TUNKU ABDUL RAHMAN UNIVERSITY COLLEGE

BEYOND EDUCATION

BACS2003 ARTIFICIAL INTELLIGENCE

202301 Session, Year 2022/23

Assignment Documentation

Full Name: LOH WEI LUN
Student ID: 22WMR05681
Programme: RSW2
Tutorial Class: G4
Project Title: MACHINE LEARNING(SUPERVISED) AND NATURAL LANGUAGE GENERATION
Module In-Charged: K Nearest Neighbor Classifier

Other team members' data

No	Student Name	Module In Charge
1	LIM MENG LEONG	RANDOM FOREST CLASSIFIER
2	LEE JING JET	SUPPORT VECTOR MACHINE
3		

Lecturer: Dr. Goh Ching Pang	**Tutor : Ho Chuk Fong**

Table of Contents

1. Introduction

1.1. Problem Background

Artificial intelligence adoption has been accelerating over the past few years, and there is no doubt that AI offers a large number of benefits and prospects. Providing relevant or useful information, knowledge, research, and prediction are a few examples. Artificial intelligence and the machine learning algorithms enable prediction tasks such as the prediction of heart disease, which is the subject of our assignment. Our team created a heart disease prediction software primarily because we firmly believe that excellent health is the best investment because it allows you to do anything.

However, Malaysians have a poor level of awareness of heart illnesses. Malaysians don't seem to care about their own health, which may be because they aren't exposed to enough health information. About 80% of Malaysia's elderly may experience chronic health issues like heart attacks, which may cause premature mortality, according to research by Murat N. Through regular medical examinations that help to identify any early indications, one of the greatest methods to prevent heart disease is to do so. However, CodeBlue has conducted research on this issue and from their findings it showed that nearly half of Malaysians are lacking health coverage beyond public care. It is very clear that the number of Malaysians who undergo monthly or yearly medical checkups are significantly lower than our neighboring countries. Our chief statistician Mohd Uzir Mahidin has stated that heart diseases have remained the principal cause of death in Malaysia with an increase of 5.4% from 11.6% in 2000 to 17% in 2020. Therefore, in Malaysia it is now more important than ever to prevent heart diseases.

The high cost of medical care is one of the main reasons Malaysians do not visit the doctor. Since, there are no obvious indicators of a significant health issue, they felt they are in good condition but it's vital to remember that not all medical issues will have readily observable early signs. Our team has developed a heart disease prediction tool that helps users determine whether they are susceptible to heart illnesses or not in the spirit of the adage "prevention is better than cure". Our team has employed a supervised machine learning algorithm in our heart disease prediction programme to determine whether a user was likely to develop heart disease. In order to ensure that people are more aware of this fatal disease, a good data driven approach helps to improve the entire research and prevention process.

A prediction system can be created by analyzing large and complex amounts of data with the aid of machine learning. Age, Sex, Type of chest discomfort, Resting Blood Pressure, Cholesterol, Fasting Blood Sugar, Result of a Resting ECG, Highest Heath Rate attained during exercise, and an older peak heart rate are the factors that be used to classify people who is being at risk for getting heart disease. The 3 different algorithms included Random Forest, K-nearest neighbor and Support Vector Machine(SVM).

1.2. Objectives/Aims

The main objective of this project is to increase the awareness of heart disease among Malaysians because heart disease is one of the principal causes of death in Malaysia. The project allows the users to check if they are suffering from the risk of getting heart disease or not. The notice message will be prompt on showing whether the users are prone to heart disease or not. There is a far greater possibility that they will either be able to reverse their heart condition or take the steps to stop it from worsening.

The heart disease prediction application helps to reduce the amount of cost used by the people to undergo monthly or yearly medical checkups as the users can get a grasp on how healthy their heart is. Although the application can only be taken with a grain of salt, it still lets users get an idea on the condition of their heart. Therefore, it helps the users to plan for their medical checkup to save costs by preventing excessive medical checkups.

The adoption of machine learning for heart disease prediction helps users to save their time by knowing their heart condition without reading any relevant books, browse the internet to further find the symptoms of heart disease. The users can know whether they are prone to have heart disease or not by just need enter relevant details that are necessary into the system.

1.3. Motivation

There is a saying that goes, “no human being is perfect”. Doctors might make the mistakes when they are curing a patient but the mistakes they made may lead to a worse case which is a life lost. For instance, if a person who is healthy was said to have heart disease which was diagnosed wrongly by a doctor, this human error could cause serious complications to a person’s health. With the wrong prescription of medicines, it could take a major toll on the person’s health and might lead to other complications of diseases. The easiest and cheaper solution for this problem is to create a program for avoiding the problem. Therefore, our team has created this application that uses machine learning and natural language as the algorithm to make the tasks automated and provide a highly accurate and time efficient result.

1.4. Timeline/Milestone

Task\Week	1	2	3	4	5	6	7	8	9	10	11	12
Research and choose the suitable topic												
Searching datasets												
Study the implementation of model to Machine Learning System												
Draft Flowchart												
Code implantation of Machine Learning System												
Code testing												
Report Documentation												
Submission												

2. Research Background

2.1. Background of the applications

In recent years, the world has started changing, everythings is connected to a data source and it is digitally recorded. Most of the tasks are being done automatically with the help of machine learning such as health prediction systems, stock price prediction systems and even financial prediction systems. With these prediction systems around the world, it cannot be denied that machine learning has given humans a lot of convenience in life. Machine learning is an algorithm that can automatically improve itself over time without requiring human programmers to feed in additional information. This is because after analyzing large amounts of data, Machine learning will start to modify itself in response to the data's quality and this helps to increase the precision and accuracy of the entire system overtime.

By adding and feeding Machine Learning with large amounts of medical history data, it will help to predict whether the person is prone to heart disease or not. This is because it will recognize whether the individual is having any symptoms of heart disease such as high blood pressure, old age or different levels of chest pain. Before planning for a medical checkup, users can use our application to get a grasp on what is the condition of their heart. This helps the users to save money and time.

The application utilizes the machine learning algorithms to predict whether a user is prone to heart disease. Our team has chosen 4 types of algorithms which are Random Forest, K-Nearest Neighbor(KNN), Support Vector Machine (SVM) and Stacking CV technique with the Extreme Gradient Boost algorithm as the basis of models for improving the performance of models. The reason for implementing multiple algorithms to use is because it helps to increase the accuracy and precision of the data being fed. Before any data is being sent to the models, we will be converting all the raw data into an intelligible format which is during the data preprocessing stage. During the data preprocessing stage, missing values, cleaning of data and also normalization will be done. Therefore, the accuracy and performance of our models can be easily evaluated through a variety of performance metrics.

2.2. Analysis of selected tool with any other relevant tools

Tools comparison	**Remark**	**Jupyter Notebook**	**Excel**
Type of license and open source license	State all types of license	Release under the modifier DSB 3-Clause "New" or "Revised" License	Microsoft Office License Required
Year founded	When is this tool being introduced?	First Release in 2014	1985
Founding company	Owner	Created by a team of developers and researchers from a variety of academic and industry institutions, including the University of California, Berkeley, Cal Poly San Luis Obispo, and Continuum Analytics	Microsoft Corporation
License Pricing	Compare the prices if the license is used for development and business/commercialization	Free	Office Home & Business 2021 $249.99 365 Personal Plan $69.99/year 365 Family Plan $99.99/year
Supported features	What features that it offers?	Support a variety of programming languages, including Python, R, Julia etc. Inline plotting, markdown cells for rich text formatting and support for interactive widgets.	● Inserting pivot table ● Sorting of tabulated data ● Visualize the data
Common applications	In what areas this tool is usually used?	Data analysis, scientific computing, and machine learning as well as for educational purposes.	Perform data analysis

Customer support	How the customer support is given, e.g. proprietary, online community, etc.	A large and active community of developers and users with online documentation, forums and resources available for troubleshooting and assistance	Microsoft support
Limitations	The drawbacks of the software	• Not be well suited for large scale production deployments or highly specialized use cases. Requires some technical expertise to set up and configure for certain applications.	• Hard to detect fraud / corruption

2.3. Justify why the selected tool is suitable

Tools	Reason
Jupyter notebook	● Support Python Programming Language which is used by our team to develop this machine learning project ● Allows for interactive development and testing of code, making it easy to experiment with different models and techniques ● Support inline plotting which allows for visualization of data and model performance ● Support the use of markdown cells, which allow for the inclusion of rich text and documentation within the project ● Allows the use of interactive widgets, which can provide an intuitive and user-friendly interface for inputting data and exploring model results ● Open Source and has large and active community providing access to resources and support for the project

3. Methodology

3.1. Description of dataset

heart.csv 126.92 kB

Detail Compact Column 10 of 12 columns

About this file

The data contains 3932 observations with 12 attributes.

# Age age	Sex sex	ChestPainType chestpain	# RestingBP restbp	# Cholesterol choles
30 – 77	1 72% 0 28%	3 34% 2 27% Other (1538) 39%	0 – 200	0 – 603
40	1	1	140	289
49	0	2	160	180
37	1	1	130	283
48	0	3	138	214
54	1	2	150	195
39	1	2	120	339
45	0	1	130	237
54	1	1	110	208
37	1	3	140	207
48	0	1	120	284
37	0	2	130	211
58	1	1	136	164
39	1	1	120	204
49	1	3	140	234

The heart.csv dataset was obtained through kaggle, This data is being used to build a heart disease prediction system. There are a total of 3932 rows and 12 columns in this dataset. The 12 columns are age, sex, chest pain, restbp, choles, fastbs, restecg, maxhr, exagina, oldpeak, stslope and target respectively.

