HEART FAILURE PREDICTION

PRITESH DAGAR

Submitted for the Degree of Master of Science in

MSc Data Science And Analytics

Department of Computer Science
Royal Holloway University of London
Egham, Surrey TW20 0EX, UK

June 16, 2014

Declaration

This report has been prepared on the basis of my own work. Where other published and unpublished source materials have been used, these have been acknowledged.

Word Count: 12810

Student Name: Pritesh Dahar

Date of Submission: 2/9/2022

Signature: pritesh dagar

Heart Failure Prediction

Acknowledgements:

I firstly, would like to thank Professor Vovk,V at Royal Holloway University of London. The discussions and guidance were always helpful whenever I was confronted with any difficulties I came across, while constructing this project. Without his keen interest and knowledge my research would not have been successfully finished.

Abstract:

According to NHS, a Coronary heart condition (CHD) is typically caused by a build-up of fatty deposits (atheroma) on the walls of the arteries round the heart (coronary arteries).

Heart disease generally refers to conditions such as narrowed or blocked blood vessels that lead to heart failure, along with pain, and a reduced blood flow to the heart with the potential to cause a stroke.

I want to develop a system using machine learning techniques to predict heart disease. Currently It is impractical and expensive for people to undergo tests such as MRI scans. This system would be accessible, practical and reliable in predicting the likelihood of a cardiovascular problem (Heart Disease).

So, I tend to suggest developing an app that can predict cardiovascular disease based on basic symptoms like age, gender, cholesterol level, etc. Machine learning algorithms or models have proven to be the most correct and reliable formula and are therefore used in the system plan.

Heart disease prediction occurs in three phases:

Feature Selection Process: In this process, we automatically or manually select those features that contribute most to your predictor variable or desired outcome.

In the second phase, the machine learning algorithms such as Decision Tree, Random Forest, AdaBoost and XGBoost are applied, in which the data is trained and tested.

The third and last phase is the user interface, where the user enters their data and then the machine learning models predict whether the user is experiencing heart disease or not.

Contents

---

1. **Introduction:**

Humans in their early stages knew that the heart was important and powerful and when Ancient Egyptians Recognise the pain in the breast side of the heart and the arms suggested death was approaching.
Even the research and study have found the evidence of heart disease present in mummies.
Heart disease generally refers to any abnormal conditions occurred in the heart, research have found that heart disease applies to the conditions like collection of the fats around the heart muscles which is recognise to be harmful for human health, due to reduce blood flow to the heart which leads to further chest pain or stroke.
South Africa was the 1st to be successful for the heart transplant in 1967. However many surgeons were included in the operation which was for the successful but the patient couldn't live Or survive more than 18 days.
In 1993 there were total 2300 heart transplant was carried in United States. Through the knowledge in this field and research have been expanded drastically in past years but still heart disease remains and threat to human life.
On an average 900,000 Americans lose their life each year due to heart disease.
Each representative should always seek help from NHS or a professional doctor if found symptoms like shortness of breath, High maximum BP, unstable cholesterol level, fainting or any other unusual abnormal conditions occurred in human body.

An early stage recognition of all the symptoms which truly occurs heart disease using some machine learning models can immensely help human life when it comes to the matter of staying fit and to control death rate.

---

2. **Background Research:**

2.1.1 Project Lifecycle:

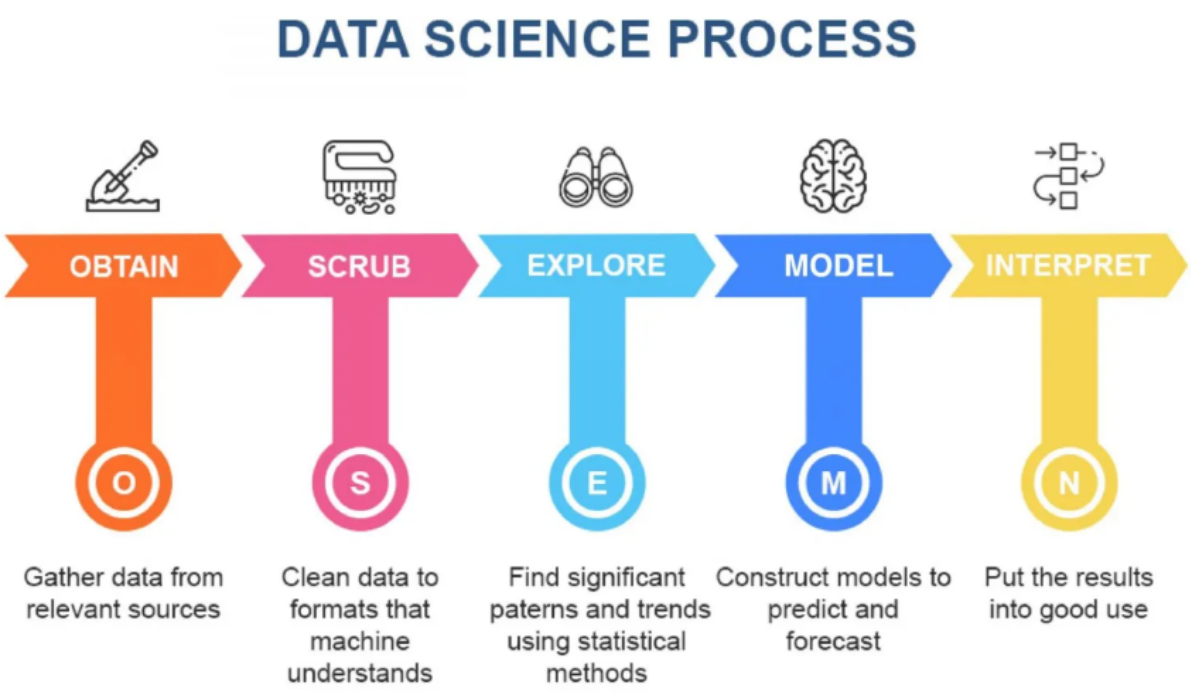


Image credit: **AnalytixLabs , reference:** https://www.analytixlabs.co.in/blog/data-science-life-cycle/

Obtaining the data:

Obtaining the data is the process in which one inherits the observed data which can be in Tabular form.
Whereas in this stage one having some technical knowledge about data storage like Oracle, MySQL, Mongo DB is very much beneficial when it comes to process and generate the data. It is easy to import tabular data that might be in the form of excel to some languages like Python or R.

Scrubbing the data:

In this process the data is been cleaned basically what it means that cleaning of all the null values present in the data set or any categorical values impacts a lot when it comes to modelling the data. So data cleaning or scrubbing in walls the format of cleaning the data in which machine learning model can understand and can predict results.

Exploring the data:

Exploring the data is the process in which the data is been thoroughly analysed with respect to each and every data points or features, Unique values and this features or plot and visualised for better understanding which does help drastically for a developer to decide which modelling structure could be used and could be irrelevant for the present features in data set.

Modelling the data:

Modelling Is the process in which all the refined data or clean data is been passed to various machine learning model in order to make prediction. Note: it is difficult for model to initialise categorical values because of this reason data scrubbing or cleaning place a vital role also when it comes to modelling.

Interpreting the data:

Interpreting the data is the final and important stage where all the Analysis, data cleaning, data modelling generates the final outcome which has to be deployed or interpret to the end client in order to improve business development decisions. Where in this place the data scientist develops a graphical user interface where the end client is able to interact and improve business decisions.

2.1.2 Hardware Requirement:

Desktop Computer

Processor: Intel i3 or above

AMD Ryzen3 or above

RAM: 2gb and above

HDD: 100gb or above

2.1.3 Software Requirement:

Operating System: Linux, MacOs, Windows 7 or above.

Python version 3 or above

Flask, Jsonify

Jupyter Notebook

Browser: Chrome, Firefox or any

2.1.4 Technology Used:

Libraries:

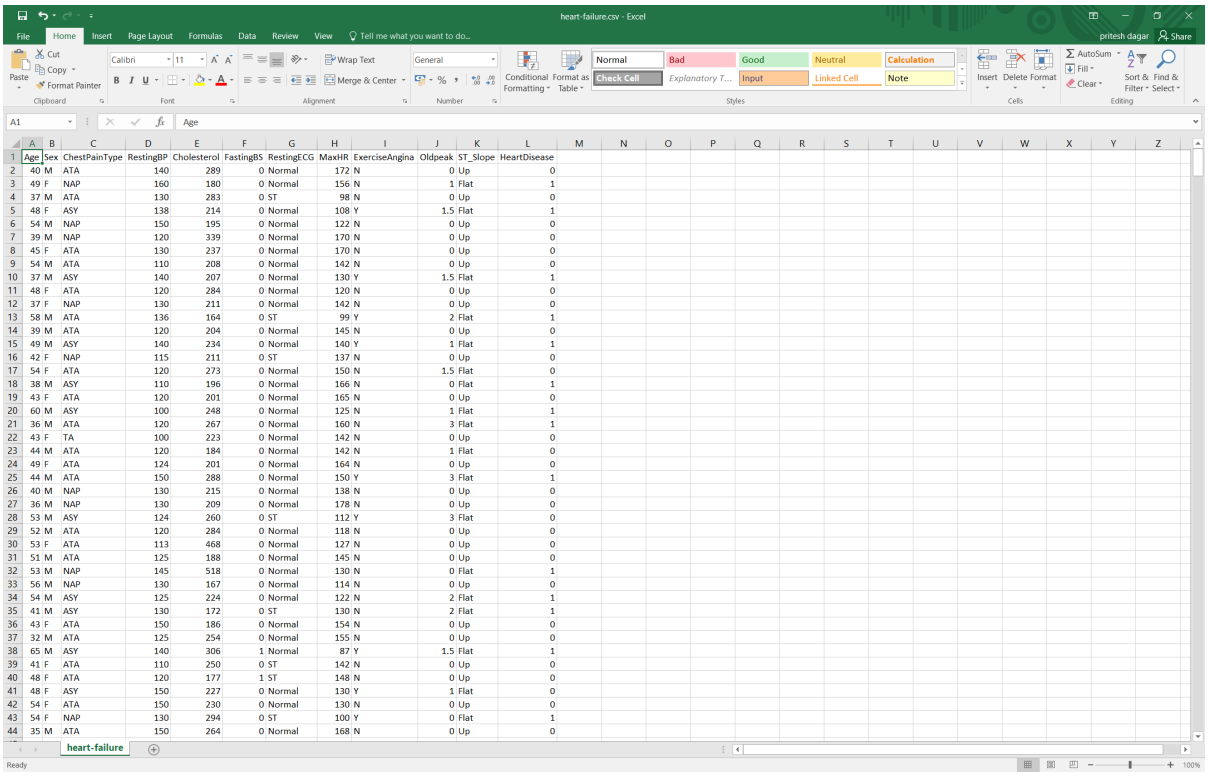
- `scikit-learn 1.1.2`
- `plotly`
- `numpy`
- `pandas`
- `cufflinks`
- `matplotlib`
- `seaborn`

2.1.5 About Software:

We have been using python throughout the journey because of its high level language and interactive environment, Another platform we have been using which is called as jupyter notebook.
Millions of data science libraries are integrated in python which makes it very easy to use pretty much just import and it’s ready to use.
Python is used for range of applications which includes video processing, image processing, Constructing of graphical user interface with the help of pyqt libraries.
There are plenty of Build in maths function in python which are ready to use and which are way more faster than spreadsheet or any traditional programming languages like Java or C++.

2.2 Data:

Dataset taken from: https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction



Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope	HeartDisease
40	M	ATA	140	289	0	Normal	172	N	0	Up	0
49	F	NAP	160	180	0	Normal	156	N	1	Flat	1
37	M	ATA	130	283	0	ST	98	N	0	Up	0
48	F	ASY	138	214	0	Normal	108	Y	1.5	Flat	1
54	M	NAP	150	195	0	Normal	122	N	0	Up	0
39	M	NAP	120	339	0	Normal	170	N	0	Up	0
45	F	ATA	130	237	0	Normal	170	N	0	Up	0
54	M	ATA	110	208	0	Normal	142	N	0	Up	0
37	M	ASY	140	207	0	Normal	130	Y	1.5	Flat	1
48	F	ATA	120	284	0	Normal	120	N	0	Up	0
37	F	NAP	130	211	0	Normal	142	N	0	Up	0
58	M	ATA	136	164	0	ST	99	Y	2	Flat	1
39	M	ATA	120	204	0	Normal	145	N	0	Up	0
49	M	ASY	140	234	0	Normal	140	Y	1	Flat	1
42	F	NAP	115	211	0	ST	137	N	0	Up	0
54	F	ATA	120	273	0	Normal	150	N	1.5	Flat	0
38	M	ASY	110	196	0	Normal	166	N	0	Flat	1
43	F	ATA	120	201	0	Normal	165	N	0	Up	0
60	M	ASY	100	248	0	Normal	125	N	1	Flat	1
36	M	ATA	120	267	0	Normal	160	N	3	Flat	1
43	F	TA	100	223	0	Normal	142	N	0	Up	0
44	M	ATA	120	184	0	Normal	142	N	1	Flat	0
49	F	ATA	124	201	0	Normal	164	N	0	Up	0
44	M	ATA	150	288	0	Normal	150	Y	3	Flat	1
40	M	NAP	130	215	0	Normal	138	N	0	Up	0
36	M	NAP	130	209	0	Normal	178	N	0	Up	0
53	M	ASY	124	260	0	ST	112	Y	3	Flat	0
52	M	ATA	120	284	0	Normal	118	N	0	Up	0
53	F	ATA	113	468	0	Normal	127	N	0	Up	0
51	M	ATA	125	188	0	Normal	145	N	0	Up	0
53	M	NAP	145	518	0	Normal	130	N	0	Flat	1
56	M	NAP	130	167	0	Normal	114	N	0	Up	0
54	M	ASY	125	224	0	Normal	122	N	2	Flat	1
41	M	ASY	130	172	0	ST	130	N	2	Flat	1
43	F	ATA	150	186	0	Normal	154	N	0	Up	0
32	M	ATA	125	254	0	Normal	155	N	0	Up	0
65	M	ASY	140	306	1	Normal	87	Y	1.5	Flat	1
41	F	ATA	110	250	0	ST	142	N	0	Up	0
48	F	ATA	120	177	1	ST	148	N	0	Up	0
48	F	ASY	150	227	0	Normal	130	Y	1	Flat	0
54	F	ATA	150	230	0	Normal	130	N	0	Up	0
54	F	NAP	130	294	0	ST	100	Y	0	Flat	1
35	M	ATA	150	264	0	Normal	168	N	0	Up	0

2.2.1 Context:

As discussed above in the introduction section which generally determines the risk factor for cardiovascular diseases Which is also termed as heart disease where on an average 900,000 Humans lose their life each year. The data set contains 11 columns which are also termed as features and one more column named as heart disease can also be termed as target feature.

An early stage recognition of all the symptoms which truly occurs heart disease using some machine learning models can immensely help human life when it comes to the matter of staying fit and to control death rate.

2.2.2 Feature Variable:

Age: Age determines age of participant or person

Sex: Sex determines gender of participant or person

ChestPainType: Chest pain type a deter mind as TA, ATA, NAP, ASY

RestingBP: Resting BP determines The count of participant or person

Cholesterol: Cholesterol determines numerical value for participant or person

FastingBS: If fasting blood sugar > 120 MG = 1, else 0

RestingECG: Resting electrocardiogram <normal> <st>

MaxHR: Maximum heart rate ranges from 60 to 202

ExerciseAngina: Exercise angina Determines whether there is exercise included yes or no

Oldpeak: Oldpeak is the value generated when the person or participant is in depression

ST_Slope: ST_slope is the peak exercise segment which is distributed as up, flat, down

---

2.2.3 What the Problem is:

- In the given data set we have a classification problem considering the data of all the people present in the dataset which will help further to make prediction on target variable heart disease.
- Data set consist of categorical values which will further need conversion in numerical form in order to pass the values in machine learning models.
- The outcome would be testing and training the data sets on various classification models and then comparing the best models that gives prediction on heart disease.
- After comparing the best model will be integrated to graphic user interface in order to accept input from the user and predict the target variable which is heart disease.

