
```

```
Confusion matrix :
 [[34 10]
 [ 5 42]]
```

```
[ ]  # f1 score metrics
     f1_score_value = f1_score(y_test,y_pred,average='weighted')
     print("F1 score : ", f1_score_value)
```

```
F1 score :  0.8780865876575448
```

Accuracy score – 0.84

Precision for class 0 – 0.87,Precision for class 1 – 0.81

Recall for class 0 – 0.77,Recall for class 1 – 0

F1 score – 0.87

In the output, there are 34 and 42 actual/correct predictions, with 5 and 10 wrong predictions, respectively

Decision tree :-

```
[ ] print("Confusion matrix : \n",confusion_matrix(y_test,predictions))

    Confusion matrix :
     [[28 16]
     [ 3 44]]
```

```
[ ] print(classification_report(y_test,predictions))

                  precision    recall  f1-score   support

               0       0.90      0.64      0.75        44
               1       0.73      0.94      0.82        47

        accuracy                           0.79        91
       macro avg       0.82      0.79      0.78        91
    weighted avg       0.82      0.79      0.79        91
```

```
[ ] # f1 score metrics
    f1_score_value = f1_score(y_test,y_pred,average='weighted')
    print("F1 score : ", f1_score_value)

    F1 score :  0.7857971312176919
```

Accuracy score – 0.79

Precision for class 0 – 0.90,Precision for class 1 – 0.73

Recall for class 0 – 0.64,Recall for class 1 – 0.94

F1 score – 0.78

In the output, there are 28 and 44 actual/correct predictions, with 3 and 16 wrong predictions, respectively

Knn

```
[ ]  print("Confusion matrix : \n",confusion_matrix(y_test,y_pred))

     Confusion matrix :
      [[35  9]
      [26 21]]
```

```
[ ]  print(classification_report(y_test,y_pred))

                   precision    recall  f1-score   support

                0       0.57      0.80      0.67        44
                1       0.70      0.45      0.55        47

         accuracy                           0.62        91
        macro avg       0.64      0.62      0.61        91
     weighted avg       0.64      0.62      0.60        91
```

```
[ ]  # f1 score metrics
     f1_score_value = f1_score(y_test,y_pred,average='weighted')
     print("F1 score : ", f1_score_value)

     F1 score :  0.6040626040626039
```

Accuracy score – 0.62

Precision for class 0 – 0.57,Precision for class 1 – 0.70

Recall for class 0 – 0.80,Recall for class 1 – 0.45

F1 score – 0.60

In the output, there are 35 and 21 actual/correct predictions, with 9 and 26 wrong predictions, respectively.

Ann

```
22/22 [==============================] - 0s 1ms/step - loss: 0.3884 - accuracy: 0.8443
Train score: 0.3883908689022064
Train accuracy: 0.8443396091461182
10/10 [==============================] - 0s 1ms/step - loss: 0.3503 - accuracy: 0.8352
Test score: 0.35032644867897034
Test accuracy: 0.8351648449897766
[[35  9]
 [ 6 41]]
```

```
[ ]  # f1_score
     from sklearn.metrics import f1_score
     # f1 score metrics
     f1_score_value = f1_score(y_test,y_pred,average='weighted')
     print("F1 score : ", f1_score_value)
```