Column’s Name	Description
Age	Data under 28 to 77 years old
Sex	1: Male 0: Female
Chestpain	0: Typical Angina 1: Atypical Angina 2: Non-Aginal Pain 3: Asymptomatic
restbp	A range of up to 200
choles	A range of up to 600
fastbs	0: No 1: Yes
maxhr	A range from 60 to 200.
exagina	0: No 1: Yes
oldpeak	A range from 2.6 to 6.2
stslope	0: Upsloping = better heart rate with exercise(uncommon) 1: Flatsloping = minimal change(typical healthy heart) 2: Downsloping = signs of unhealthy heart
target	0:No Heart Disease 1: Heart Disease

3.2. Applications of the algorithm(s)

3.2.1 Data Representation

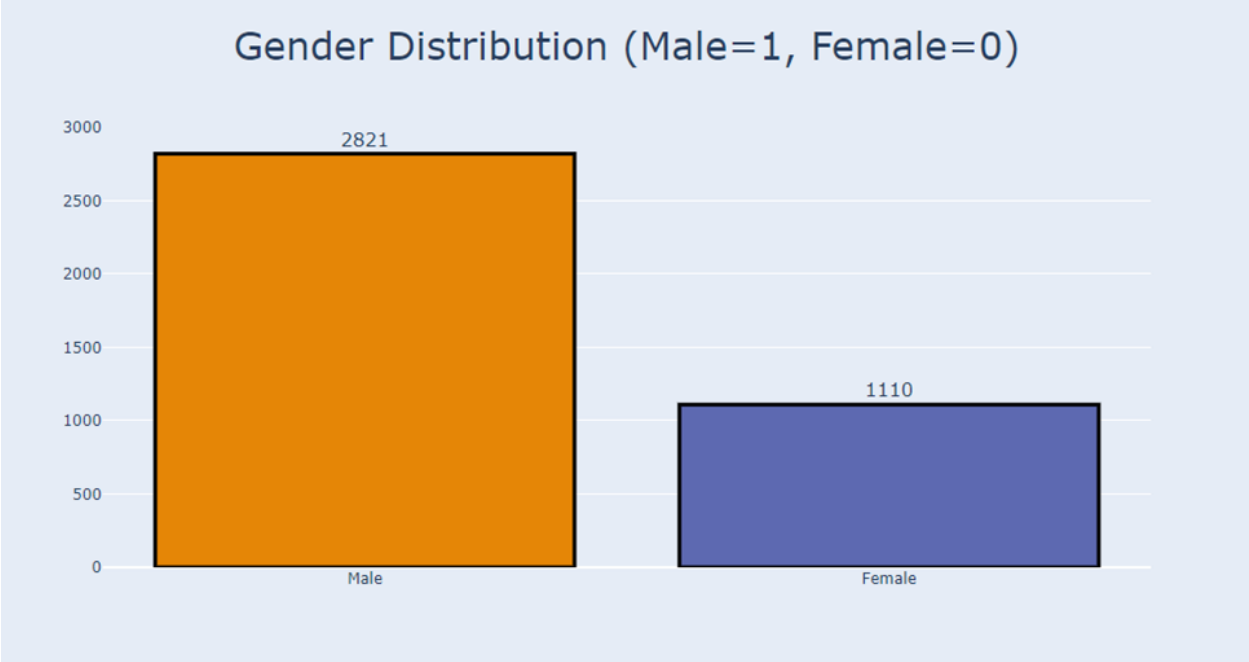

Figure 1: Bar Chart of the Gender Distribution

Figure above shows the bar chart of the gender distribution inside the dataset. The orange bar represents the number of males and the purple bar represents the number of females in the dataset. The total number of males is 2821 and the total number of females is 1110.

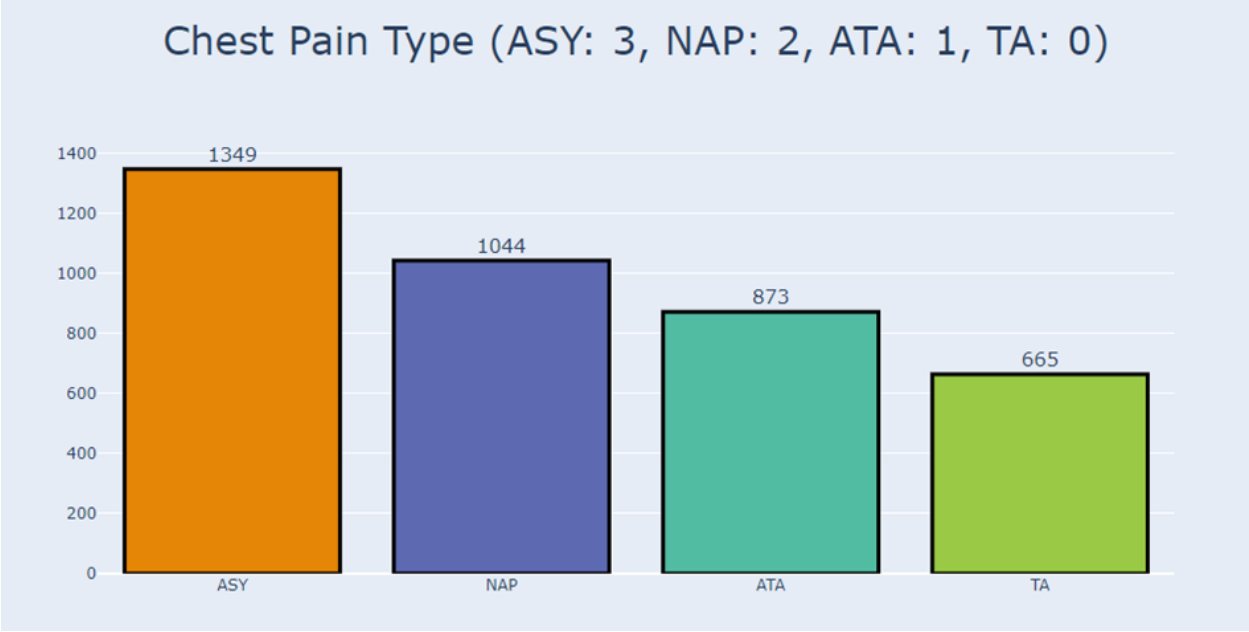

Figure 2: Bar Chart of the Chest Pain on different types

Above figure shows the bar chart of Chest Pain different types.
Orange Bar: Asymptomatic(ASY). Total = 1349
Purple Bar: Non-Anginal Pain(NAP). Total = 1044
Green Bar: Atypical Angina(ATA). Total = 873
LIght Green Bar: Typical Angina (TA). Total = 665

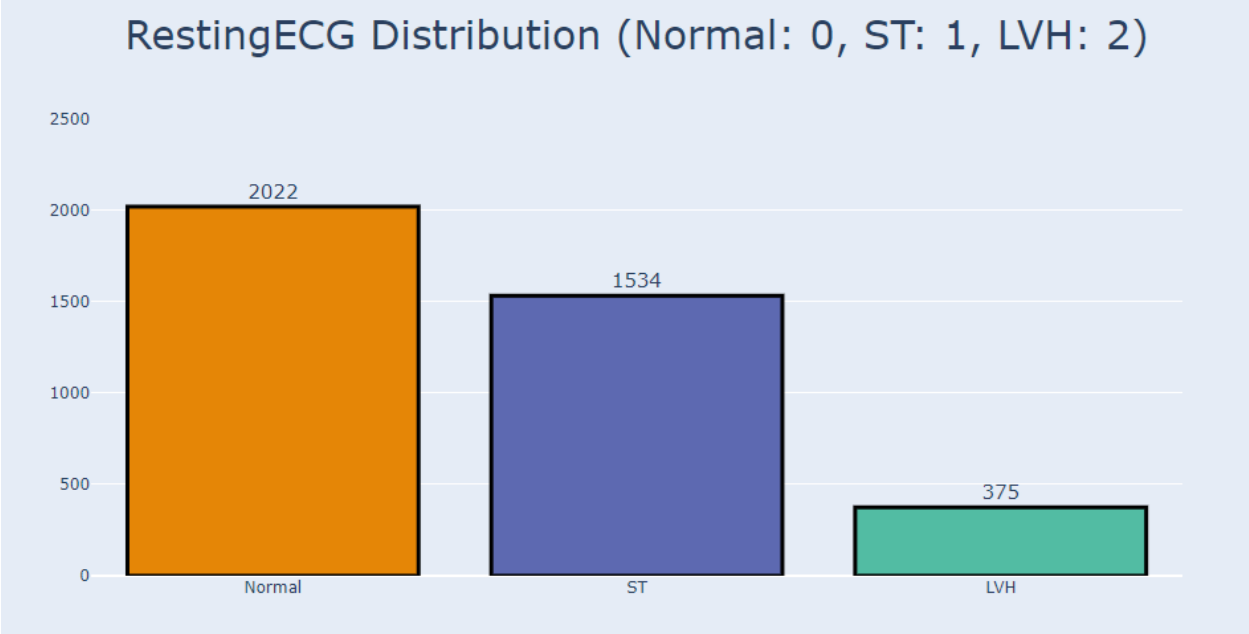

Figure 3: Bar Chart of the Resting ECG Distribution

Above figure shows the bar chart of the number of people who have the different types of resting electrocardiograms.
Orange Bar: Normal of resting electrocardiograms(Normal). Total = 2022
Purple Bar: ST-T Wave Abnormality(ST). Total = 1534
Green Bar: Left Ventricular Hypertrophy(LVH). Total = 375

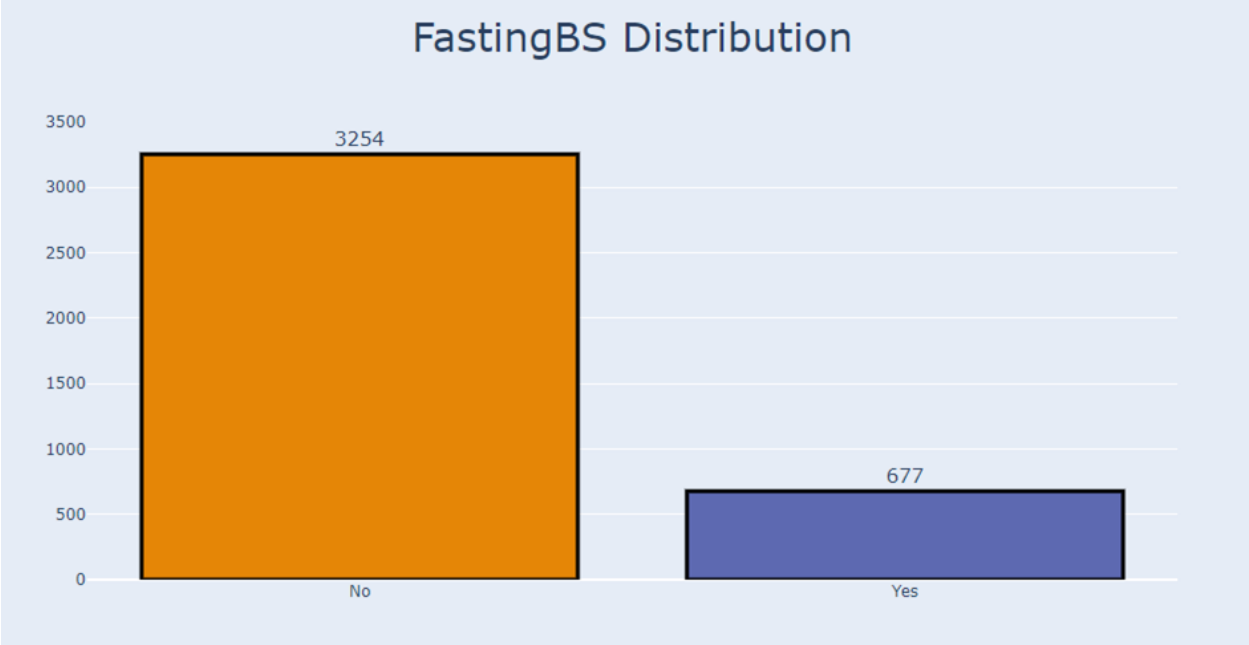

Figure 4: Bar Chart of the FastingBS Distribution

Above figure shows the bar chart of the fasting blood sugar distribution in the dataset.
Orange Bar: who have a blood sugar of less than 120 mg/dl. Total = 3254
Purple Bar: who have a blood sugar of more than 120 mg/dl. Total = 677