2.2.4 Target Variable:

Heart Disease: Heart disease is the target feature where one determines that the person is having a risk of heart disease and zero determines person having no risk of heart disease.

2.2.5 List of PYTHON Function:

`read_csv`: reads comma separated values files into dataframe

`sample()`: it will return any random values from scquencial data

`nunique()`: this will return number count or unique values from feature

`groupby()`: this will split the data into groups

`size()`: this will return array size

`describe()`: this Describes the statistics summary only on numerical data, not on categorical data.

`info()`: this will print the information of given feature column

`head()`: this will return the 1st row from given dataset

`value_counts()`: will return how many unique values present in each and every feature column

`unique()`: will return unique values from every feature column

`hist()`: will create histogram chart for given feature column

`show()`: show function from matplotlib library will display the plot for given features

`bar()`: show rectangular bars

`barplot()`: show categorical and numerical values feature column together

`distplot()`: Seaborn disptlot() method plots the line on histogram

`pairplot()`: show relationship between one and two variables

`countplot()`: show counts from various feature wrt bars

`heatmap()`: shows correlations between features

`subplot()`: divides plots into number of plots inn single frame

`violinplot()`: shows feature selected in the form of leaves

`swarmplot()`: shows the visualization in the form of high to low or low to high density bullet dots

`pointplot()`: shows difference between 2 features in the form of points adjoining with line

`train_test_split()`: splits the data into random which can be defined by any numeric value eg: birthdate or any and divides the data in percentage for training and testing purpose. eg: 70-30

`StandardScaler()`: performs data task in standardize format for the features in dataset which are different in scale

`fit()`: fits the data for modelling whish was trained

`transform()`: transforms the value on dataset parameters in order to normalize value

`predict()`: predicts the value on the basis of trained data, in our case its 0 or 1.

`dump()`: for storing the model into a file

`DecisionTreeClassifier()`: by using decision trees it creates classification model

`accuracy_score()`: this returns the accuracy score of each algorithm or model applied on test data

`barh()`: this plots horizontal columns as bars

`RandomForestClassifier()`: this solves classification or regression problem which tries to fit in various decision trees on features of dataset to get good accuracy score and also controls

overfitting

`XGBClassifier()` : divides the data into parallel subset decision tree which leads to boosting for classification or regression problem

`AdaBoostClassifier()` : which which help to improve the performance of model or algorithm

2.3 Reading The Data:

```
# importing csv format data to dataframe names as 'heart'
heart = pd.read_csv(r'./heart-failure.csv')
```

```
heart
```

	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope	HeartDisease
0	40	M	ATA	140	289	0	Normal	172	N	0.0	Up	0
1	49	F	NAP	160	180	0	Normal	156	N	1.0	Flat	1
2	37	M	ATA	130	283	0	ST	98	N	0.0	Up	0
3	48	F	ASY	138	214	0	Normal	108	Y	1.5	Flat	1
4	54	M	NAP	150	195	0	Normal	122	N	0.0	Up	0
...	...	...	...	...	...	...	...	...	...	...	...	...
913	45	M	TA	110	264	0	Normal	132	N	1.2	Flat	1
914	68	M	ASY	144	193	1	Normal	141	N	3.4	Flat	1
915	57	M	ASY	130	131	0	Normal	115	Y	1.2	Flat	1
916	57	F	ATA	130	236	0	LVH	174	N	0.0	Flat	1
917	38	M	NAP	138	175	0	Normal	173	N	0.0	Up	0

918 rows × 12 columns

```
The above dataset columns represents: Age,Sex,ChestPainType,RestingBP,Cholesterol,
FastingBS,RestingECG,MaxHR,ExerciseAngina,Oldpeak,ST_Slope. These will be refered to as 'features' or 'X'.
The HeartDisease column is the 'target feature', or 'Y', which we are going to 'predict' given the data supplied.
```

```
heart.sample(10)
```

	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope	HeartDisease
375	73	F	NAP	160	0	0	ST	121	N	0.0	Up	1
127	48	F	ASY	108	163	0	Normal	175	N	2.0	Up	0
810	55	F	ATA	135	250	0	LVH	161	N	1.4	Flat	0
767	54	F	NAP	108	267	0	LVH	167	N	0.0	Up	0
440	52	M	NAP	128	0	0	ST	180	N	3.0	Up	1
577	67	M	ASY	146	369	0	Normal	110	Y	1.9	Flat	1
793	67	M	ASY	125	254	1	Normal	163	N	0.2	Flat	1
160	59	M	ASY	140	264	1	LVH	119	Y	0.0	Flat	1
358	38	M	NAP	115	0	0	Normal	128	Y	0.0	Flat	1
123	58	F	ATA	180	393	0	Normal	110	Y	1.0	Flat	1

```
heart.columns
```

```
Index(['Age', 'Sex', 'ChestPainType', 'RestingBP', 'Cholesterol', 'FastingBS',
       'RestingECG', 'MaxHR', 'ExerciseAngina', 'Oldpeak', 'ST_Slope',
       'HeartDisease'],
      dtype='object')
```

```
heart.nunique() #how many unique values we have in our Dataset for each column
```

```
Age               50
Sex                2
ChestPainType      4
RestingBP         67
Cholesterol      222
FastingBS          2
RestingECG         3
MaxHR            119
ExerciseAngina     2
Oldpeak           53
ST_Slope           3
HeartDisease       2
dtype: int64
```

```
heart['HeartDisease'] # Total number of rows in dataset ie: 917 w.r.t target feature.
```

```
0      0
1      1
2      0
3      1
4      0
      ..
913    1
914    1
915    1
916    1
917    0
Name: HeartDisease, Length: 918, dtype: int64
```

```
heart.groupby('HeartDisease').size()
```

```
HeartDisease
0    410
1    508
dtype: int64
```

```
The above analysed data shows how many people have heart disease, and those who do not have heart disease in the given data set.
0 refers to the number of people free from heart disease, which in this case is 410.
1 refers the number of people who have heart disease, which in this case is 508.
```

```
heart.groupby('Sex').size() # based on gender from the whole dataset
#no of females and males present in whole dataset.
```

```
Sex
F    193
M    725
dtype: int64
```

```
heart.groupby(['Sex', 'HeartDisease']).size() # based on gender, but in group wise targated variable.
#In the statistical data below we can observe that among females, there is a total of 143 females who do not have heart disease.
#and a total 50 females who have heart disease.
#This conclusion includes all female paricipants who took part in the study, totalling 193.
#Males
#267 males who do not have heart disease.
#458 males who has heart disease.
#total 267+458=725 males in overall dataset.
```

```
Sex  HeartDisease
F    0              143
     1               50
M    0              267
     1              458
dtype: int64
```

```
heart.describe()
#This Describes the statistics summary only on numerical data, not on categorical data.
#If we need to analyse statistical values of
#categorial data then we will have to clean the data and convert categorial values into numerical values.
#eg: in the data below we can compare the cholesterol levels between the 918 rows of data, with the healthy average being 198.
# where some people have cholesterol levels as low as 0.0, the  maximum cholestrol level reaches 603.
#25% of people have cholesterol levels of 173.
#50% of people have cholesterol levels of 223.
#75% of people have cholesterol levels of 267.
```

	Age	RestingBP	Cholesterol	FastingBS	MaxHR	Oldpeak	HeartDisease
count	918.000000	918.000000	918.000000	918.000000	918.000000	918.000000	918.000000
mean	53.510893	132.396514	198.799564	0.233115	136.809368	0.887364	0.553377
std	9.432617	18.514154	109.384145	0.423046	25.460334	1.066570	0.497414
min	28.000000	0.000000	0.000000	0.000000	60.000000	-2.600000	0.000000
25%	47.000000	120.000000	173.250000	0.000000	120.000000	0.000000	0.000000
50%	54.000000	130.000000	223.000000	0.000000	138.000000	0.600000	1.000000
75%	60.000000	140.000000	267.000000	0.000000	156.000000	1.500000	1.000000
max	77.000000	200.000000	603.000000	1.000000	202.000000	6.200000	1.000000

```
For the purpose of this study, converting the categorical values into numerical values will not bring any benefits at this
stage. For example, if we convert male and female into 0 and 1 the answer from the describe function, being 0.5, does not help.
This is not usable when compared to other categorical variables with regard to chest pain type. At this stage converting the
various chest pain types into 0,1,2,3, will not be beneficial in a data analytics perspective.  We will, however, need to
convert the category Variables when it comes to modelling or passing the data in models.
```

```
# Handling Null Values

heart.info() #information about dataframe.
#where non-null count determines that there is no null values or empty values in any of the columns
#Dtype refers to data type of each column where age is int64 (integer value) & all the categorical values will be determined
#as object dtype
#so ml model can take only integer and float value but not object values so thats why further while modelling we will convert
#all the categorical values to numerical values.
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 918 entries, 0 to 917
Data columns (total 12 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Age             918 non-null    int64
 1   Sex             918 non-null    object
 2   ChestPainType   918 non-null    object
 3   RestingBP       918 non-null    int64
 4   Cholesterol     918 non-null    int64
 5   FastingBS       918 non-null    int64
 6   RestingECG      918 non-null    object
 7   MaxHR           918 non-null    int64
 8   ExerciseAngina  918 non-null    object
 9   Oldpeak         918 non-null    float64
 10  ST_Slope        918 non-null    object
 11  HeartDisease    918 non-null    int64
dtypes: float64(1), int64(6), object(5)
memory usage: 86.2+ KB
```

2.4 Decision Tree Algorithm and Flow Chart:

Decision trees is supervised machine learning technique where on the based on various conditions the decisions are been made.

Decision tree works as a normal tree structure includes branches which are interconnected to root nodes and leaf nodes.

When the data is been submitted to decision tree algorithm, Root node get selected first.

In the consideration of features and target feature From our given data set, where the targets

feature Is fixed and the rest other feature variables are independent. All the independent feature variables are candidate for root node, But which feature variable will be selected for the root node Depends on two criteria Which are information gain IG and gini index.
Decide the two criteria is from which decision tree internally decides Which feature column to select for root node Or subsequent notes Invert filter criteria to give.

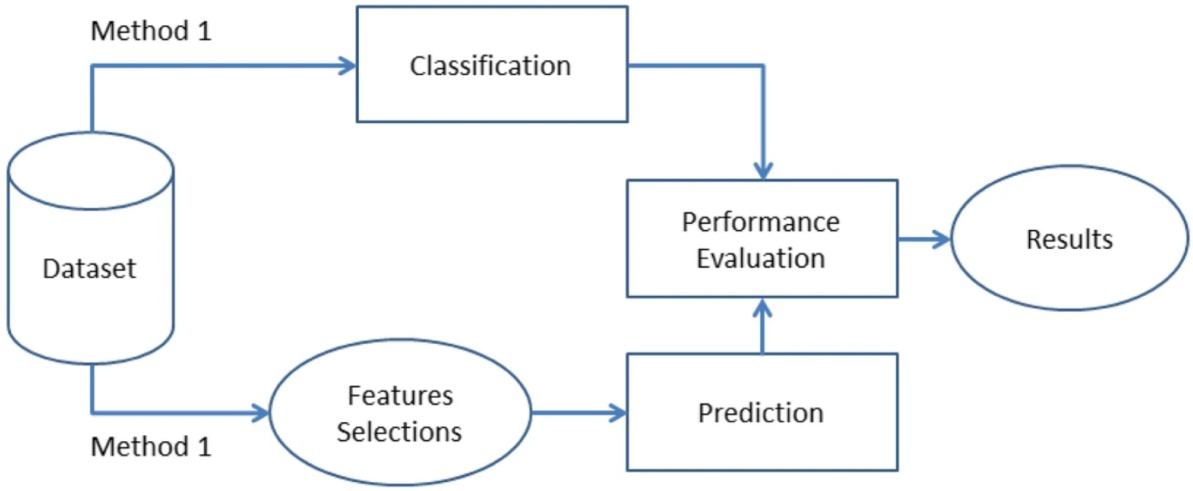


Image Credit: **BMC Bioinformatics , Reference:** https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-020-03626-y/figures/10

2.5 Random Forest Algorithm and Flow Chart:

What is ensemble learning?
ensemble learning is the method in which it takes input from multiple models and combines the learning.
The advantage of decision tree is It is easy to understand, Easy to train the model, Easy to interpret because decision tree works on nested if else condition.
The disadvantage of decision tree is that it trends to overfit. Indecision tree when the branches keep growing till the last node or leaf, There is a tendency of decision tree Of overfitting or in other words we can say high variance.

The advantage of random forest is it reduces the high variance, Which means it minimises the variance of the model. In random forest decision trees are built using bagging techniques.

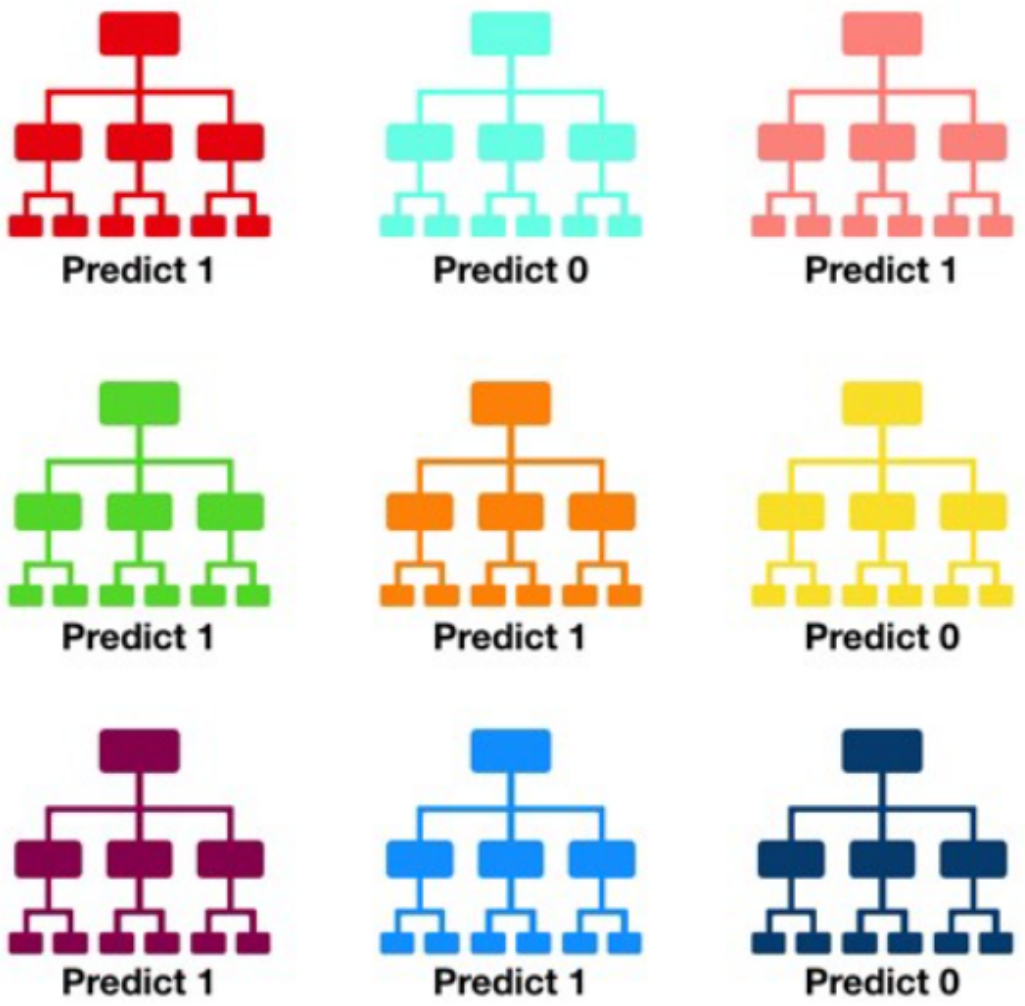


Image credit: Tony Yiu, reference: https://towardsdatascience.com/understanding-random-forest-58381e0602d2

In the above diagram each tree in different colours predict some value and the prediction value containing majority of the trees becomes the model prediction.