```
F1 score :  0.8348049767091611
```

Train score – 0.38

Train accuracy – 0.84

Test score – 0.35

Test accuracy – 0.83

F1 score for ANN is 0.83.

Positive & Negative testing for each algorithms

Positive testing for logistic regression

1)In the domain of machine learning, logistic regression is considerably simpler to apply than other techniques. The setup of a machine learning model necessitates the training and testing of the model. The act of identifying trends in the input data such that the model can translate a certain input (say, a picture) to a specific output (such as a label) is known as training. In comparison to other techniques, logistic regression is simpler to train and apply.

2) When the data is linearly separable, logistic regression performs well: If a straight line can be drawn to separate two groups of data from one another, the dataset is shown to be linearly separable.

Negative testing for logistic regression

1)A continuous result is not predicted by logistic regression. To further appreciate this restriction, consider the following scenario. Logistic regression cannot be utilized to forecast how much a pneumonia patient's fever would climb in medical applications. This is due to the continuous measuring scale (Logistic regression only operates with dichotomous dependent or outcome variables).

Positive testing for decision tree -:

1) Decision trees are capable of generating rules that are easy to comprehend.

2) Decision trees are used to accomplish categorization without the need for extensive calculation.

Negative testing for decision tree-:

1) The training of a decision tree may be time-consuming and computationally costly. Computing resources are required to complete the process of building a decision tree. Each potential splitting column must be processed at each node before the optimum split can be determined for that field.

Positive testing for random forest -:

1. It helps to enhance decision tree accuracy by reducing overfitting.

2. It can handle both classification and regression issues with ease.

Negative testing for random forest-:

1. It necessitates a significant amount of computing power and resources since it constructs many trees and combines their results.