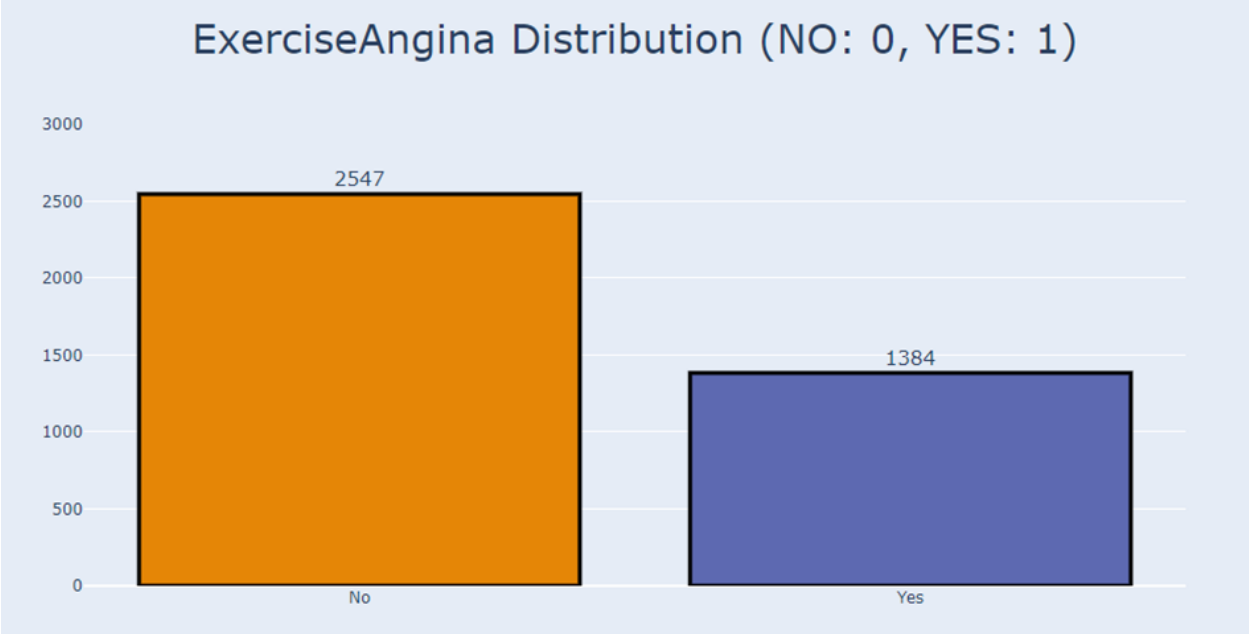

Figure 5: Bar Chart of the ExerciseAngina Distribution

Above figure shows the exercise induced angina distribution in the dataset.
Orange Bar: who do not have exercise induced angina. Total = 2574
Purple Bar: who have exercise induced angina. Total =1384

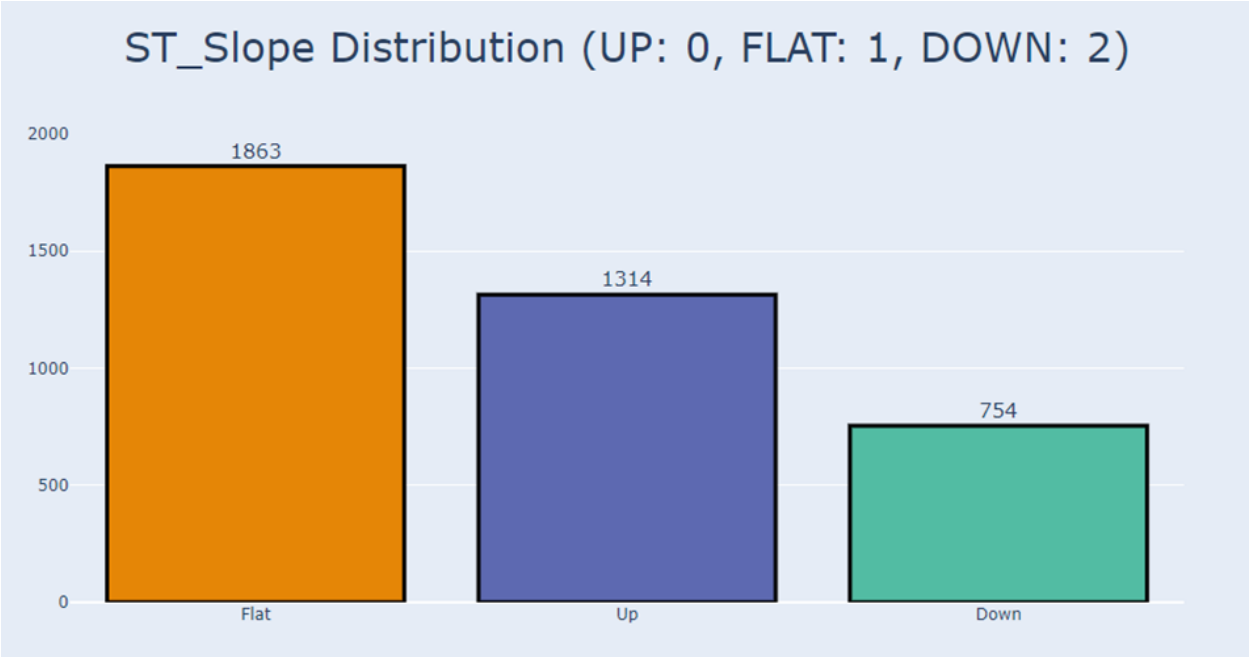

Figure 6: Bar Chart of the ST_Slop Distribution

Above figure shows the number of individuals who have different types of ST Slope in the dataset.
Orange Bar: who have a flat slope. Total = 1863
Purple Bar: who have an up slope. Total = 1314
Green Bar: who have a down slope. Total = 754

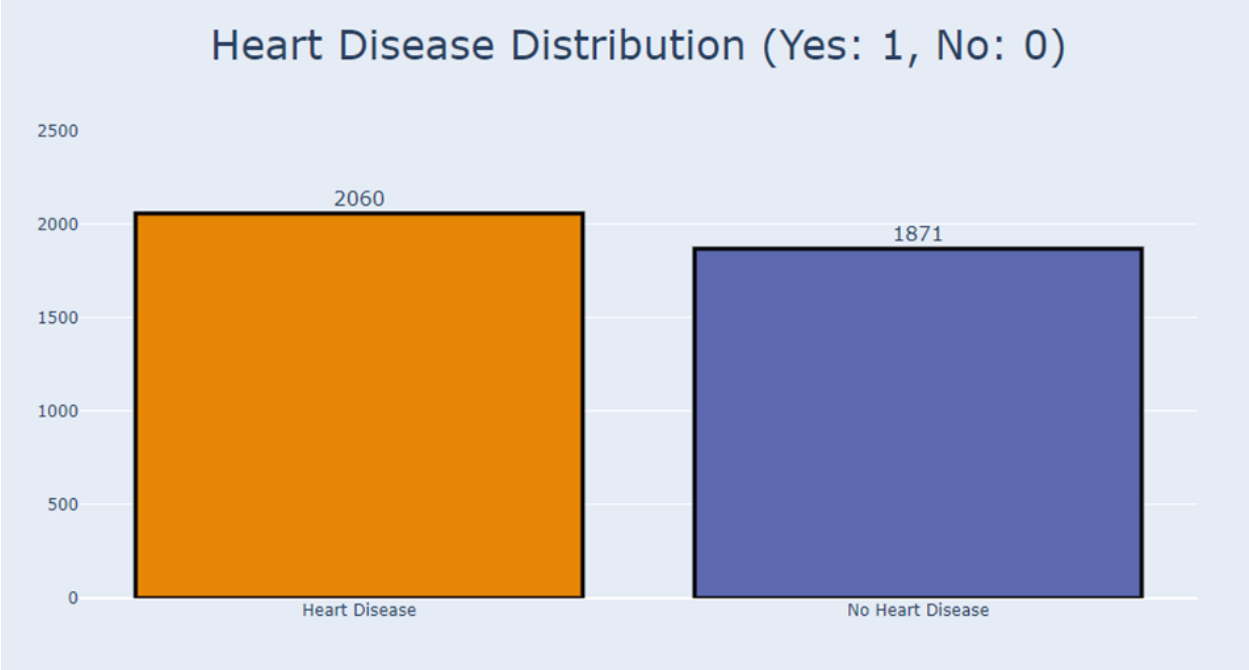

Figure 7: Bar Chart of the Heart Disease Distribution

Above figure shows the number of individuals who have different types of ST Slope in the dataset.
Orange Bar: who has heart disease. Total = 2060
Purple Bar: who does not have heart disease. Total = 1871