2.6 XGBoost Classifier Algorithm:

XGBoost is a boosting algorithm.
What is boosting algorithm?
Boosting algorithm is an ensemble technique of sequential learning.
Sequential learning means the difference between bagging and boosting is where bagging is a parallel ensemble And boosting is a Sequential ensemble, which means in boosting different models get trained one after another.
In that manner many models will combine together in order to give best result.
XGBoosting is an extension of gradient boost.

2.7 AdaBoost Classifier Algorithm:

What is a ensemble learning?
And symbol learning is basically I learning technique in which Multiple individual model combine together in order to create Master model This process is called ensembling.
Let’s discuss what is the difference between random forest and adaboost.
Random Forest is a parallel learning process where as adaboost is a sequential learning process.
The sequential process which we follow in ada boost, One tree is dependent on the another tree.
Which means if there are multiple machine learning model implemented.
In all the cases the next model is dependent on the output from previous model. This process where all the models are dependent on each previous model is called as sequential learning.
In adaboost All the trees or all the models do not have equal weights. Which means some of the models will have more weights in final model and some of the individual model will have less weights in the final mode.
In Adaboost trees are not fully grown, Rather the trees or just one root and two leaves.
Specifically they are called as stumps.
In adaboost some of the weak learners have higher weight as compare to other weak learners.

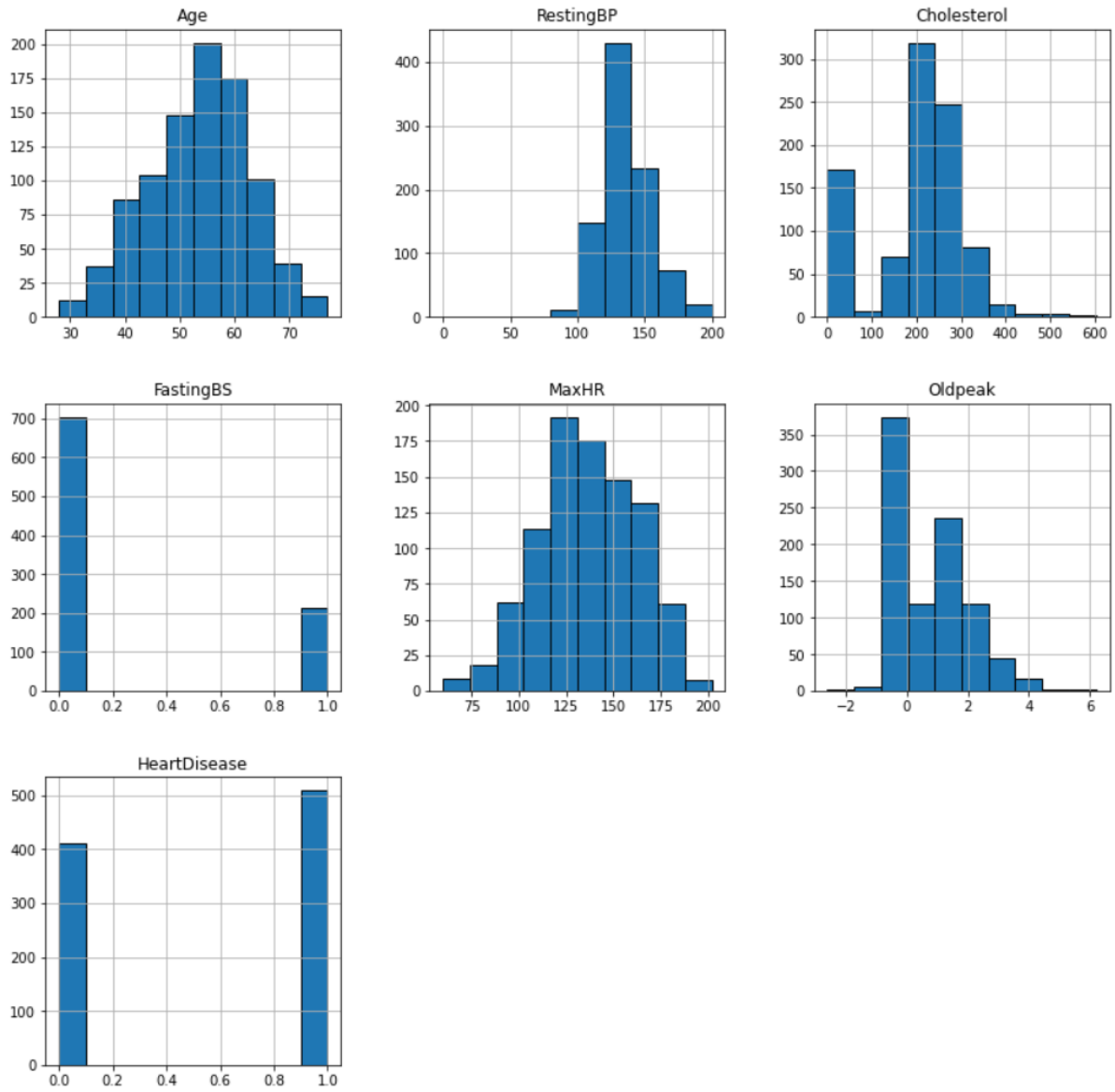
3. **Implementation:**

3.1 Exploratory Data Analysis (EDA):

Why do we need Exploratory Data Analysis??

Exploratory data analysis Is an approach you make sense of the data provided to you and summarise their main characteristics often in the form of visualisation is the most important step in debt analysis process.
The objective to understand your data is very crucial and most important step in the process of data and assess where you need to ask the right question.
With the help of explorative data and assess it gives you clear idea about feature attributes and target attribute a feature.
EDA is the process where the insights are been generated In order to have a clear idea about business problem and to analyse all the essentials and non-essential features of the data set.
It also includes cleaning of the data where null values are been removed or conversion of categorical values into numerical format where the data has been pre-processed and ready to pass it to models for making predictions.

3.2 Visualization And Explanation



Age: X-axis we have age of people range from 0-80

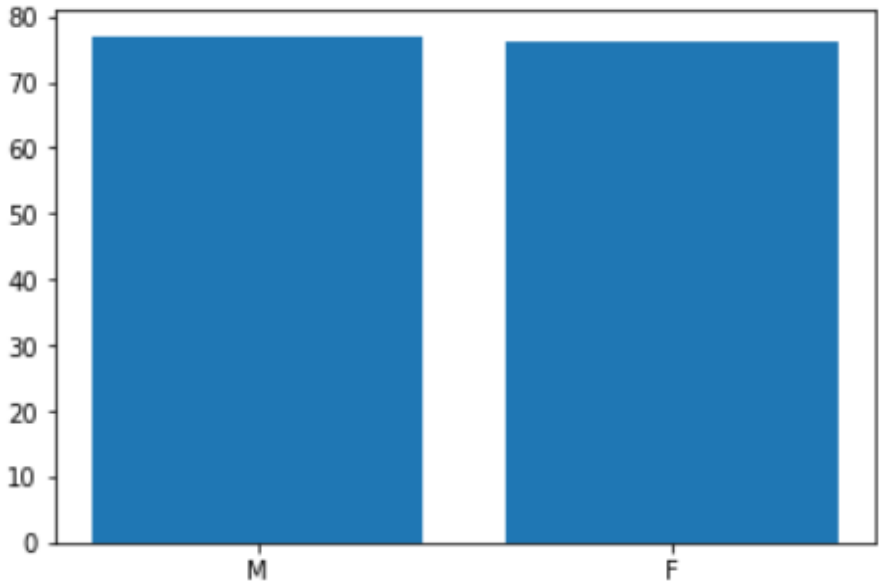
y-axis we have number of people

This bar chart shows that how many number of people having heart disease with respect to their age throughout the given dataset.

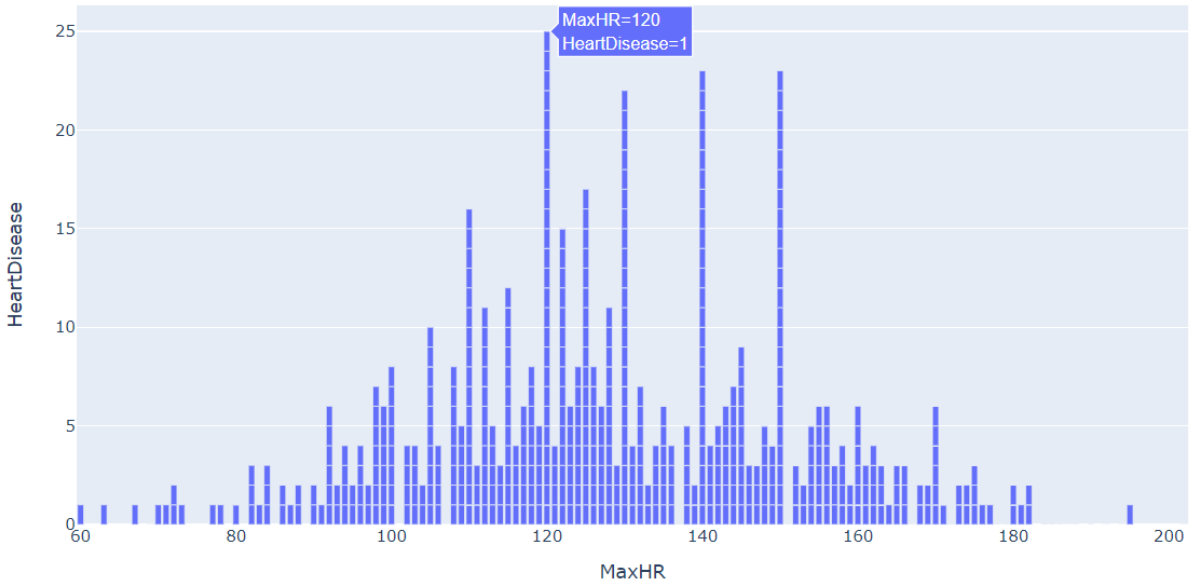
Example:
By looking at the graph we can say that almost 200 people having age range from 50-60 are affected by heart disease.

Heart Disease: X-axis we have float values 0.0 means no heart disease
Where as 1.0 means heart disease

On the Y-axis we have number of people
So graph preview helps us to clearly determine that around 400 peoples do not have heart disease, and around 500 people are affected by heart disease.

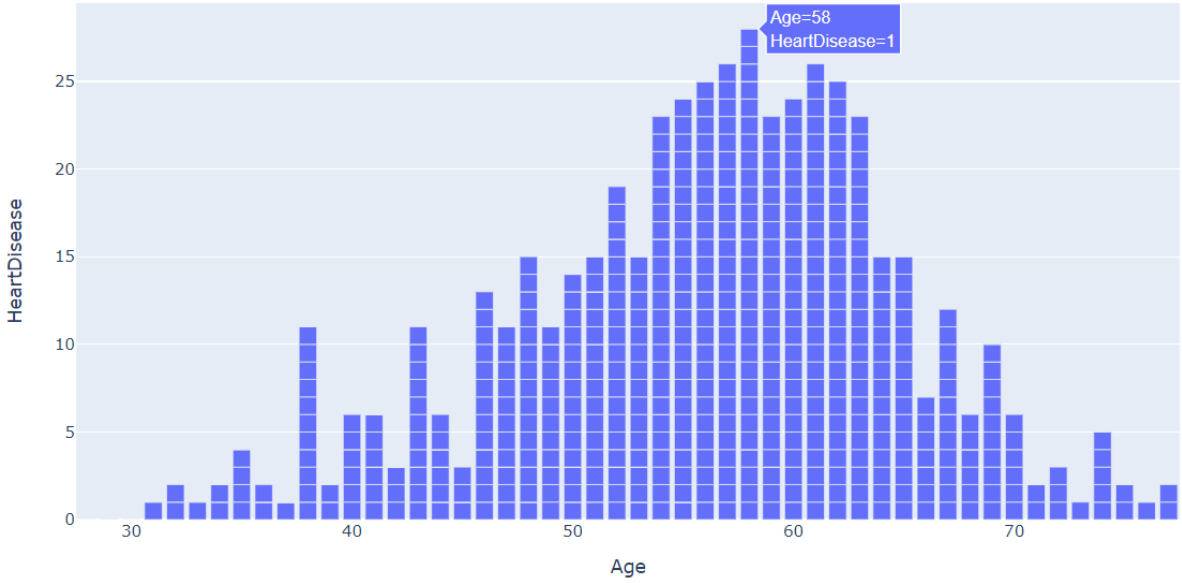


This bar chart above indicates the age factor of male and female in present dataset, which clearly indicated that males slightly have more age then female.

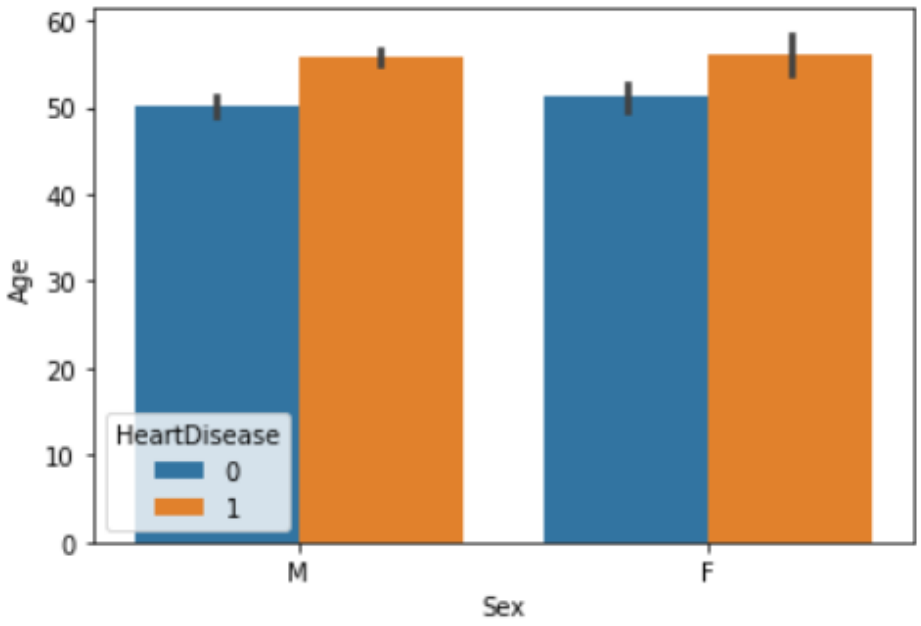


In this diagram we Are only getting the information of all those people having heart disease, When the cursor is Hoovered On the graph it is easy to understand that how the Maximum heart rate value changes.
X-axis = maxhr
Y-axis = number of people
In The overall data set total 25 people are having maximum heart rate of 120.