2. It also takes a long time to train since it uses a number of decision trees to decide the class.

Positive testing for knn-:

There was no training. In the absence of a model, K-NN labels the new data input-based learning from previous data, rather than building one. Most recent neighbour's class is assigned to new data entries.

Negative testing for knn

An unbalanced data set may create issues When the data is unbalanced, KNN doesn't function well

Positive testing for ANN:-

Data is saved on the whole system, not in a database, as it is in conventional programming. The network continues to function despite the loss of a few bits of data at one location.

Negative testing for ANN :-

Network activity that has gone unnoticed: It is ANN's most important problem. When ANN creates a testing solution, it does not explain why or how it was created. It erodes the network's trustworthiness.

FUTURE WORK:-

1) There is a requirement for additional machine learning matrics for comparison .

2) A large data set can be used for training
3) Develop a system or a methodology for automating the prediction of cardiac disease in the workplace.
4) Using deep learning, we build a Convoluted Neutral Network (CNN) that can understand and make smart decisions by layering algorithms.

CONCLUSION

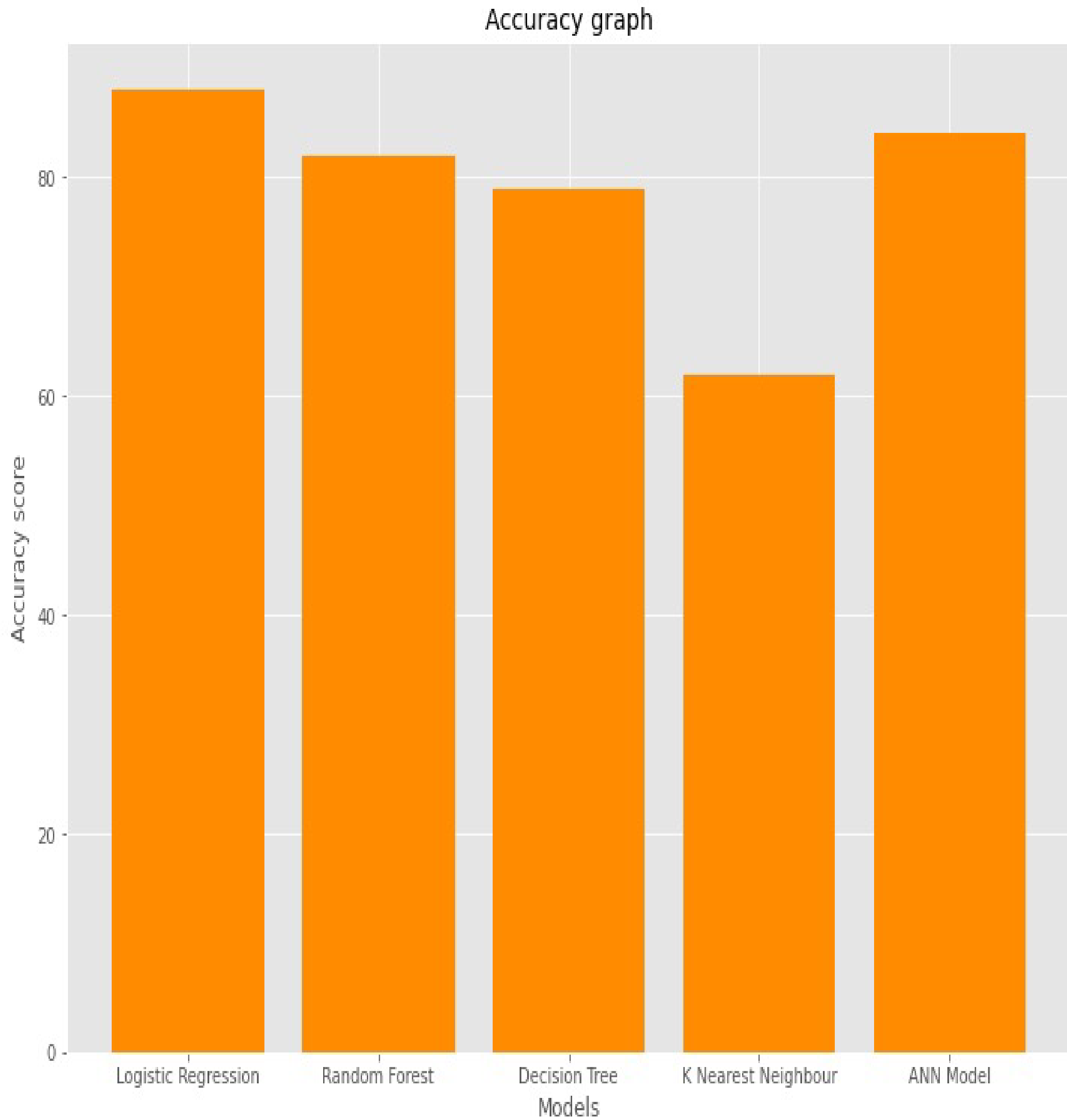
The significance of obtaining useful insight from raw material has extremely positive implications in many areas of life, including the medical profession, business, and many others. In this project , we tried to implement a multiple stage detection system for cardiac disease that was based on five algorithms, with the goal of determining which performed better. This report conducted a comparative analysis among machine learning algorithms

namely logistic regression, decision tree, KNN, random forest, and ANN. Through this analysis it was found out that the most accurate algorithm for conducting or predicting heart disease in a patient is logistic regression as it has the highest accuracy and F1 score among all the algorithms.The suggested detection algorithms were evaluated on the well-known Cleveland dataset in attempt to offer a fair comparison to previous research analysed through literature review. The results were promising. According to the experimental findings, the