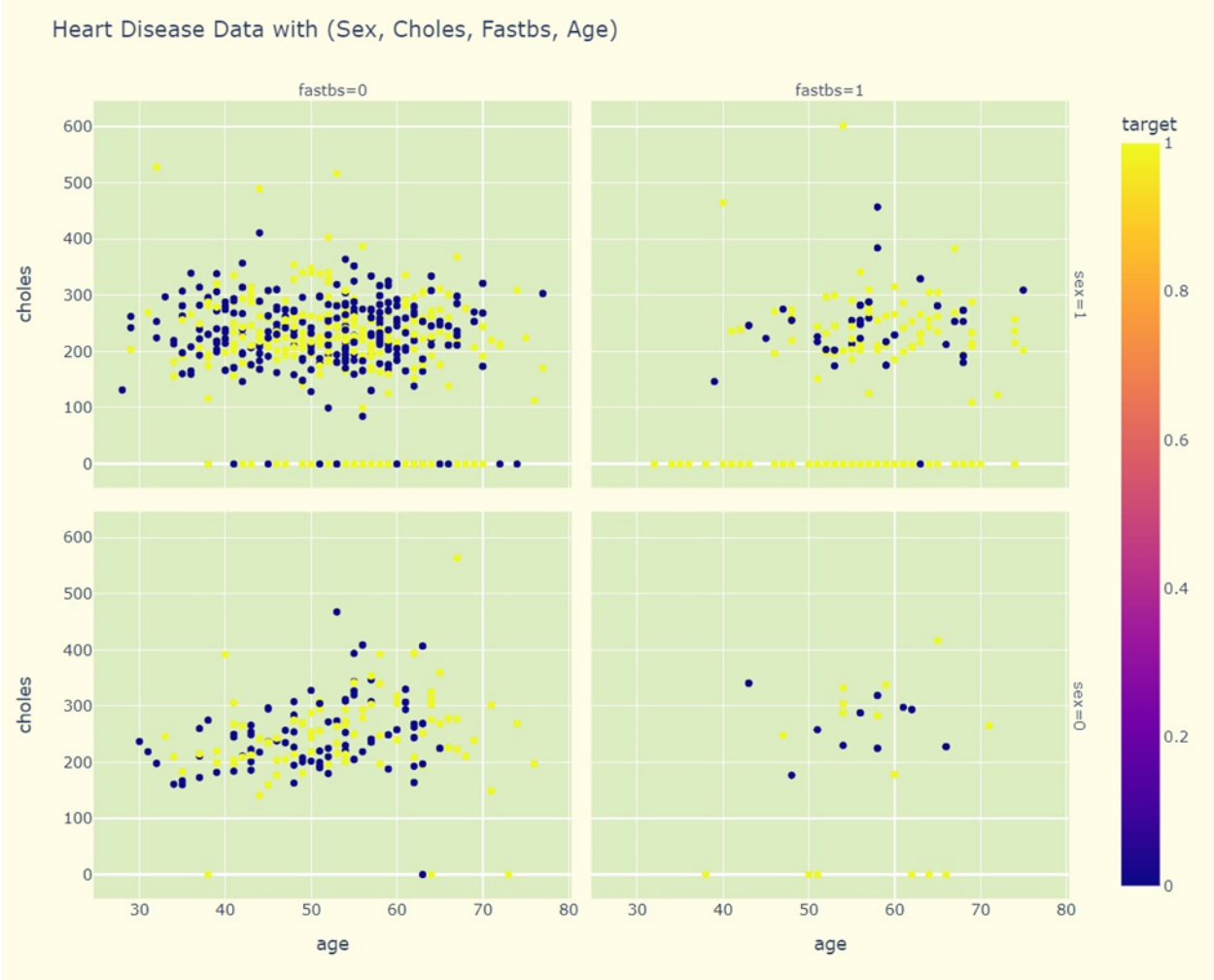

Figure 8: Scatter Plot Correlation Graph

Above figure shows the scatter plot graph which indicates the correlation between sex, choles, fastbs and age.
Blue Color Dots: who do not have heart disease
Yellow Color Dots: who have the heart disease
Top Left of the graph: who have a fasting blood sugar of less than or equal to 120 mg/dl.
Top Right and Bottom Right of the graph: shows both sex with fasting blood sugar of more than 120 mg/dl.
Above graph shows that male who are in the age of 40 to 70 and have a cholesterol level between 150 to 300 mg/dl are more prone to heart disease.
The females who are in the age of 45 to 60 and cholesterol level between 150 to 300 mg/dl are more prone to heart disease.

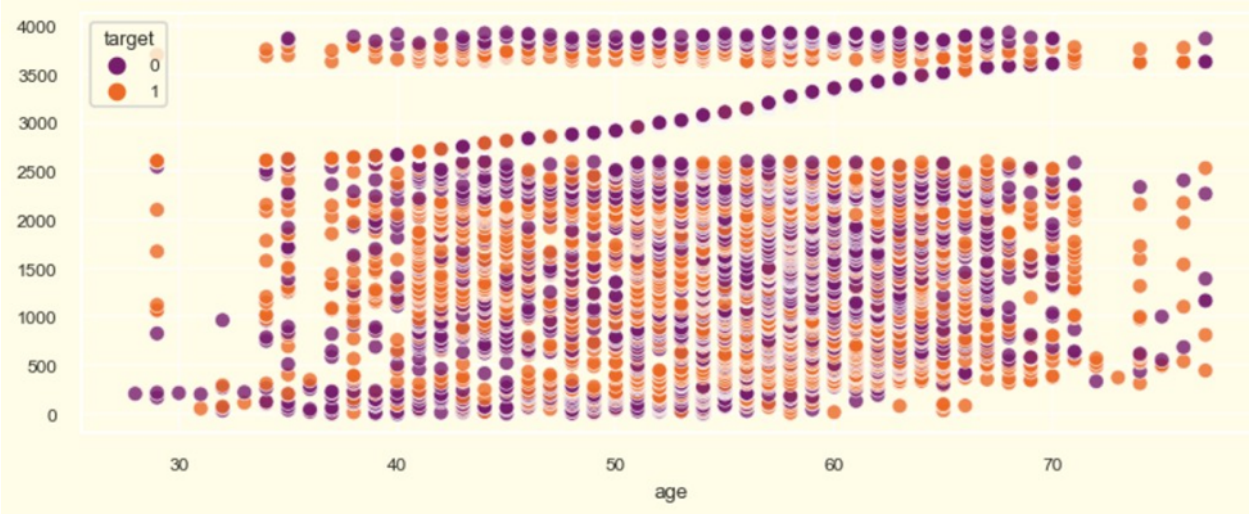

Figure 9: Scatter Plot of Age with target (heart disease)

Above figure shows the scatter plot of individuals whose age is between 28 to 77 and it shows which individual is more likely to have heart disease.
Those aged 30 to 70 are more prone to heart disease because the orange dots (1)representing those who have heart disease are frequently shown between age 30 to 70.

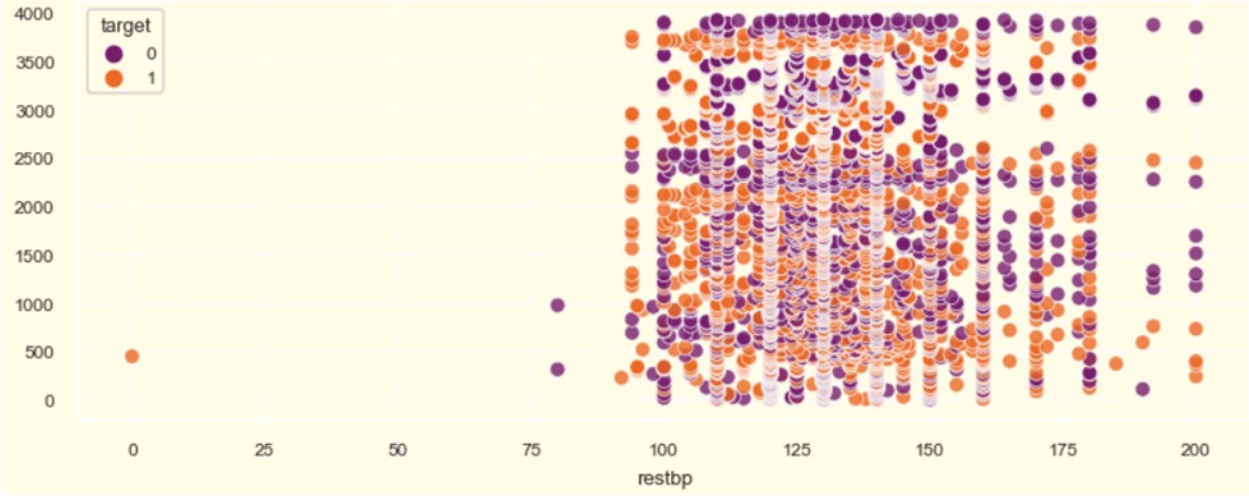

Figure 10: Scatter Plot of Resting Blood Pressure with target(heart disease)

Above figure shows the scatter plot of the resting blood pressure that determines whether an individual is prone to heart disease or not.
From the above figure, most of the individuals have a resting blood pressure between 100 to 180.

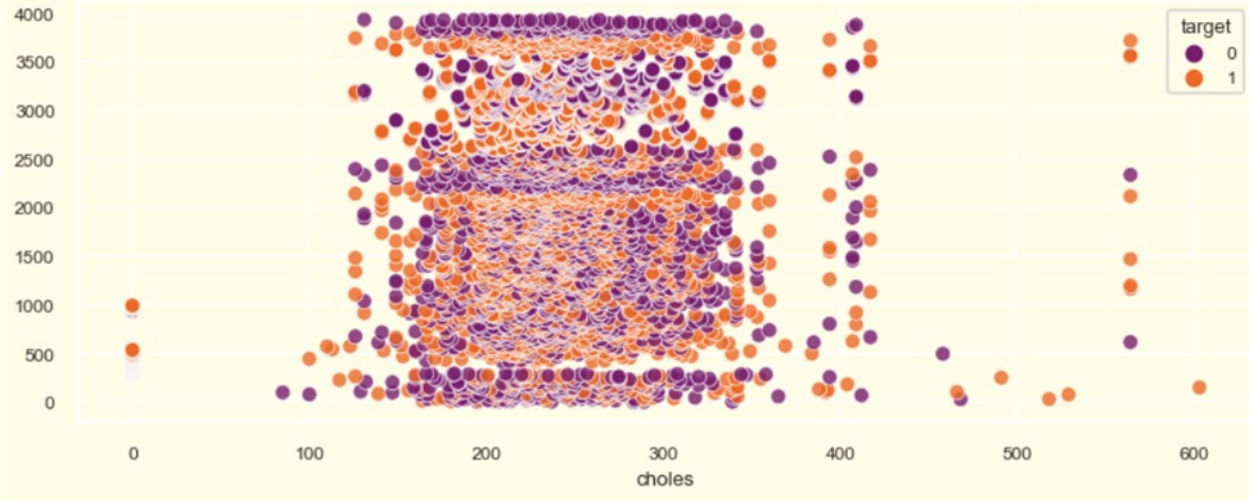

Figure 11: Scatter Plot of Cholesterol with target(heart disease)

Above figure shows the scatter plot of the cholesterol that determines whether an individual is prone to heart disease or not.
From the above figure, most of the individuals have a cholesterol between 150 to 350.