Up to 16 people or having maximum heart rate of 110.



In the above graph each block height is equivalent to one single person having heart disease in the Database.
X-axis = age of people having heart disease
Y-axis = number of people having heart disease
Which says that there are 28 number of peoples in the dataset of age 58 having heart disease.



This Graph above uses seaborn library, where with the help of this library we have tried to visualise 3 attributes or features from dataset.
X-axis consist of male and female
Y-axis consist of age of male and female
and color combination consist of heart disease where again 0 means no heart disease and 1 means heart disease.

So for males who does not have heart disease falls at maximum age of 50
Kindly note this graph does not gives information about average age of male or female having or not having heart disease.

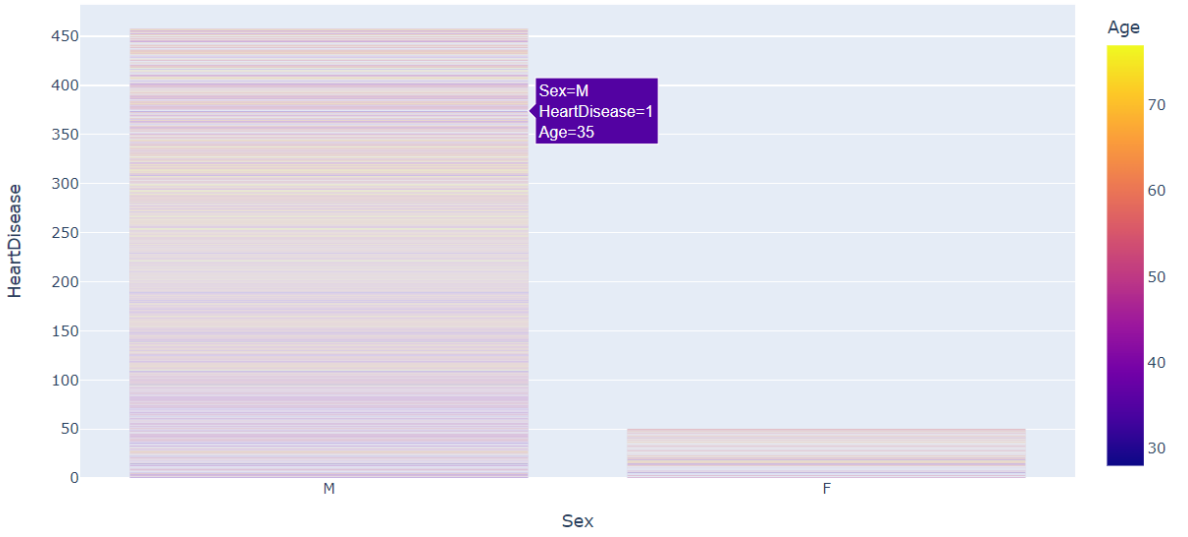
Instead each bar starting from male shows that maximum age of male not having heart disease is 50.
Maximum age of male having heart disease have age 55 approximately
Maximum age of female not having heart disease is around 51-52
Maximum age of female having heart disease is around 55-56.

Flaw: This particular graph does not show much insight because there males and females having heart disease who's age is 30 or there are again males and females not having heart disease whose age is 30.

```
heart.groupby(['Sex', 'HeartDisease']).size()
```

```
Sex  HeartDisease
F    0              143
     1               50
M    0              267
     1              458
dtype: int64
```

In above statically data we can observe that among female there are total 143 female not having heart disease.
and total 50 females having heart disease so that total makes to 143+50=193 females in overall dataset.
Where as
267 males not having heart disease
458 males having heart disease
total 267+458=725 males in overall dataset



X-axis = males and females
Y-axis = All number of people having heart disease

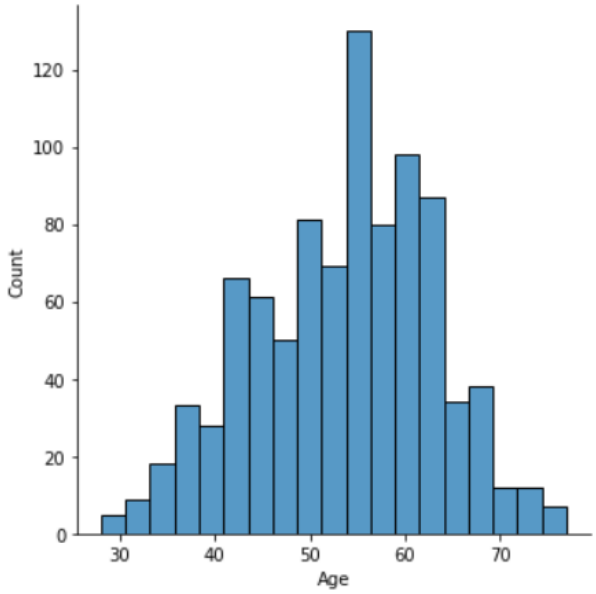
Age scale consist of some color combination where people age range from 30-40 have dark blue color

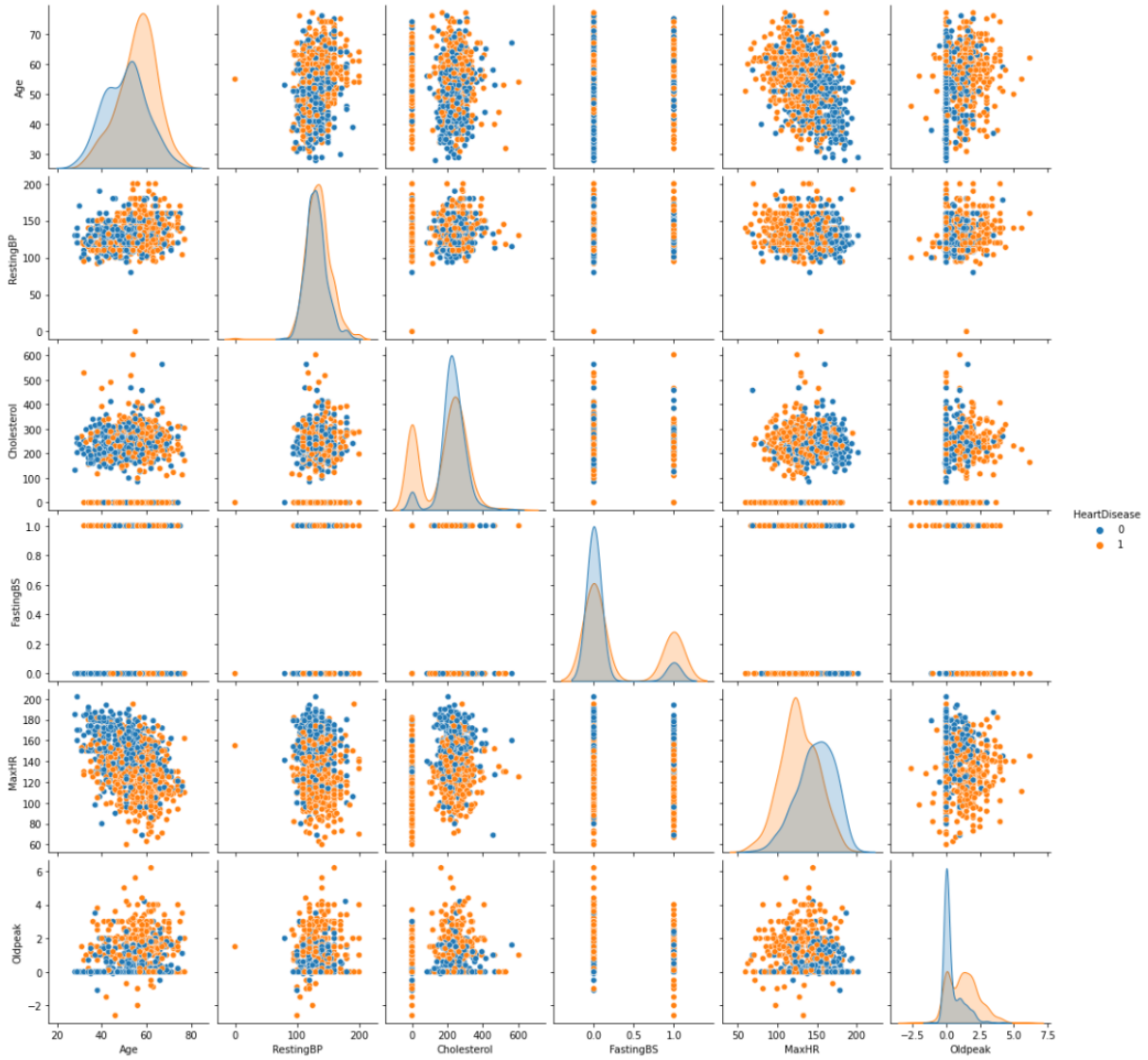
People with age 50-65 bit orangeish color
People above 70 with yellow color

So The reason why this visualisation is called as interactive because when you try to Hoover your cursor at the plot it is easy to observe colour combination of all the males with respect to their age.
For example if you hoover the cursor right by the scale of 50 we can see that male whose age is 44 having heart disease which is clearly seen in Dark purple colour this Goes same with the male having age 49 and 50.

Whereas in females it is easy to identify the age of females having heart disease by colour combination because of the interactive design or visualisation where hoovering the cursor will show you the age of the female and in what colour range does her age falls into with respect to having heart disease.

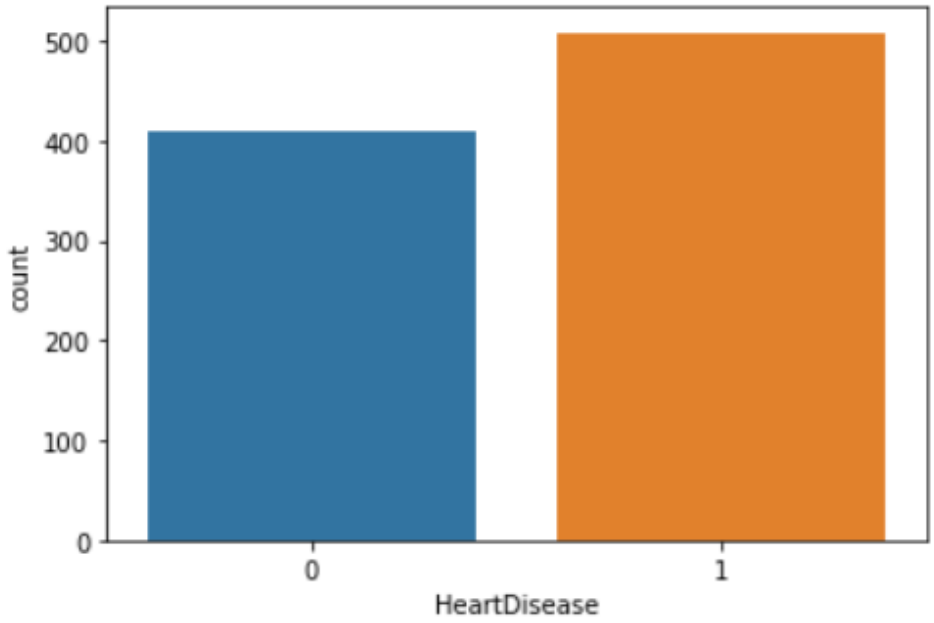
```
sns.displot(heart["Age"]) #seaborn library
plt.show() #reason we use function from matplotlib in order to display the graph
           #is to overcome from <seaborn.axisgrid.FacetGrid at 0x1e2bae81550> this error
```



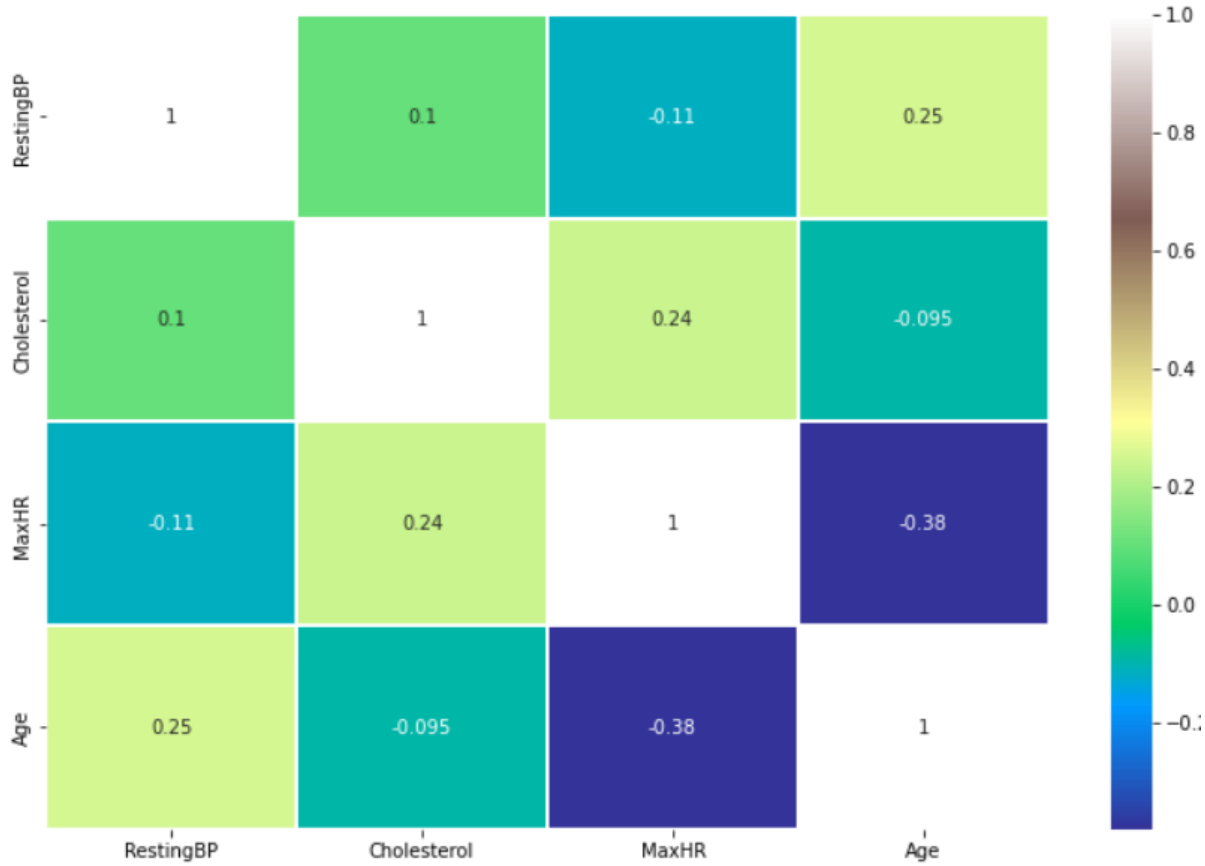


To visualize the one-dimensionality of the variables it's a decent apply to plot distribution graph and appearance for imbalance of features.

```
1    508
0    410
Name: HeartDisease, dtype: int64
```