suggested model was able to surpass heart disease detection techniques in terms of accuracy, precision, recall, and the F1 score, among other things. Logic Regression outperformed the other four techniques or algorithms in terms of overall performance, as shown by the highest overall score achieved (accuracy of 88%). Methods for feature selection were used to further improve the overall performance.

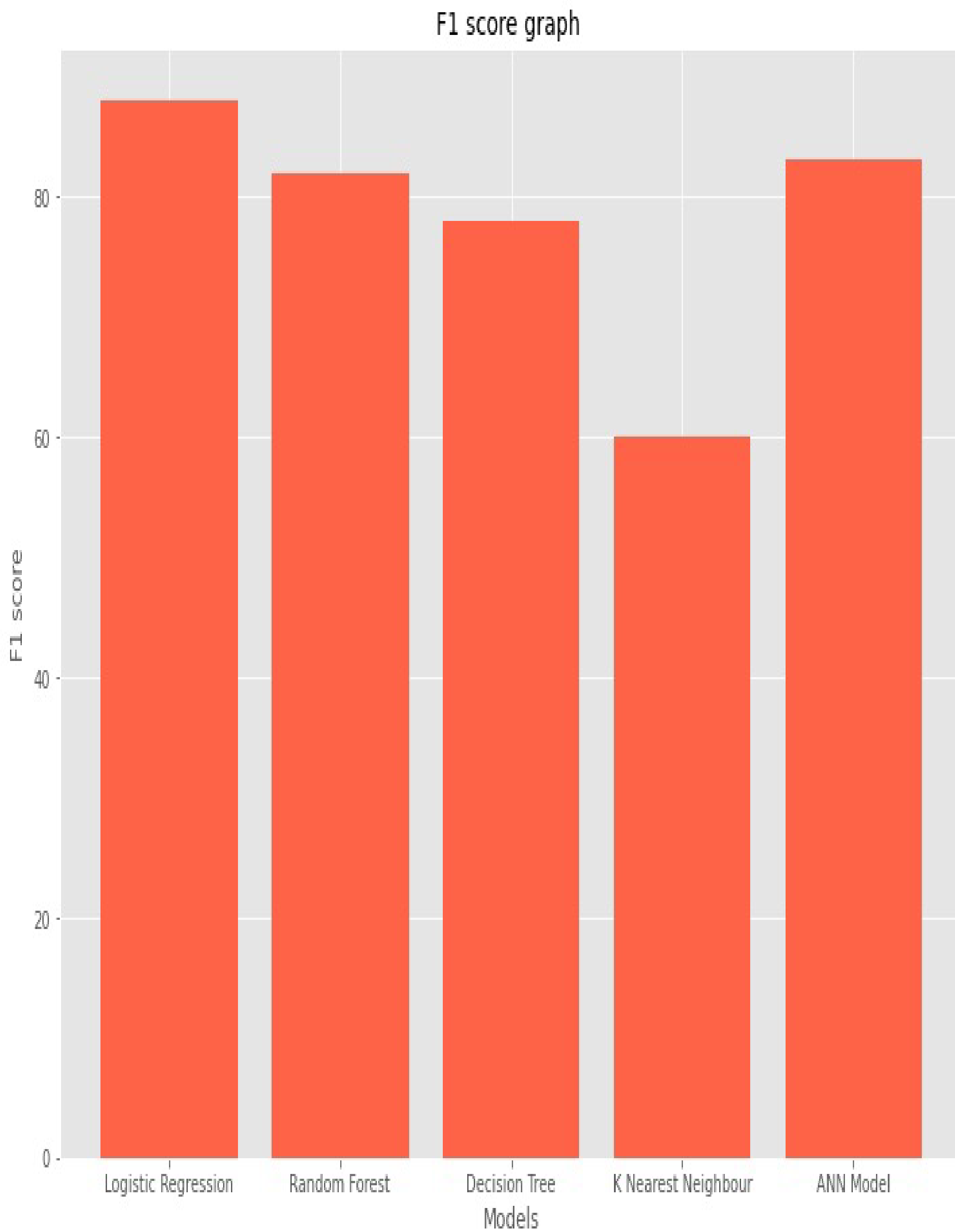
The accuracy, precision, recall, and F1 score of the ML algorithms were all improved as a result of the feature selection methods. When all five models are evaluated and assessed for effectiveness on the basis of chosen metric, the findings of the experiment indicate that LR performs greatest with an accuracy of 88%, trailed by ANN with accuracy of 84%.

Final graph prediction based on accuracy and F1 score comparing all algorithms

*ACCURACY GRAPH



**** F1 score***


F1 score graph
F1 score
80
60
40
20
0
Logistic Regression
Random Forest
Decision Tree
K Nearest Neighbour
ANN Model
Models

Our contribution:-

1) ***We tried the data that is linearly separable, logistic regression performs well: If a straight line can be drawn to separate two groups of data from one another,in our project we tried to show the dataset is shown to be linearly separable***
2) ***In Decision trees we conducted variable screening and feature selection without explicitly doing so.***

Decision trees are capable of dealing with both consistent and categorical data in the same situation.so we tried to modify data with both consistent and categorical work .

3) ***we tried to Effectively handles non-linear parameters: Unlike curve-based algorithms, non-linear factors have no effect on the efficacy of a Random Forest. As a result, if the independent variables are very nonlinear, Random Forest can surpass conventional curve-based methods. In our project we tried to handle non linear parameters .***
4) ***Basically Assumptions are not made in K-NN . But we tried to make number of assumptions that must be fulfilled in order to implement K-NN.***
5) ***We tried to implement Fault tolerance: The loss of one or more ANN cells does not prevent the network from producing output, and this***

characteristic makes the network fault tolerant.