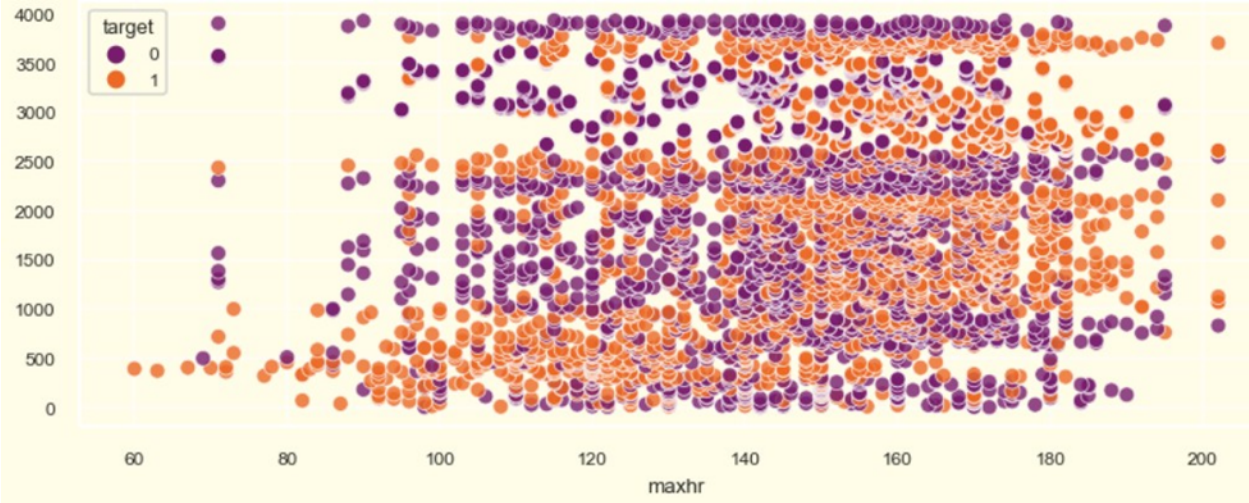

Figure 12: Scatter Plot of Maximum Heart Rate achieved with target(heart disease)

Above figure shows the scatter plot of the maximum heart rate achieved that determines whether an individual is prone to heart disease or not.
From the above figure, most of the individuals have a heart rate more than 80

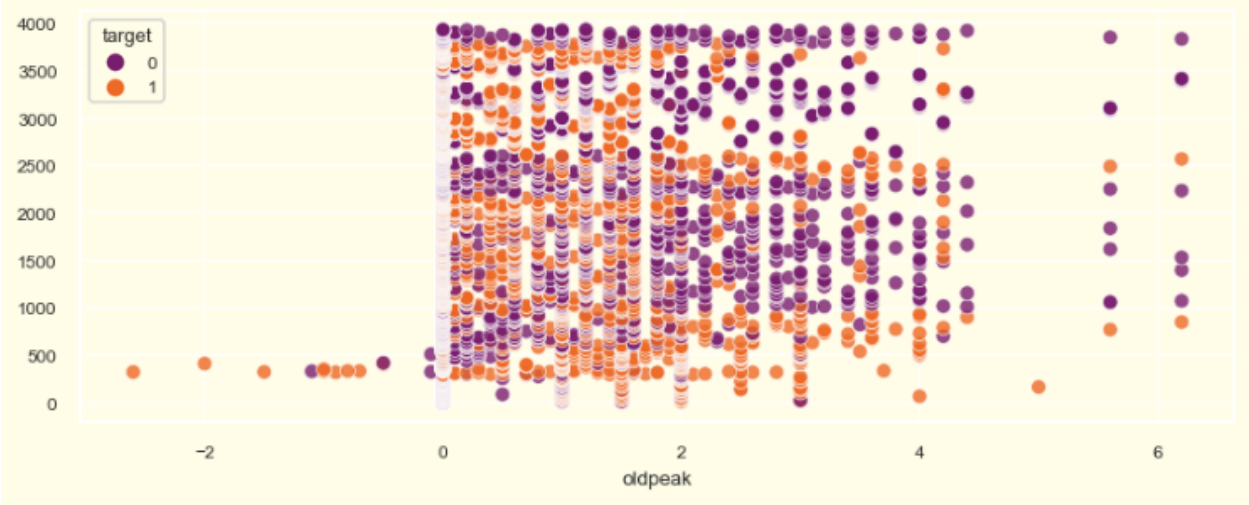

Figure 13: Scatter Plot of old peak with target(heart disease)

Above figure shows the scatter plot of the old peak with a target that determines whether an individual is prone to heart disease or not.

From the above figure, most of the old peaks are between 0 and 4.

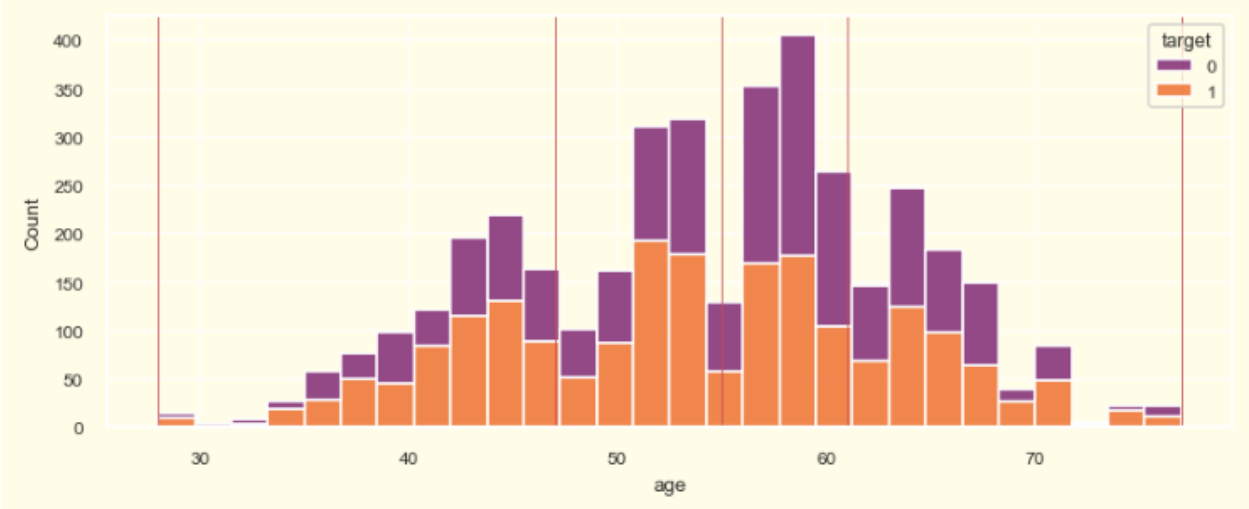

Figure 14: Histogram Plot of age with target(heart disease)

Above figure shows the histogram plot of the age with a target that determines whether an individual is prone to heart disease or not.

From the above figure, those who are aged 40 till aged 77 are at the risk of getting heart disease.

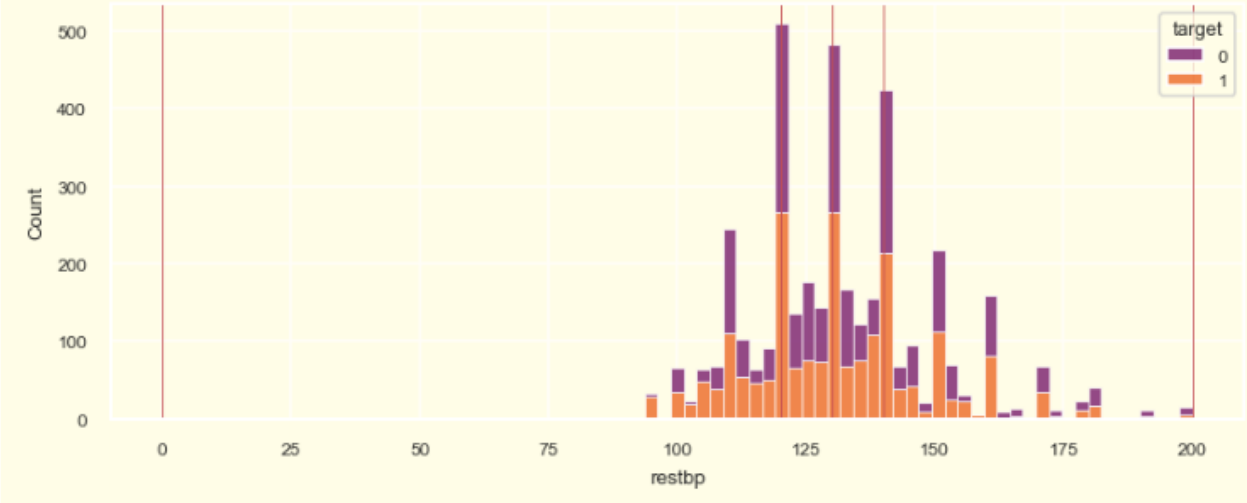

Figure 15: Histogram Plot of restbp with target(heart disease)

Above figure shows the histogram plot of the restbp with a target that determines whether an individual is prone to heart disease or not.

From the above figure, those who are have the resting blood pressure between 100 to 180 are at the risk of getting heart disease.

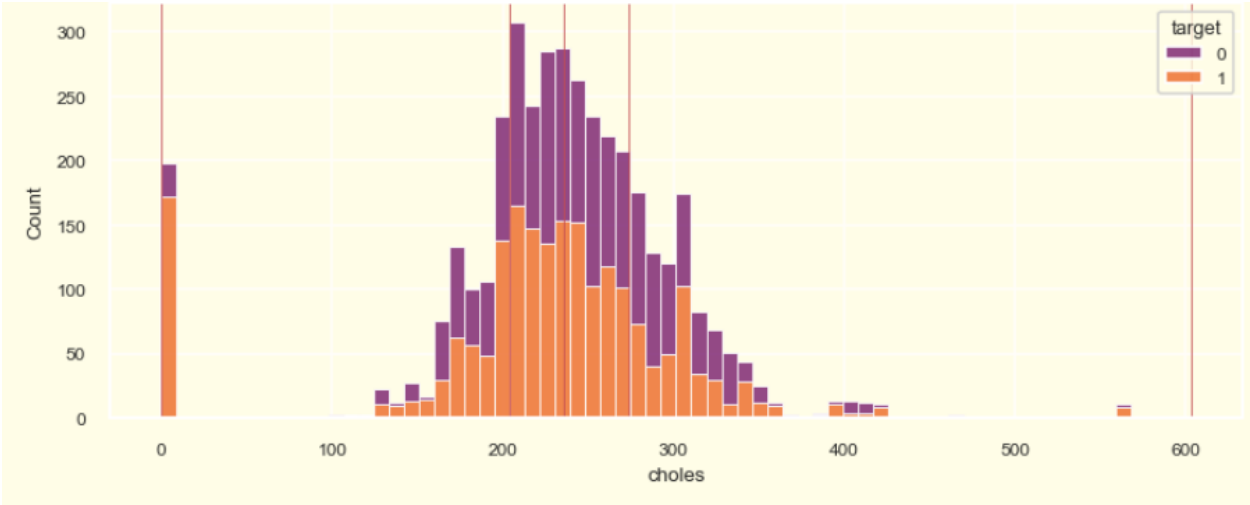

Figure 16: Histogram Plot of Cholesterol with target(heart disease)

Above figure shows the histogram plot of the cholesterol that determines whether an individual is prone to heart disease or not.

From the above figure, most of the individuals have a cholesterol between 150 to 350 s prone to heart disease.