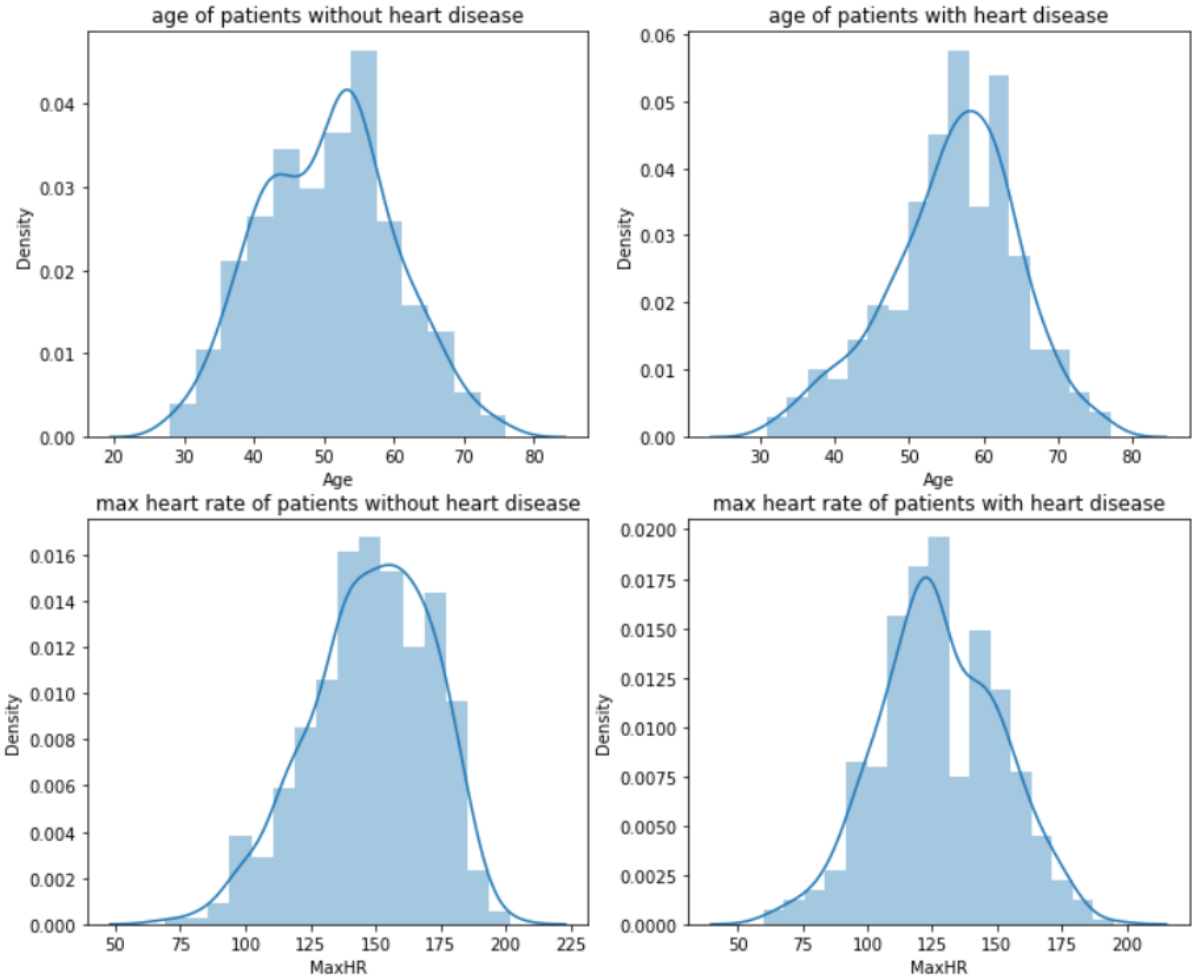


This bar plot using Seaborn library is fairly giving the same data which was previously given by pie chart displayed with the help of plotly library, easy to understand where X axis consist of zeros means no heart disease and one means having heart disease and on the Y axis is the number of people in the data set clear visualisation which introduce around 508 people in the whole data set or having heart disease and 410 people in the data set or not having heart disease but they could be either male or female any of them.



The above visualisation Describes the correlation between features present in the Dataset.
When noticing the age factor in X and Y axis which usually determines hundred percent correlation so its 1.
And when considering another features which is maximum heart rate on X axis with respect to maximum heart rate on Y axis are Hundred percent correlated this goes same with cholesterol and resting blood pressure. So in that case we need to ignore that correlation relation because there is no comparative Relation between two same features.

So when comparing resting BP on X axis with respect to age on Y axis we could suspect 0.25 correlation Between these two features
This means how much age factor is affecting resting blood pressure,
How much age factor is affecting cholesterol,
How much age is affecting maximum heart rate.



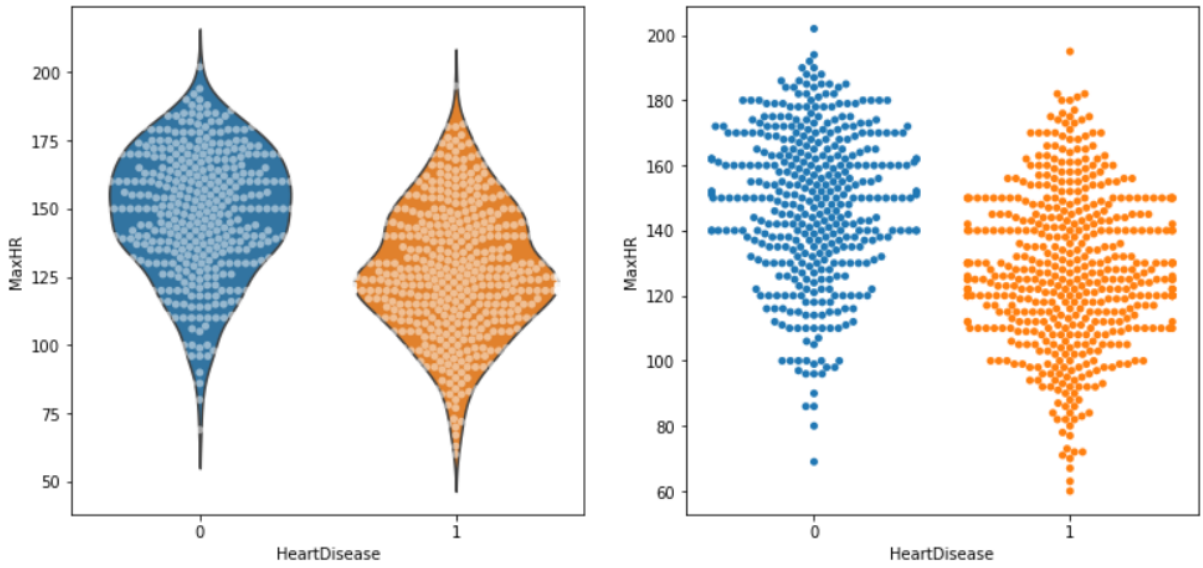
Seaborn disptlot() method plots the line on histogram.
There are many kind of variation can be Visualised with the help of displot method.
In this case we are using subplot method to show 4 variance at same time or in the same visualisation.

age of patients without heart disease: The visualisation with respect to age of patients without heart disease showcase That age varies from 20-80 Where we can suspect the peak from 52-57 Falls in the range of age where not many patients are having heart disease.

age of patients with heart disease:In this plot it is easy to suspect that people from age 58-65 Falls in the group of people with respect to age who are mostly having heart disease.

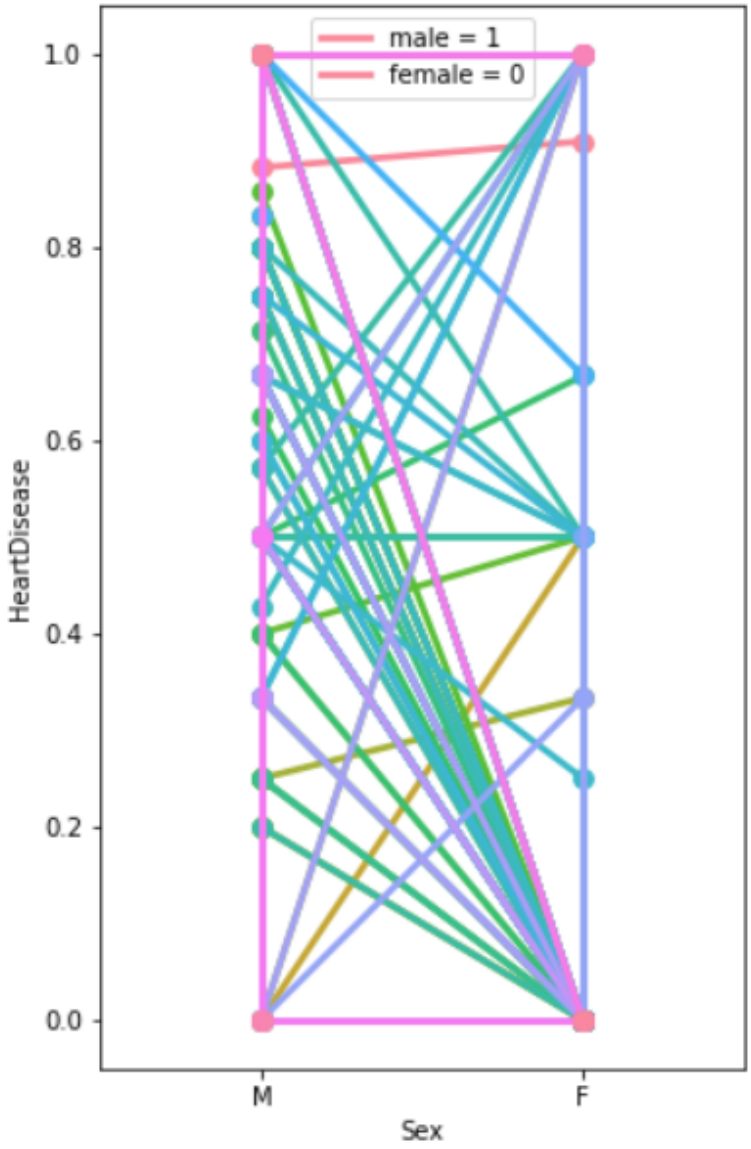
max heart rate of patients without heart disease: This plot showcase the visualisation of the patients without heart Disease with respect to maximum heart rate.

max heart rate of patients with heart disease: This plot also suspects the maximum heart rate data of the patients who are affected with a disease.



In this visualisation above this looks like a leaf structure is easy to understand considering heart disease where zero means no heart disease which is blue in colour and one means having heart disease which is orange in colour so people who are not having heart disease has an average numeric value for maximum heart rate is 150.

Where is the other plot Determines the people having heart disease have an average – maximum heart rate of 125.



This plot is usually used for plotting longform data where is hue Is the column name which is cholesterol and is used for colour encoding.
This show shows point estimation using scatterplot and provides some indication for uncertainty by the estimate using some error bars.

3.3 Data Pre-processing:

Since there are non null values present all over the Dataset Which certainly does not include much of pre-processing.
Since the machine learning model do not understand categorical values which leads to conversion of all the categorical values into numerical format.

In order to train the machine learning model we need a big chunks of data and typically data items stored in storage system like file or a sql system but whatever data present inside those Database containers it is difficult to pass those raw data into machine learning models without pre-processing it. Data has to be refined or pre-processed in order to pass through machine learning models for training purpose So that machine learning model can easily understand and give the best result on the data.
Data pre-processing involves multiple process inside it starting from loading data into machine learning program, handling the missing value, scaling of data with the help of standardisation and normalisation, Splitting the data into train and test set so that we can past train set for training purpose and we can use test data to validate our machine learning model.

3.3.1 Storing in X and y:

X,y=heart.loc[:,:'ST_Slope'],heart.loc[:,'HeartDisease']

In the above syntax we are passing all the rows and feature variables starting from age to ST-slope column in X and storing all the rows of heart disease column in y. Also splitting dataset into 2 data frame namely X,y.

3.3.2 Encoding of Categorical columns into Numerical Columns:

There are two types of categorical variables which are ordinal and nominal categorical variables. Example of ordinal categorical variables Or rank of students in class, rank of educational structure. When we talk about ordinal categorical and coding we specifically topic about label encoding whereas in the case of nominal category the rank does not matters example like gender: male or female.
So whenever given a problem statement the first and foremost part is to initialise them into categories whether it is an ordinal or nominal category.

from pre-processing we are importing label_encoder

label_encoder = preprocessing.LabelEncoder() #whatever the label, encode it and store in label_encoder
X['Sex'] = label_encoder.fit_transform(X['Sex']) #transform label sex(M,F) and store it back in X dataframe

```
heart['ST_Slope'].head(5)
```

```
0      Up
1    Flat
2      Up
3    Flat
4      Up
Name: ST_Slope, dtype: object
```

```
#categorical values transformed into numerical form in preprocessed X dataframe
X['ST_Slope']
```

```
0      2
1      1
2      2
3      1
4      2
      ..
913    1
914    1
915    1
916    1
917    2
Name: ST_Slope, Length: 918, dtype: int32
```

3.3.3 Splitting of train and test data:

In this we split the original data into training data and testing data which is called as train test and split. Once we split our original data into training data and testing data we will further process with feeding machine learning model with training data. There are several machine learning models where training data would be used to train our model so that our model can find the pattern and learn from this training data. Once it has learnt from the training data the model will be evaluated and this evaluation will be based on test data where the evaluation will be finding about how the model is performing and what is the accuracy score of the model.

X_train,X_test,y_train,y_test=train_test_split(X,y,random_state=10,test_size=0.3,shuffle=True)

So x_train with its label or target variable y_train will go for training.
For testing we will test on x_test data and compare results with y_test data.
In random state it will randomly get splited.
In test size from whole dataset 30% of data for testing and rest 70% for training.

Shuffling includes shuffling of data in unordered form. splitting the X,y dataframe into 4 dataframe: X_train,X_test,y_train,y_test.

train_set_x shape: (642, 11) -> contain 70% of data
train_set_y shape: (642,) -> contain 70% of data but contain only 1 column
test_set_x shape: (276, 11) -> contain 30% of data
test_set_y shape: (276,) ->contain 30% of data but contain only 1 column

3.3.4 Scaling of Data:

Feature scaling is a technique in which we try to bring all the features on the same scale.

Scaling means reducing the distance between values.

	Age	RestingBP	Cholesterol	FastingBS	MaxHR	Oldpeak	HeartDisease
count	918.000000	918.000000	918.000000	918.000000	918.000000	918.000000	918.000000
mean	53.510893	132.396514	198.799564	0.233115	136.809368	0.887364	0.553377
std	9.432617	18.514154	109.384145	0.423046	25.460334	1.066570	0.497414
min	28.000000	0.000000	0.000000	0.000000	60.000000	-2.600000	0.000000
25%	47.000000	120.000000	173.250000	0.000000	120.000000	0.000000	0.000000
50%	54.000000	130.000000	223.000000	0.000000	138.000000	0.600000	1.000000
75%	60.000000	140.000000	267.000000	0.000000	156.000000	1.500000	1.000000
max	77.000000	200.000000	603.000000	1.000000	202.000000	6.200000	1.000000

In the above tabular data when observing cholesterol column where on an average people are having 198 cholesterol level and maximum someone is having 603 cholesterol level which makes it very difficult for some models to convert this data because of very high distance. Example of search models are neural networks, KNN, SVM where distance is required. So while scaling of data we don’t want your model to know about test data because it might cause overfitting.

3.4 Machine Learning Modelling:

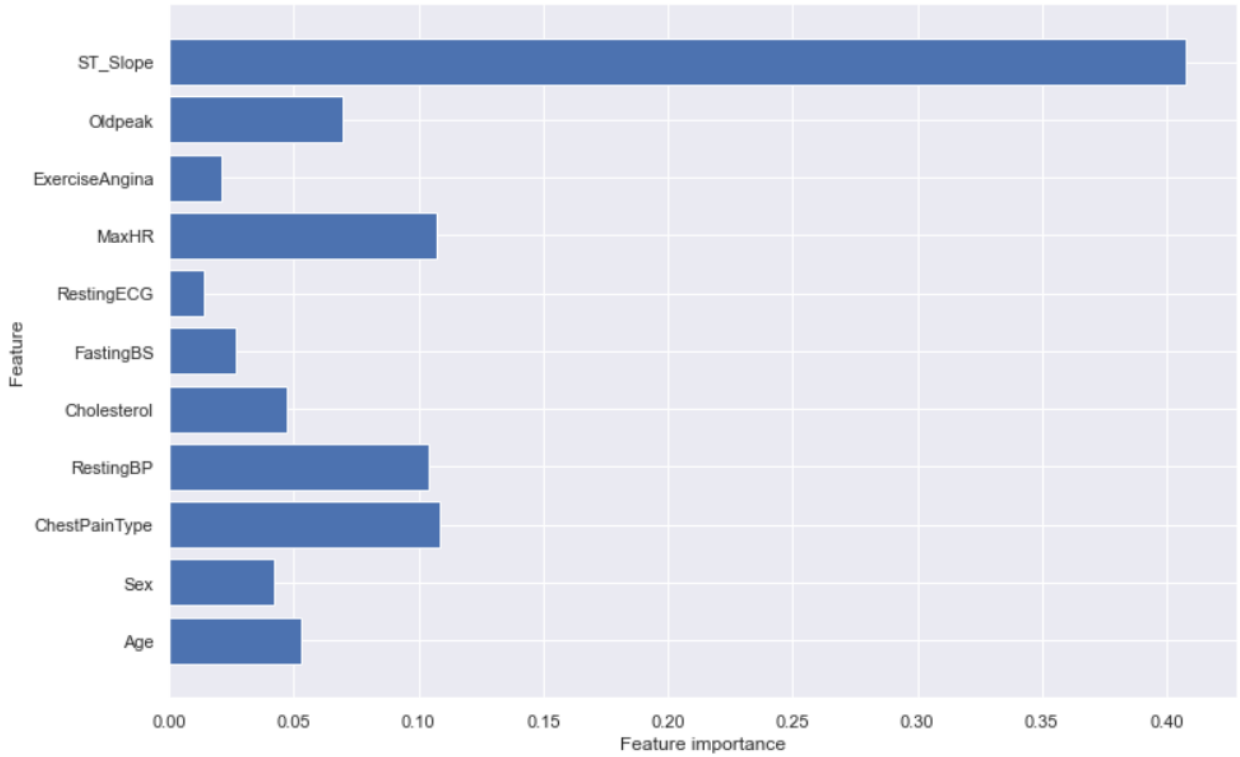
3.4.1 Decision Tree:

prediction=dt.predict(X_test) #predict on test data where X_test is 30% of data
accuracy_dt=accuracy_score(y_test,prediction) * 100 #compare the result with y_test

```
accuracy_dt
```

```
77.53623188405797
```

3.4.2 Feature Importance in Decision Trees:



The above plot determines which are the most important features from all the feature columns present in the data set. Which clearly indicates that ST_slope is very much important feature for a data set and for the modelling and then maximum heart rate and chest pain type. When compared to another scenarios when it comes to heart attack for example the age factor will be much more higher but for a study purpose we are considering heart disease. Due to which ST_Slope considered to be an important feature.

3.5 Ensemble Methods:

3.5.1 Bagging:- Random Forest:

Bagging is generally a way to learn the pattern from the given data. In this technique of bagging where bags of data is been created from actual data.

```
accuracy_rf
```

```
84.78260869565217
```

3.5.2 Boosting:- XG-BOOST:

```
accuracy_xg
```

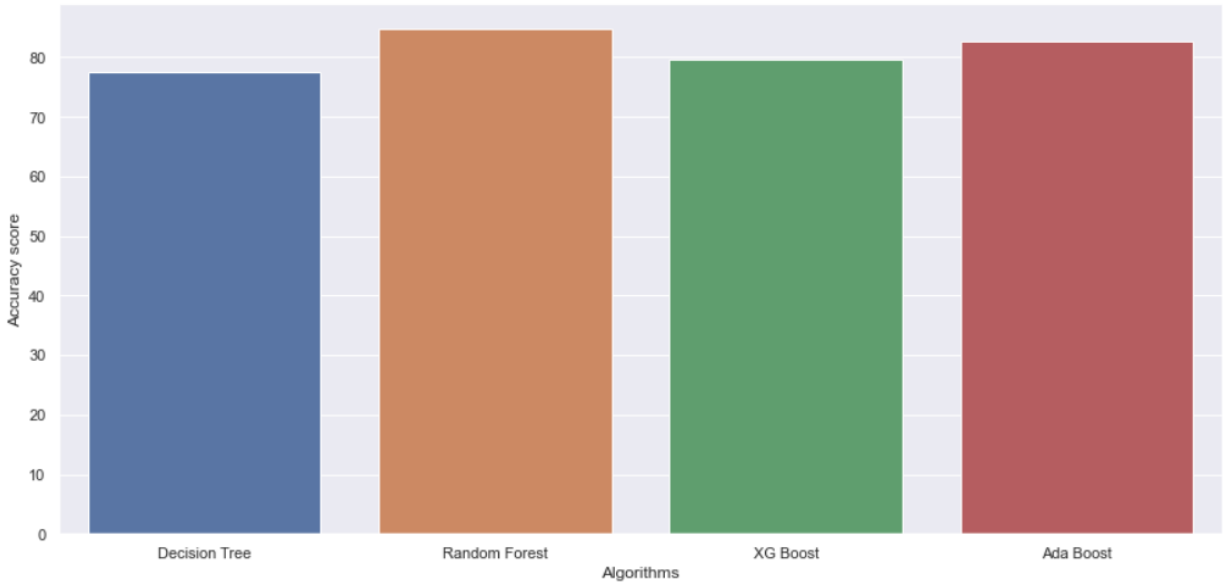
```
79.71014492753623
```

3.5.3 Ada-Boost Classifier:

```
accuracy_abc
```

```
82.6086956521739
```

3.6 The Comparison Of Models:



Comparing all the algorithms where Random Forest gives the highest accuracy of 84.78%

We considered predicting classification problem in heart disease data set using various types of models to classify the heart disease predictions in respect to target variable to determine whether the person is affected by heart disease or not.
After reviewing the accuracy level of various model which clearly determines that random Forest has the highest accuracy of prediction when it comes to this classification problem.

4. **Interpret or Deployment:**

What is deployment?

Deployment is the process in which a piece of code is process some request. For example we train a machine learning model and then we host it on the server, And once the model has been hosted the user can interact with the model by requesting some instruction and model will process those instruction and response to the user.

In our example considering heart failure prediction where model like random forest is requesting some input from the user which are likely age sex cholesterol maximum heart rate slope type etc After processing the inputs from the user the model will successfully predict whether the user with following data is affected by heart disease or not.

What is client-server architecture?

When end user interacts with any application which consist of processing the user credentials which are then rectified at the backend which is called as server Is basically termed as client/server architecture where request and response are been performed.

Considering an example of heart failure prediction where the user interface is generally designed using flask API and The application is locally hosted where the user inputs will act as a client And the host which is requesting the user for the input acts as a server Which further initialise those data points using random forest model and predict the value on the output screen.

What is web application?

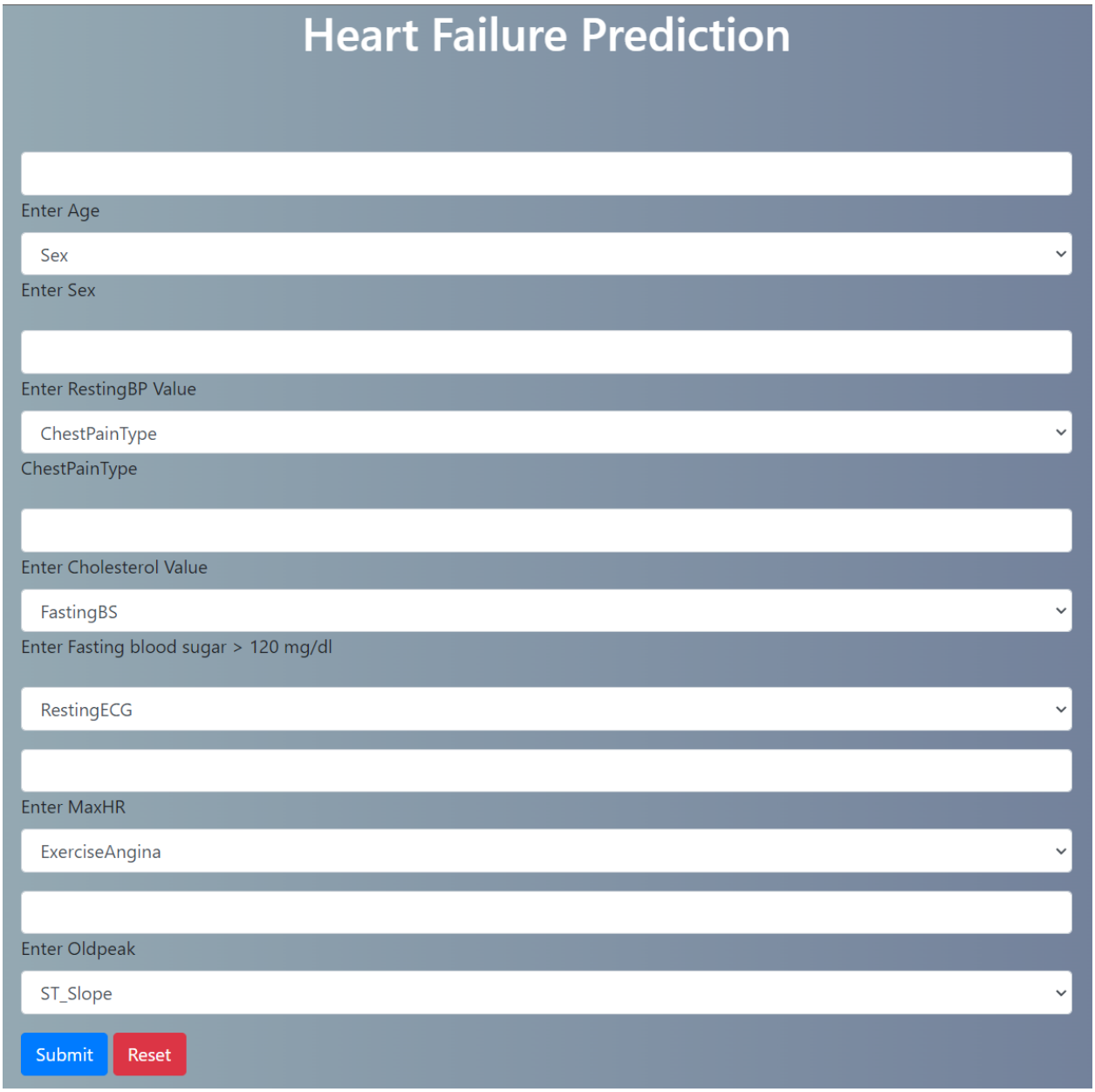
All the application which are running on world wide web are termed as web application. All those application or hosted on internet. The beneficial part of web application is it is accessible to anyone have access to internet and the web address.

What is serialisation?

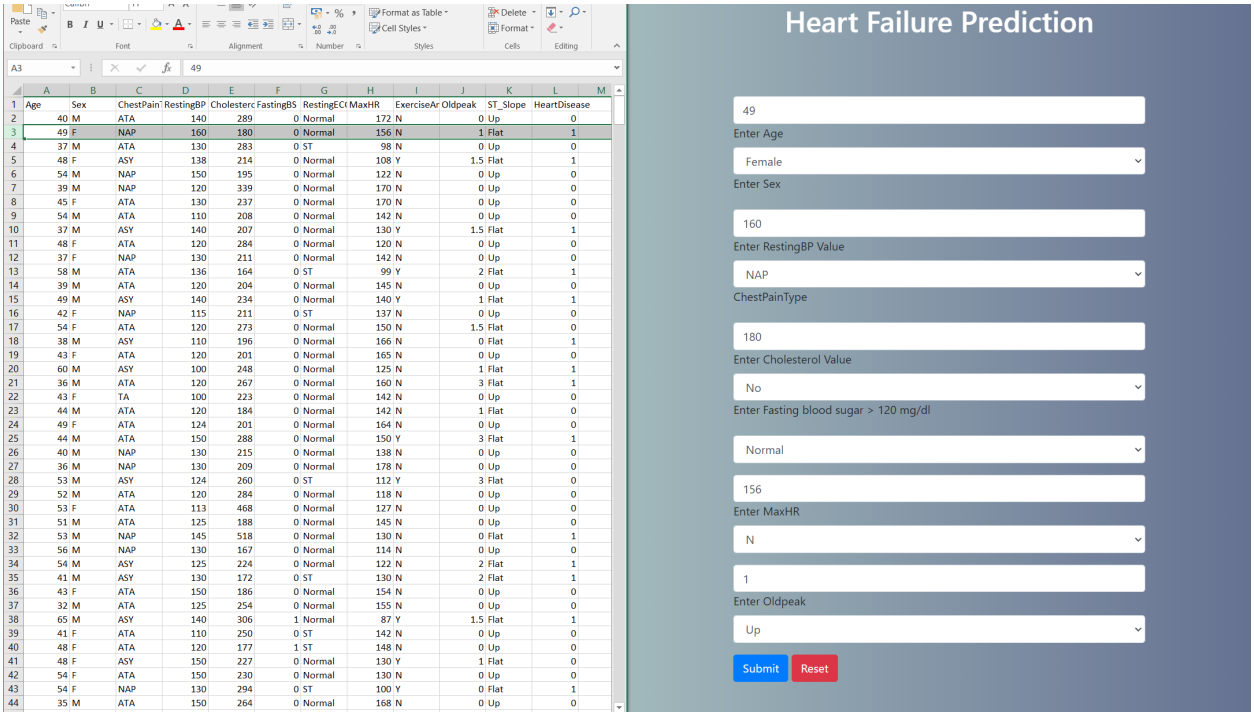
Serialisation is of Python, Java or any programming concept. In serialisation any object can be converted into form which can be saved on disk. Consider a model which is in the form of Y=mx+c on disk drive now this form is valid until the system does not shut down. With the help of serialisation it is efficient enough to save your model on disk drive and reuse it whenever needed.

Understanding flask: Flask is basically a framework in python used for developing web applications.