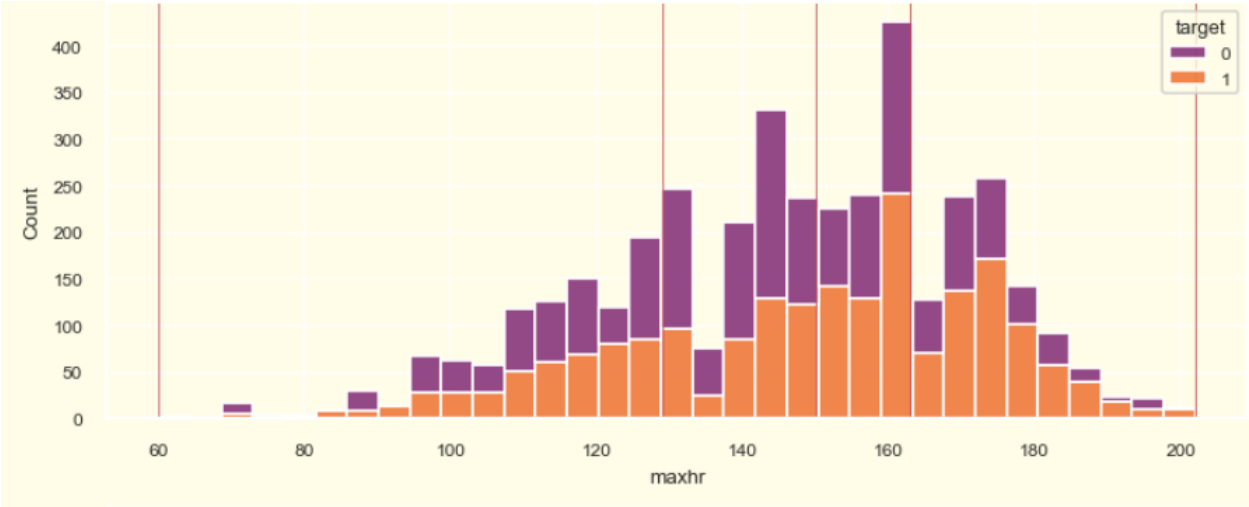

Figure 17: Histogram Plot of Maximum Heart Rate with target(heart disease)

Above figure shows the scatter plot of the maximum heart rate achieved that determines whether an individual is prone to heart disease or not.
From the above figure, most of the individuals have a heart rate more than 100 till 135 is prone to heart disease and start with 145 till 200 is prone to the heart disease.

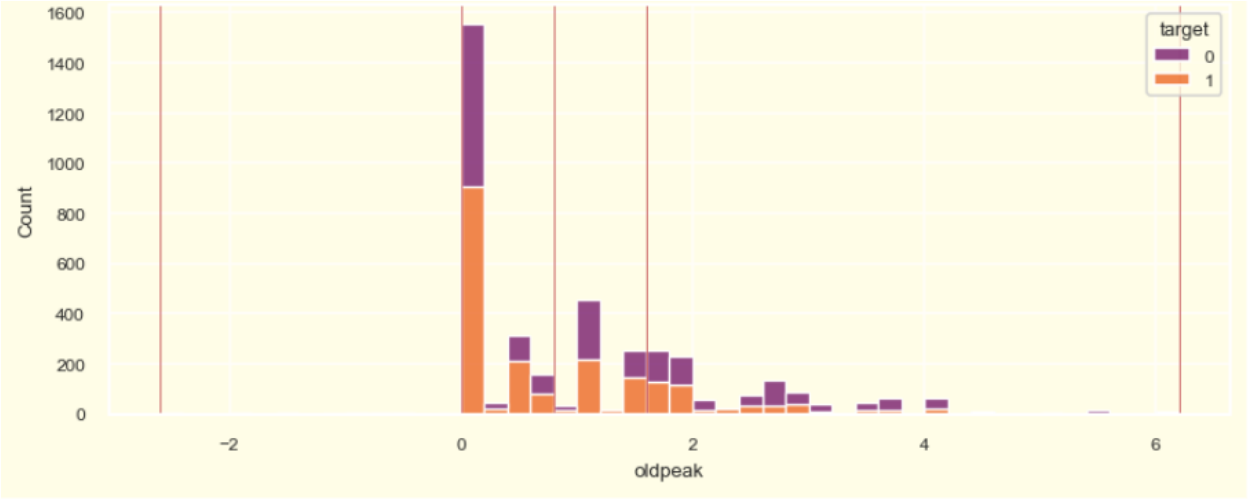

Figure 18: Histogram Plot of old peak with target(heart disease)
Above figure shows the scatter plot of the old peak with a target that determines whether an individual is prone to heart disease or not.
From the above figure, most of the old peaks are between 0 and 4 and the old peak at the 0 till 1 is prone to heart disease and starts with 1.7 to 2 also prone to heart disease.

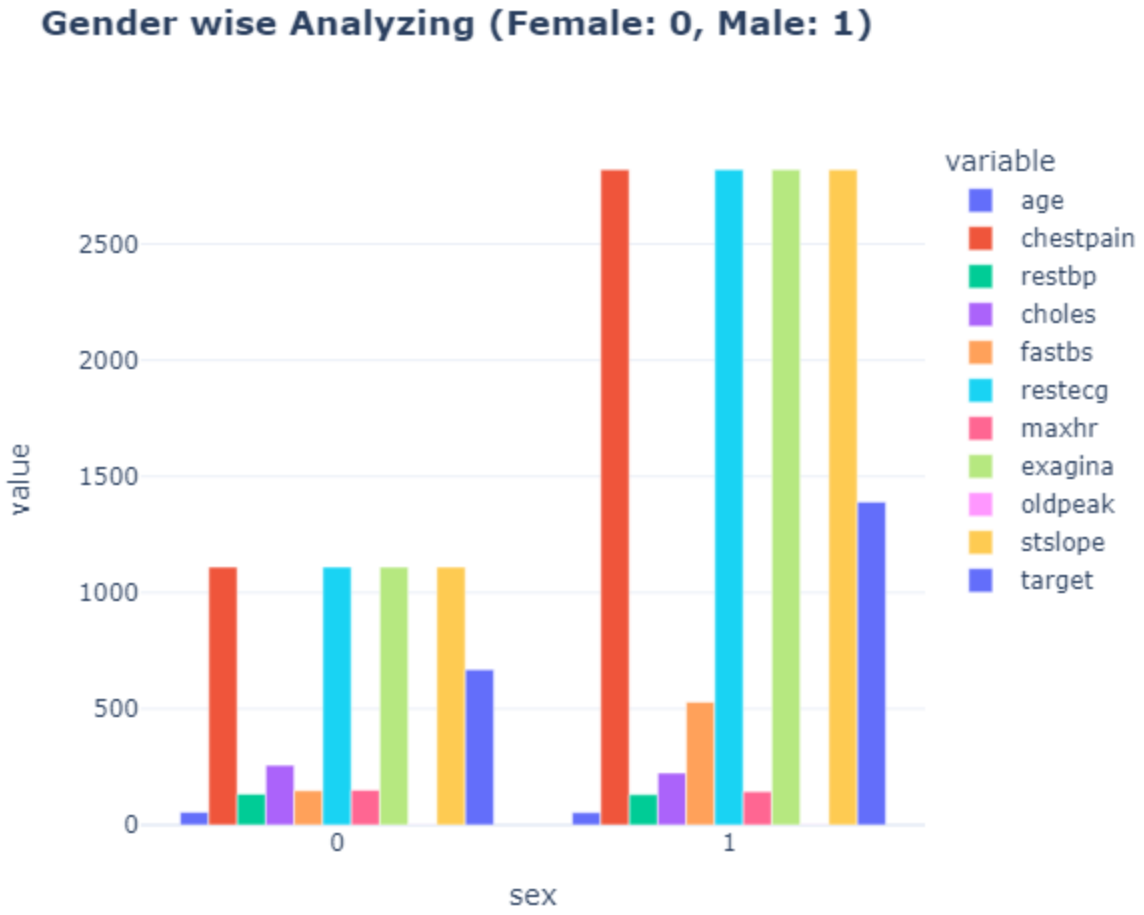

Figure 19: Bar Chart of showing the average gender wise analyzing with different variables

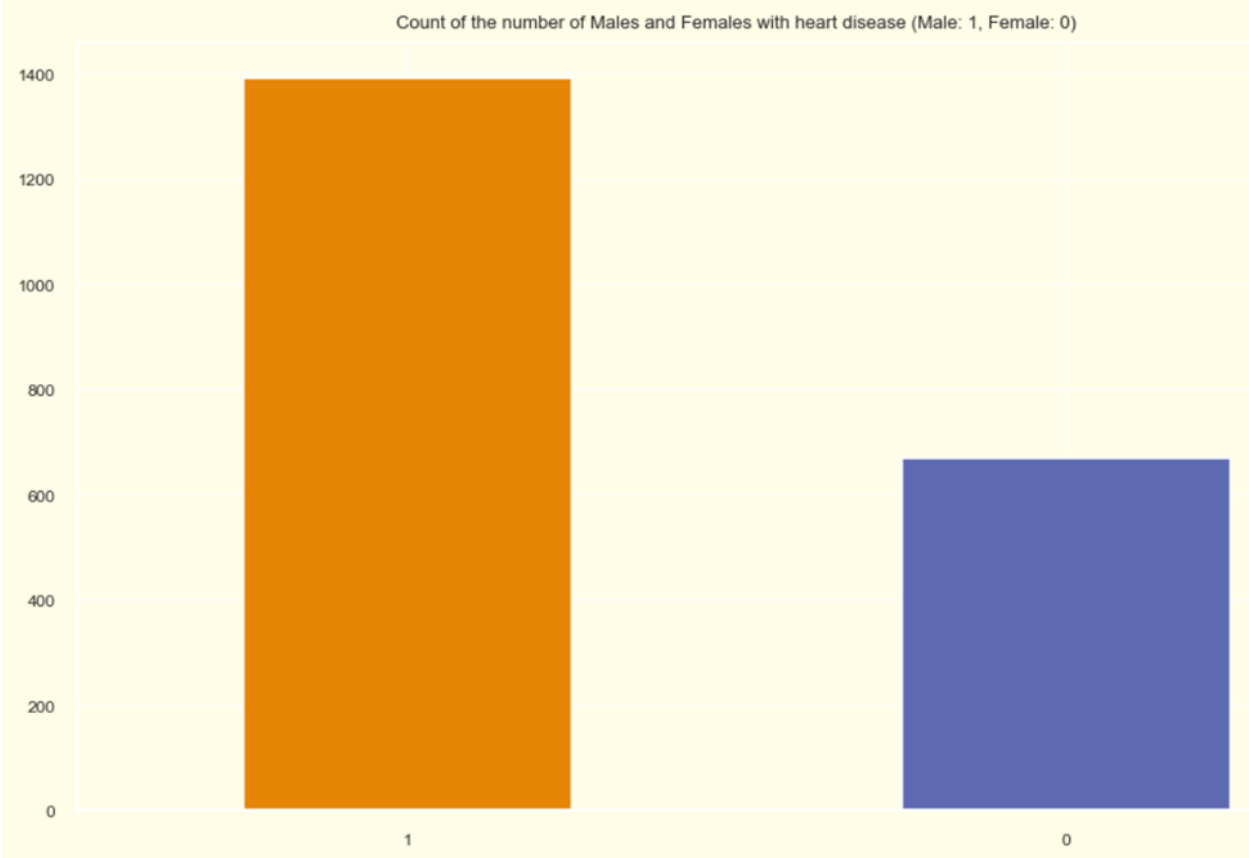

Figure 20: Bar Chart of Males and Females with Heart Disease

Above figure shows the bar chart of the genders who are prone to heart disease or not.
Orange Bar: Males who are prone to heart disease. Total = 1391
Purple Bar: Females who are prone to heart disease. Total = 669