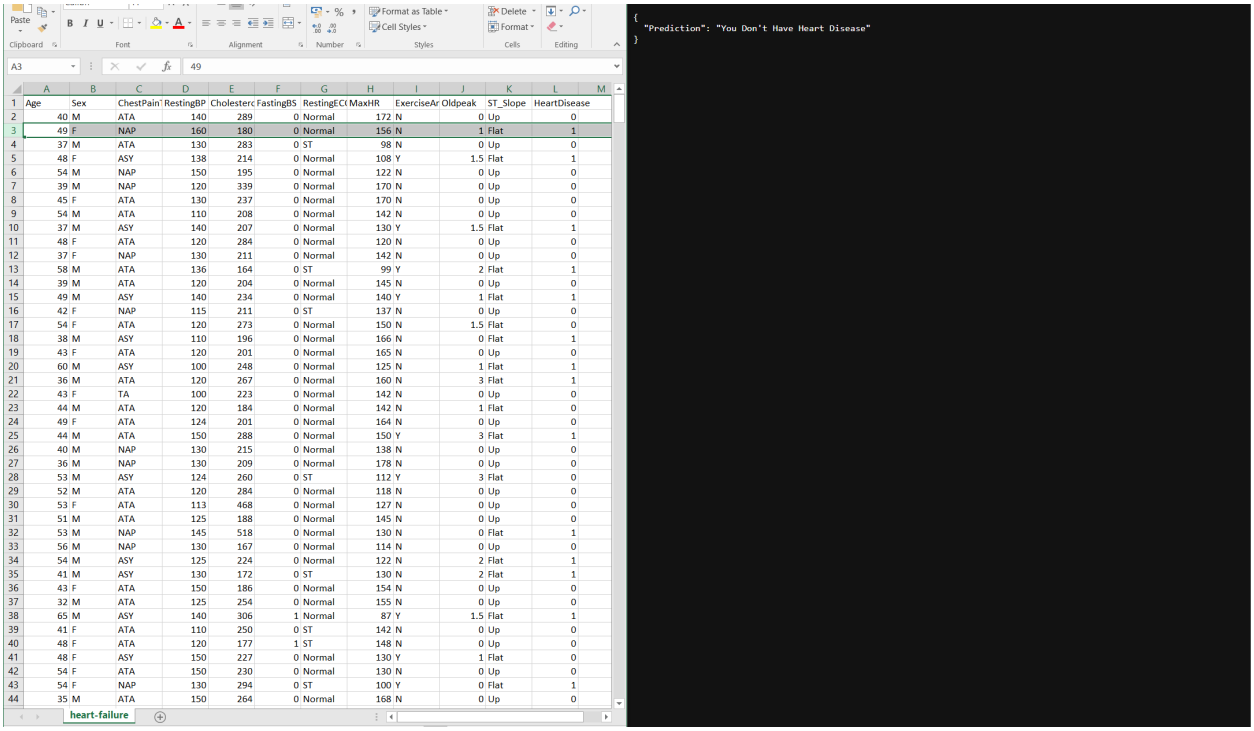
4.1 Model serving using Web Page:



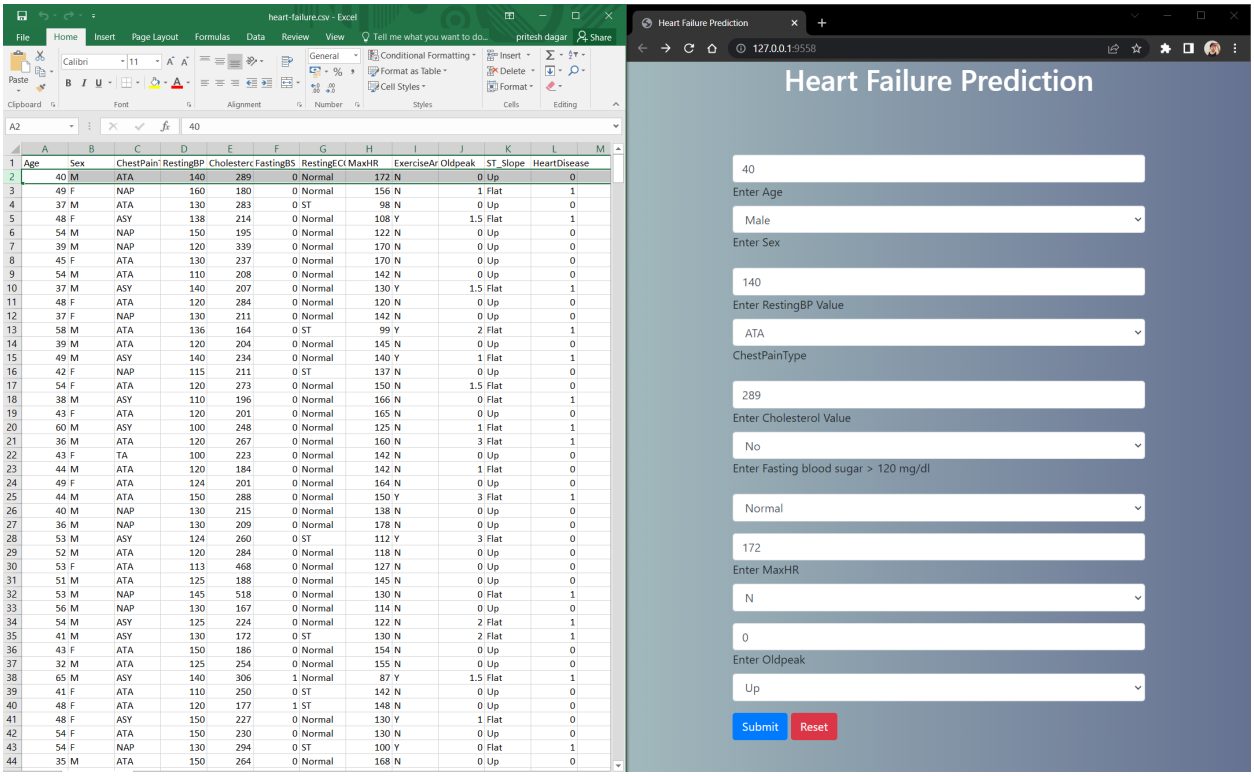
4.2 User Inputs & Results:

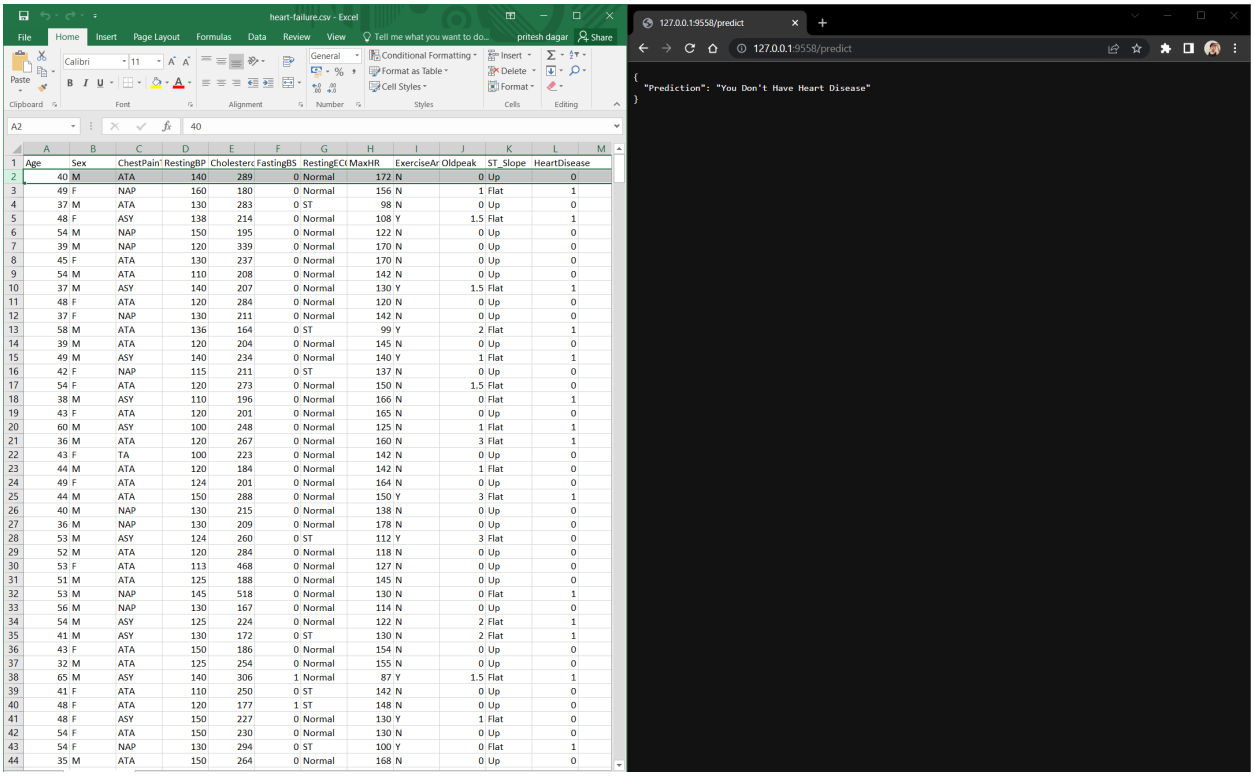


Age	Sex	ChestPain	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAr	Oldpeak	ST_Slope	HeartDisease
40	M	ATA	140	289	0	Normal	172	N	0	Up	0
49	F	NAP	160	180	0	Normal	156	N	1	Flat	1
37	M	ATA	130	283	0	ST	98	N	0	Up	0
48	F	ASY	138	214	0	Normal	108	Y	1.5	Flat	1
54	M	NAP	150	195	0	Normal	122	N	0	Up	0
39	M	NAP	120	339	0	Normal	170	N	0	Up	0
45	F	ATA	130	237	0	Normal	170	N	0	Up	0
54	M	ATA	110	208	0	Normal	142	N	0	Up	0
37	M	ASY	140	207	0	Normal	130	Y	1.5	Flat	1
48	F	ATA	120	284	0	Normal	120	N	0	Up	0
37	F	NAP	130	211	0	Normal	142	N	0	Up	0
58	M	ATA	136	164	0	ST	99	Y	2	Flat	1
39	M	ATA	120	204	0	Normal	145	N	0	Up	0
49	M	ASY	140	234	0	Normal	140	Y	1	Flat	1
42	F	NAP	115	211	0	ST	137	N	0	Up	0
54	F	ATA	120	273	0	Normal	150	N	1.5	Flat	0
38	M	ASY	110	196	0	Normal	166	N	0	Flat	1
43	F	ATA	120	201	0	Normal	165	N	0	Up	0
60	M	ASY	100	248	0	Normal	125	N	1	Flat	1
36	M	ATA	120	267	0	Normal	160	N	3	Flat	1
43	F	TA	100	223	0	Normal	142	N	0	Up	0
44	M	ATA	120	184	0	Normal	142	N	1	Flat	0
49	F	ATA	124	201	0	Normal	164	N	0	Up	0
44	M	ATA	150	288	0	Normal	150	Y	3	Flat	1
40	M	NAP	130	215	0	Normal	138	N	0	Up	0
36	M	NAP	130	209	0	Normal	178	N	0	Up	0
53	M	ASY	124	260	0	ST	112	Y	3	Flat	0
52	M	ATA	120	284	0	Normal	118	N	0	Up	0
53	F	ATA	113	468	0	Normal	127	N	0	Up	0
51	M	ATA	125	188	0	Normal	145	N	0	Up	0
53	M	NAP	145	518	0	Normal	130	N	0	Flat	1
56	M	NAP	130	167	0	Normal	114	N	0	Up	0
54	M	ASY	125	224	0	Normal	122	N	2	Flat	1
41	M	ASY	130	172	0	ST	130	N	2	Flat	1
43	F	ATA	150	186	0	Normal	154	N	0	Up	0
32	M	ATA	125	254	0	Normal	155	N	0	Up	0
65	M	ASY	140	306	1	Normal	87	Y	1.5	Flat	1
41	F	ATA	110	250	0	ST	142	N	0	Up	0
48	F	ATA	120	177	1	ST	148	N	0	Up	0
48	F	ASY	150	227	0	Normal	130	Y	1	Flat	0
54	F	ATA	150	230	0	Normal	130	N	0	Up	0
54	F	NAP	130	294	0	ST	100	Y	0	Flat	1
35	M	ATA	150	264	0	Normal	168	N	0	Up	0



Age	Sex	ChestPain	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAr	Oldpeak	ST_Slope	HeartDisease
40	M	ATA	140	289	0	Normal	172	N	0	Up	0
49	F	NAP	160	180	0	Normal	156	N	1	Flat	1
37	M	ATA	130	283	0	ST	98	N	0	Up	0
48	F	ASY	138	214	0	Normal	108	Y	1.5	Flat	1
54	M	NAP	150	195	0	Normal	122	N	0	Up	0
39	M	NAP	120	339	0	Normal	170	N	0	Up	0
45	F	ATA	130	237	0	Normal	170	N	0	Up	0
54	M	ATA	110	208	0	Normal	142	N	0	Up	0
37	M	ASY	140	207	0	Normal	130	Y	1.5	Flat	1
48	F	ATA	120	284	0	Normal	120	N	0	Up	0
37	F	NAP	130	211	0	Normal	142	N	0	Up	0
58	M	ATA	136	164	0	ST	99	Y	2	Flat	1
39	M	ATA	120	204	0	Normal	145	N	0	Up	0
49	M	ASY	140	234	0	Normal	140	Y	1	Flat	1
42	F	NAP	115	211	0	ST	137	N	0	Up	0
54	F	ATA	120	273	0	Normal	150	N	1.5	Flat	0
38	M	ASY	110	196	0	Normal	166	N	0	Flat	1
43	F	ATA	120	201	0	Normal	165	N	0	Up	0
60	M	ASY	100	248	0	Normal	125	N	1	Flat	1
36	M	ATA	120	267	0	Normal	160	N	3	Flat	1
43	F	TA	100	223	0	Normal	142	N	0	Up	0
44	M	ATA	120	184	0	Normal	142	N	1	Flat	0
49	F	ATA	124	201	0	Normal	164	N	0	Up	0
44	M	ATA	150	288	0	Normal	150	Y	3	Flat	1
40	M	NAP	130	215	0	Normal	138	N	0	Up	0
36	M	NAP	130	209	0	Normal	178	N	0	Up	0
53	M	ASY	124	260	0	ST	112	Y	3	Flat	0
52	M	ATA	120	284	0	Normal	118	N	0	Up	0
53	F	ATA	113	468	0	Normal	127	N	0	Up	0
51	M	ATA	125	188	0	Normal	145	N	0	Up	0
53	M	NAP	145	518	0	Normal	130	N	0	Flat	1
56	M	NAP	130	167	0	Normal	114	N	0	Up	0
54	M	ASY	125	224	0	Normal	122	N	2	Flat	1
41	M	ASY	130	172	0	ST	130	N	2	Flat	1
43	F	ATA	150	186	0	Normal	154	N	0	Up	0
32	M	ATA	125	254	0	Normal	155	N	0	Up	0
65	M	ASY	140	306	1	Normal	87	Y	1.5	Flat	1
41	F	ATA	110	250	0	ST	142	N	0	Up	0
48	F	ATA	120	177	1	ST	148	N	0	Up	0
48	F	ASY	150	227	0	Normal	130	Y	1	Flat	0
54	F	ATA	150	230	0	Normal	130	N	0	Up	0
54	F	NAP	130	294	0	ST	100	Y	0	Flat	1
35	M	ATA	150	264	0	Normal	168	N	0	Up	0


heart-failure.csv - Excel
Heart Failure Prediction
127.0.0.1:9558
Heart Failure Prediction
Enter Age
Enter Sex
Male
Enter RestingBP Value
ChestPainType
ATA
Enter Cholesterol Value
Enter Fasting blood sugar > 120 mg/dl
No
Normal
Enter MaxHR
N
Enter Oldpeak
Up
Submit
Reset


heart-failure.csv - Excel
127.0.0.1:9558/predict
{
"Prediction": "You Don't Have Heart Disease"
}

4.3 Self assessment:

4.3.1 Strengths:

The strength of this project is to help human life to prevent from heart disease where a system is been placed to access any time from anywhere and which is cost-effective.
The model is running behind the prototype defines the highest accuracy like a random forest have been trained and tested on the given data set, which will take any user input and will analyse the features with respect to trained and test data set and which will help to predict whether the person is having a heart disease or not.
Since most of the models were inherited through library named as Scikit learn which helps the model to work at its best without making certain human error.
Exploratory data analysis Have been done on the given data set which is further down pre processed and visualised in various plots which gives better understanding for a viewer in order to understand which attribute feature plays what role in the given data set.