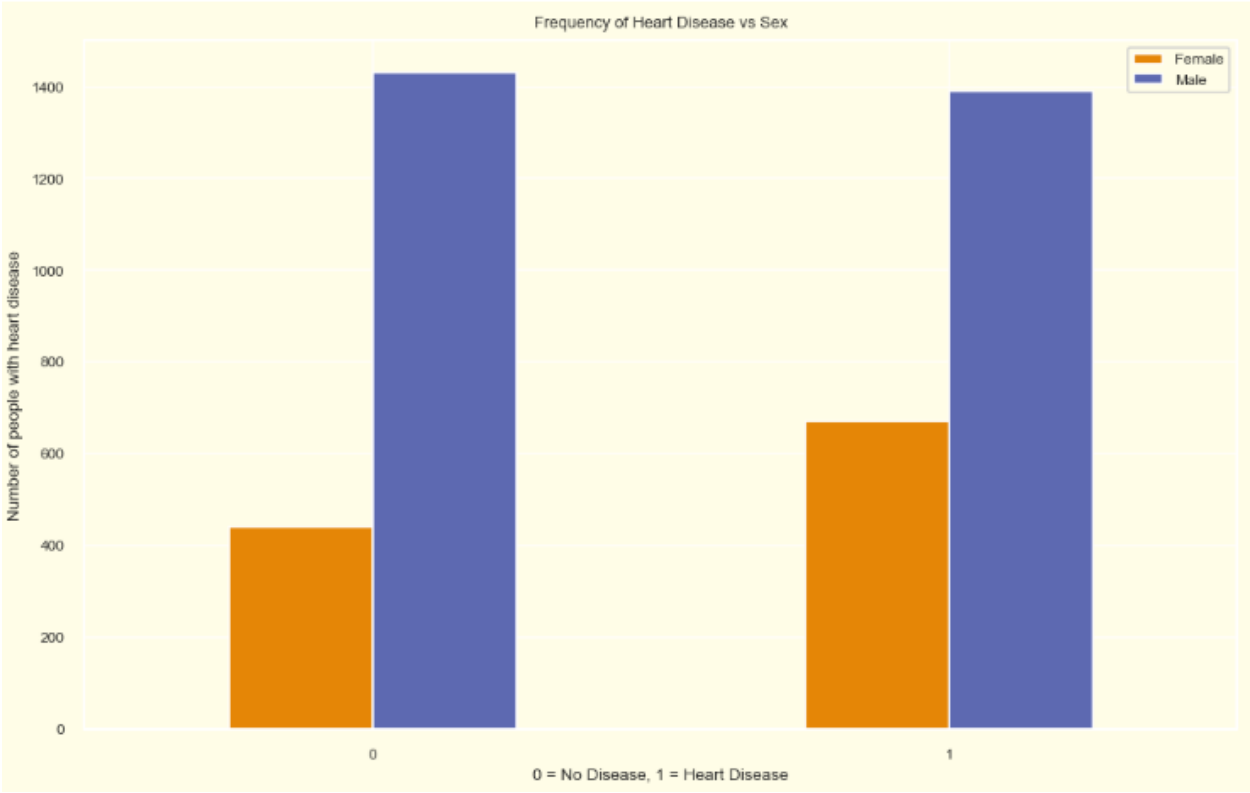

Figure 20: Bar Chart of Frequency of Heart Disease vs Sex

Above figure shows the bar chart of the Frequency of Heart Disease vs Sex.
Males have a higher risk of being prone to heart disease than Females.

3.2.2 Data Preprocessing

```
from sklearn.preprocessing import MinMaxScaler
scal=MinMaxScaler()
feat=['age', 'sex', 'chestpain', 'restbp', 'choles', 'fastbs', 'restecg', 'maxhr', 'exagina', 'oldpeak', 'stslope']
df[feat] = scal.fit_transform(df[feat])
df.head()
```

Code for performing the MinMaxScaler.

	age	sex	chestpain	restbp	choles	fastbs	restecg	maxhr	exagina	oldpeak	stslope	target
0	0.244898	1.0	0.333333	0.70	0.479270	0.0	0.0	0.788732	0.0	0.295455	0.0	0
1	0.183673	1.0	0.333333	0.65	0.469320	0.0	0.5	0.267606	0.0	0.295455	0.0	0
2	0.530612	1.0	0.666667	0.75	0.323383	0.0	0.0	0.436620	0.0	0.295455	0.0	0
3	0.224490	1.0	0.666667	0.60	0.562189	0.0	0.0	0.774648	0.0	0.295455	0.0	0
4	0.346939	0.0	0.333333	0.65	0.393035	0.0	0.0	0.774648	0.0	0.295455	0.0	0

Display the output after performing MinMaxScaler.

Transforming all our features by scaling each of the features to a given range using the MInMaxScaler. For instance, chestpain can be categorized into 4 different scales which range from 0 1 2 3. In the first row 0, 0.33333 can be seen outputted after it is being processed through the minmaxscaler, this is calculated through the formula X-X(min) / X(max) – X(min). This formula is used throughout the whole dataset as shown in the picture above.

```
X=df.drop("target",axis=1).values
Y=df.target.values
```

Code to perform drop target column

Target column is being dropped before the dataset is written into the model.
Column from age to stslope is being written into variable X
Column target is being written into variable Y

```
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.2,random_state=42)
```

Code to split data into train and test data

The data is split into train and test data and used 80/20 train/test split.
Train size is 80%
Test size is 20%

3.2.3 Classification Method

Our team has a total of 3 different types of algorithm which are Random Forest, K-Nearest Neighbor(KNN) and Support Vector Machine. After we have identified all the accuracy scores of all the algorithms, our team will try to stack up 2 algorithms together to increase the accuracy of the algorithms.

K-Nearest Neighbor (KNN) algorithm

The KNN algorithm is a popular supervised machine learning algorithm that can be used for both classification and regression problems. The algorithm works by first loading the training dataset and defining the value of K, which represents the number of nearest neighbors that will be used to determine the class or value of the test point. For each test point, the algorithm calculates the distance to all the points in the training dataset using a distance metric such as Euclidean distance. Then, the K nearest neighbors are found by sorting the distances and selecting the K nearest points. Finally, the class or value of the test point is assigned based on the majority class or average value of the K nearest points. This process is repeated for all the test points in the dataset, and the performance of the algorithm is evaluated by comparing the predicted classes or values with the actual classes or values in the testing dataset using evaluation metrics such as accuracy, precision, recall, and F1 score. While the KNN algorithm is simple to implement, the choice of K and distance metric can greatly affect the performance of the algorithm, and it can be computationally expensive for large datasets.

Stacking Cross Validation (SCV) Classifier Model

After our team identified all the accuracy of the 3 algorithms which are Random Forest, K-Nearest Neighbors and {}. Our team decided to use a stacking cross validation(SCV) classifier model to stack all the algorithms together to produce a higher accuracy model for better prediction accuracy. SCV uses the ensemble learning technique in order to combine multiple classification models together. This SCV is an extension of the usual stacking algorithm and by using cross validation to prepare input data at the level 2 classifier. This might cause overfitting as the dataset to fit into the level 1 classifier will be used as the inputs for the level 2 classifier. However, through the concept of cross validation, it allows the dataset to be split up into k folds and in k successive rounds. The k-1 folds will be used to fit the level 1 classifier and after that in each round, the remaining 1 subset that was applied to the level 1 classifier will be added in each iteration. Next, the level 2 classifier will be used to predict the results that were stacked as the input data. After the SCV has finished training, the level 1 classifier will be fitted entirely into the dataset as the figure shown below.

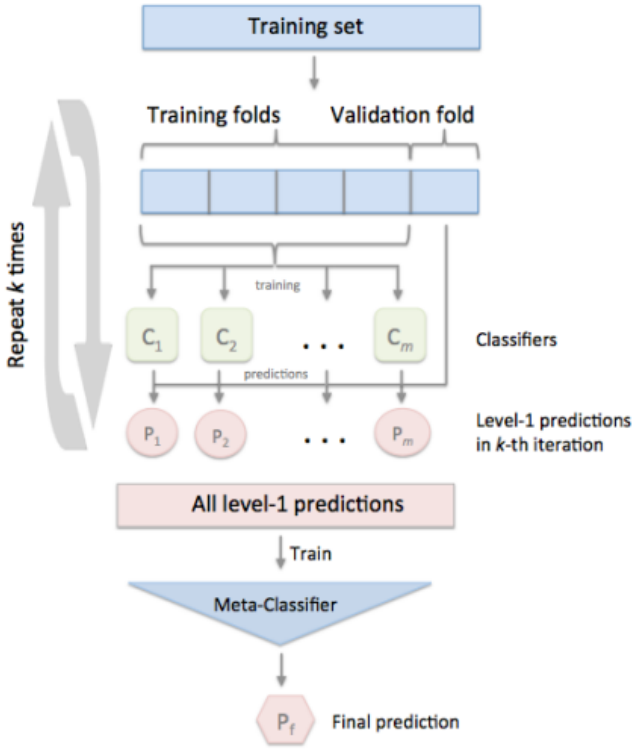

Figure 21: Stacking CV Classifier

Source Code

K-Nearest Neighbor (KNN) algorithm

```
np.random.seed(42)
from sklearn.neighbors import KNeighborsClassifier
#Define Model
Knn_clf=  KNeighborsClassifier(n_neighbors=5)
#Fit the Model
Knn_clf.fit(X_train,Y_train)
Knn_Y_pred=Knn_clf.predict(X_test)
#Make Predictions
Knn_score=Knn_clf.score(X_test,Y_test)
#print(Knn_score)
evaluation(Y_test,Knn_Y_pred)
```

```
{'accuracy': 0.936, 'recall': 0.945, 'F1 score': 0.94}
```

Figure 22: K-Nearest Neighbor (KNN) Code

Figure 22 shows the source code of building the K-Nearest Neighbor model. First, we will have to import the K-Nearest Neighbor model from the scikit learn library. Next, we need to define the name of the model where in this case we have named the model as “Knn_clf”. After defining the model, we will be training the model by fitting the training set of X_train and Y_train. Aside from that, the test data (X_test, Y_test) will be used to predict the score of the model. Based on figure , the accuracy score of the model is 0.936.