4.3.2 Weaknesses:

When it comes to weakness where I have implemented my own algorithm for bagging classification and regression where the accuracy is 49% which clearly determines the human mistake and error while consideration of the model implementation which further leads to not to use in my final prototype.
Since my data set consist of classification problem where building this bagging algorithm consist of classification and regression both where it cannot only be tested for classification problem but it can also be tested for other type of regression problems.

```
my_accuracy_dt
```

```
49.275362318840585
```

4.3.3 Opportunities:

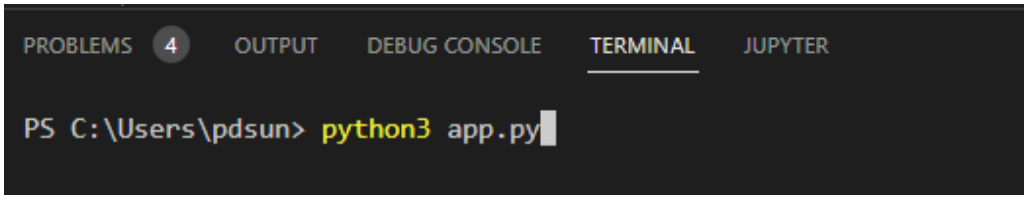
The biggest opportunity for this prototype is when it will actually help human terminology to reduce death rates caused by heart disease.
This prototype model can further be used by professional human care representative or by a normal person who doesn’t know the following features required in order to process the models through this prototype.

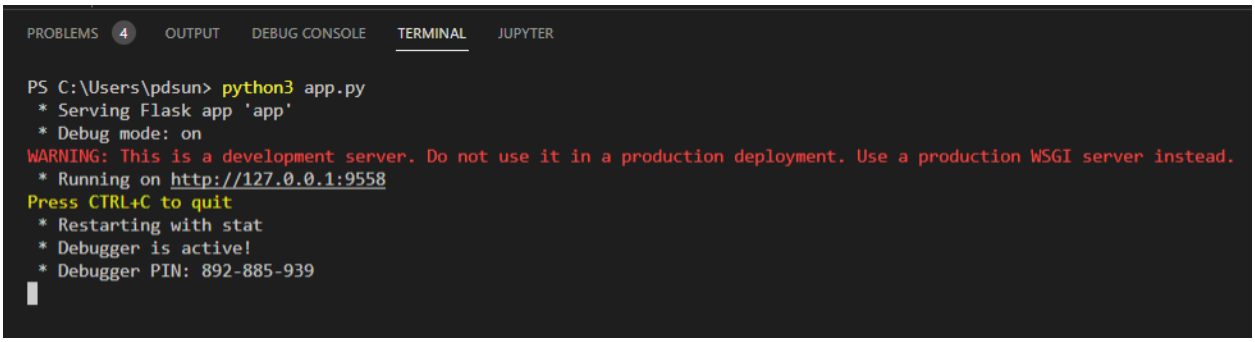
Example if the user do have details about the cholesterol level, maximum heart rate, slope type, age, sex etc Which makes it easy for a user to use this prototype and to predict whether he or she is affected by heart disease or not.

---

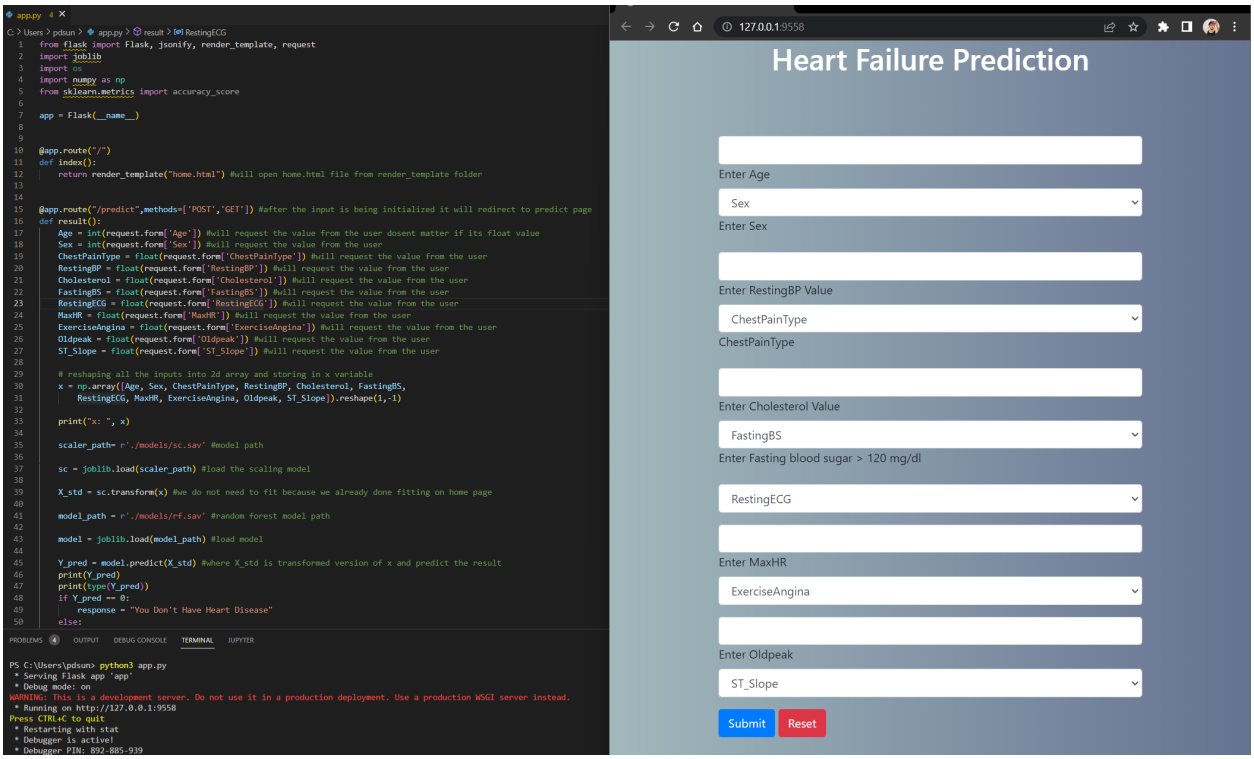
5. How to Use My Project:

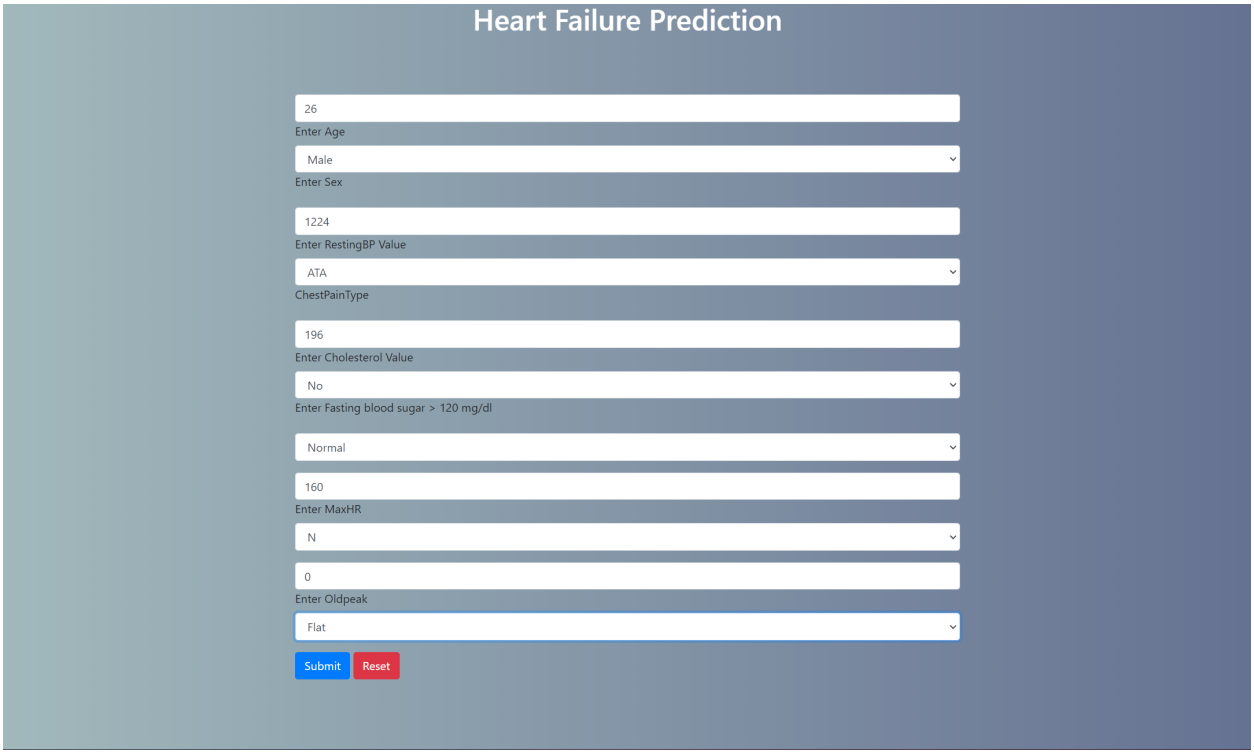
In order to use my project one must fulfil the software requirements.
The file name app.py Can easily be run using terminal or power shell with the command python3 app.py
Which will further generate a local host address which can be opened in the browser in order to access the user interface.



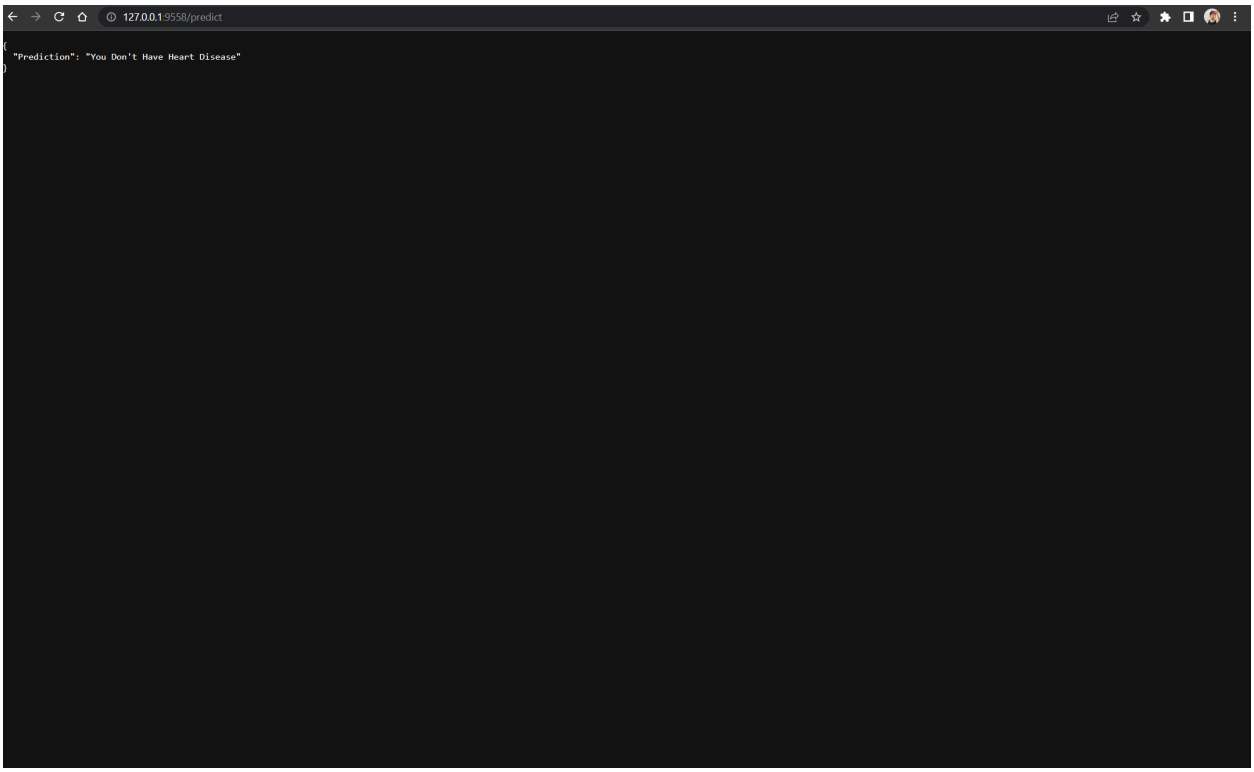


5.1 Demo Project





Random user input in above figure



Outcome of prediction in above screen

5.2 Professional issues

5.2.1 Usability:

The usability issues discovered while implementation of the prototype or wall construction of the Document or some user prompts Which consist of poor grammar mistakes. While computing the output screen With the help of running flask api which took a lot of rendering power Which further leads to low performance over all the system. When coming to the next stage of determining the problem whether it was local or global which was further navigated running the entire application on a local host.
The solution for high consumption of rendering power would be high end System which is likely to encounter less scaling problem.

5.2.2 Plagiarism:

Plagiarism comes with consequences which further leads to ethical and legal terminology.
This makes the student reputation fall down most of the educational institutions have academic integrity committees, plagiarism not just destroys professional reputation but it also destroys academic reputation.

Research about plagiarism is very much important where one must always Tries to produce its own content rather than copying it from available sources.

5.2.3 Licensing:

All the work done should be licensed meaning and ethical theory dealing with some particular problem and which is concerned and involved in governing the society seems to belong in most of the papers present.
It is very much necessary in order to protect one's own content and licensing it so that if used without permission can further lead to act in law.

5.2.4 Privacy:

Data privacy plays a vital role all over the internet where companies or trying to take user data to make further business decisions where consumers loses the rights on its own data where in place data privacy is very much important because this can be misused.

5.3 Conclusion:

The Dissertation includes of various machine learning ensemble methods Which is very much profitable for the medical industry. Using various machine learning techniques like decision tree, bagging boosting helps to predict the heart disease.
In this prototype we have constructed ETA and then re-processed the data while converting categorical values into numerical format which makes it more reliable to pass the feature attributes in machine learning model. The first algorithm decision tree had a good accuracy in the range of 70 to 80. Where is further future importance in decision tree have been carried out with the help of that we were able to analyse the most important feature column from the data set. Which will further help to predict higher accuracy.
The next algorithm used was random forest which gave the highest accuracy of 80 to 90%. We used XG boost which also gave the best accuracy but not as compare to random forest and then finally boosting its been done using ada boost classifier which gave the prediction accuracy of 80 to 90% but which was quite close to random forest.
After the models were trained the models were saved and further imported user interface using flask API where the user can interact with user interface providing 13 medical attribute conditions which will predict whether the person is having heart disease or not.
Any contribution to this project in future will be appreciated.

5.4 Future scope:

The introduced prototype can further be used for many medical sectors which can be tuberculosis prediction, lungs failure prediction, blood cancer prediction and many other also this application can further be constructed to be used on various different platforms like android or iOS where the user can just download the app and feed the data and the models will predict whether the user is having heart disease or not.
Due to Data protection it is very likely relevant to use some security measures in future while creating or re-creating this application. This prototype cannot only be used in medical sector but it can also be used for sports data analytics where the user can predict whether his or her favourite football team is going to win or not.

References:

https://towardsdatascience.com/understanding-random-forest-58381e0602d2

https://towardsdatascience.com/from-a-single-decision-tree-to-a-random-forest-b9523be65147

https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction

https://www.kaggle.com/competitions/heart-failure-prediction-uwi

https://www.kaggle.com/code/asifpervezpolok/heart-failure-prediction-with-eda/data

https://www.kaggle.com/code/mohamedzayton/heart-failure-prediction-nn-xgboost

https://www.kaggle.com/code/karnikakapoor/heart-failure-prediction-ann