Stacking CV Classifier Model (SCV)

```
from mlxtend.classifier import StackingCVClassifier
#Stack the the model
scv=StackingCVClassifier(classifiers=[Knn_clf,RF_clf],meta_classifier= Knn_clf)
scv.fit(X_train,Y_train)
scv_score=scv.score(X_test,Y_test)
scv_Y_pred=scv.predict(X_test)
evaluation(Y_test,scv_Y_pred)
```

```
{'accuracy': 0.978, 'recall': 0.983, 'F1 score': 0.98}
```

Figure 23: Stacking CV Code

Figure 23 shows the source code for the Stacking CV Classifier Model (SCV). First, we will have to import the StackingCVClassifier from the mlxtend library. Next, we need to define the name of the model where in this case we have named the model as "scv". In order to stack the 2 algorithms together, we would have to include the two models that we have identified which are K-Nearest Neighbor (KNN) and Random Forest (RF). After stacking the models together, we will be training the model by fitting the training set of X_train and Y_train. Aside from that, the test data (X_test, Y_test) will be used to predict the score of the model. Based on Figure 18, we can see that the accuracy score of the SCV model is 0.978.

3.2.4 Comparing the result of different classification models

Accuracy Score of Random Forest, K-Nearest Neighbor and Stacking CV Classifier

	Model	Accuracy
0	SCV	97.839898
1	Random Forest	97.331639
2	SVM	81.194409
3	KNN	93.646760

Figure 24: Accuracy Result of RF, KNN,SVM and SCV

From figure 24, we can see that after stacking up the 2 algorithms together, the accuracy score of the Stacking CV classifier is 97.84% which is the highest when compared to the other 3 models. Random Forest has an accuracy score of 97.33% whereas Support Vector Machine has accuray of 81.19% and K-Nearest Neighbor has an accuracy score of only 93.65%. From the figure above we can see that StackingCV has the highest accuracy, moving down to the Random Forest then K-Nearest Neighbor model then only Support Vector Machine. Therefore, it can be said that the StackingCV Classifier is the most accurate model to be used for our heart disease prediction application as compared to other models.

ROC Curve

Roc Curve is used to summarize the trade off between the true positive rate and false positive rate for the predictive model using different probability thresholds. All the roc curve shows below can be seen that all the models are slanting towards the y-axis. The accuracy for the Roc Curve of all the models are followed by 97.81% and 97.33%, 93.59%, 81.19%. Hence, stacking cv will be the highest accuracy score and if compared to algorithms Random Forest will be the highest compared to KNN and SVM.

Stacking CS(SCV)

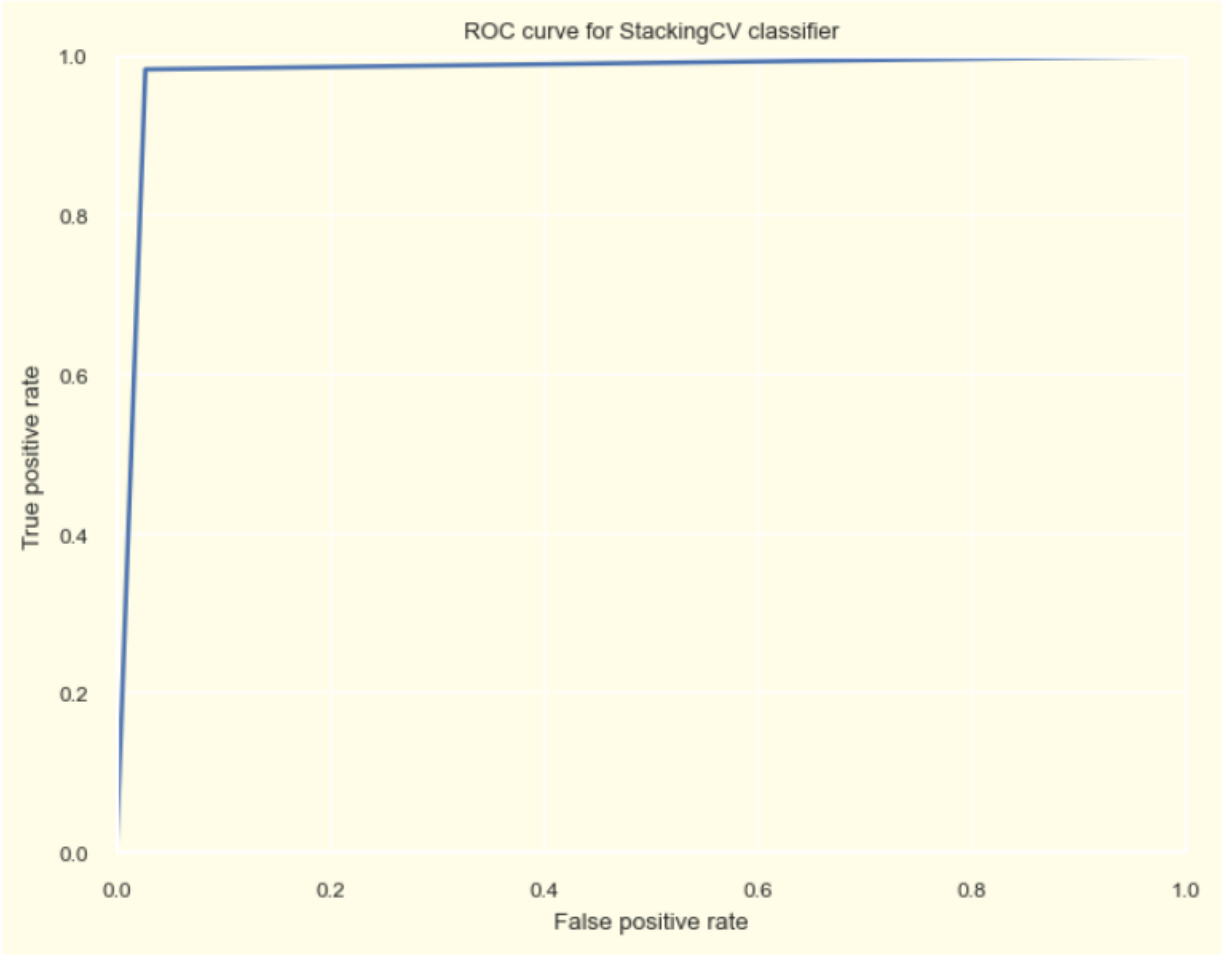

Figure 25: ROC Curve of SCV

Random Forest

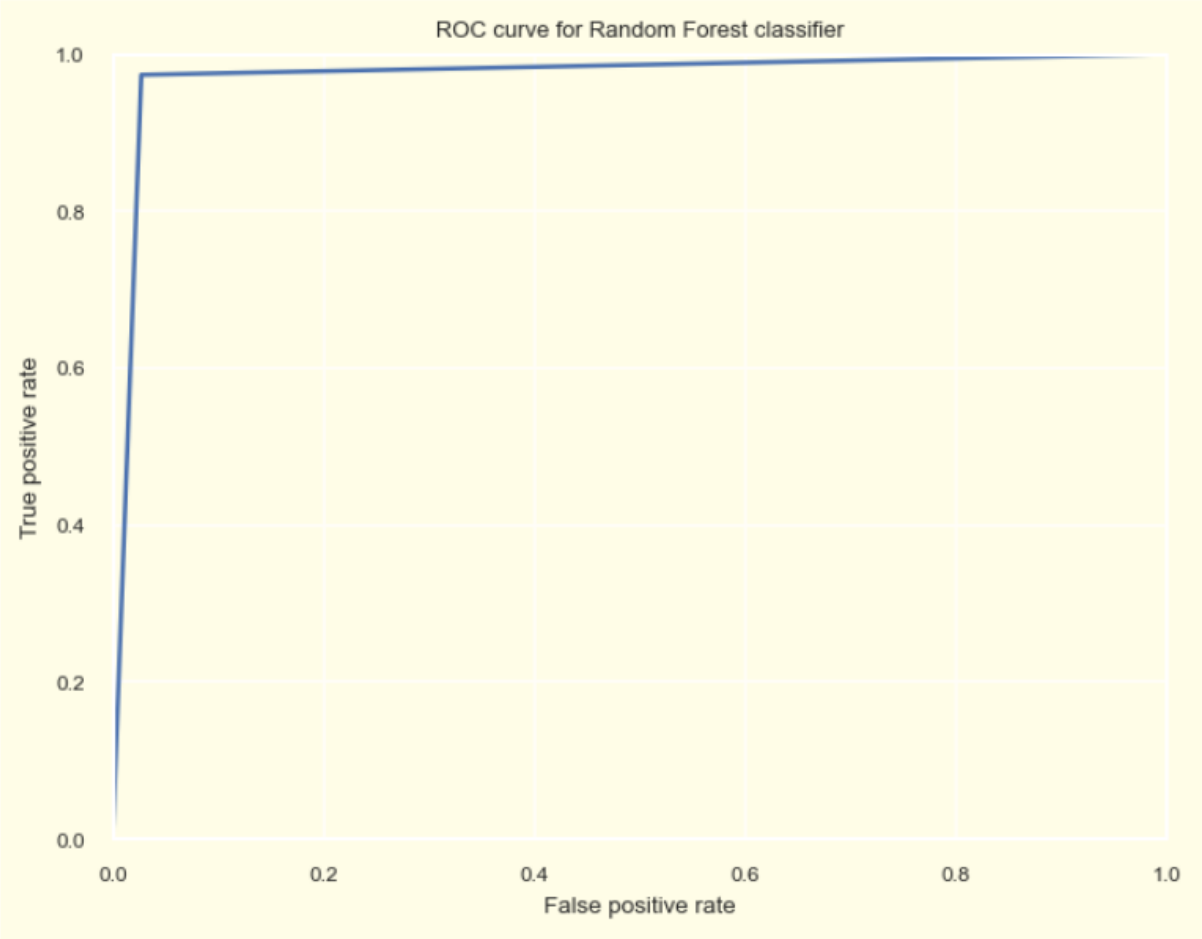

Figure 26: ROC Curve of Random Forest

K-Nearest Neighbor

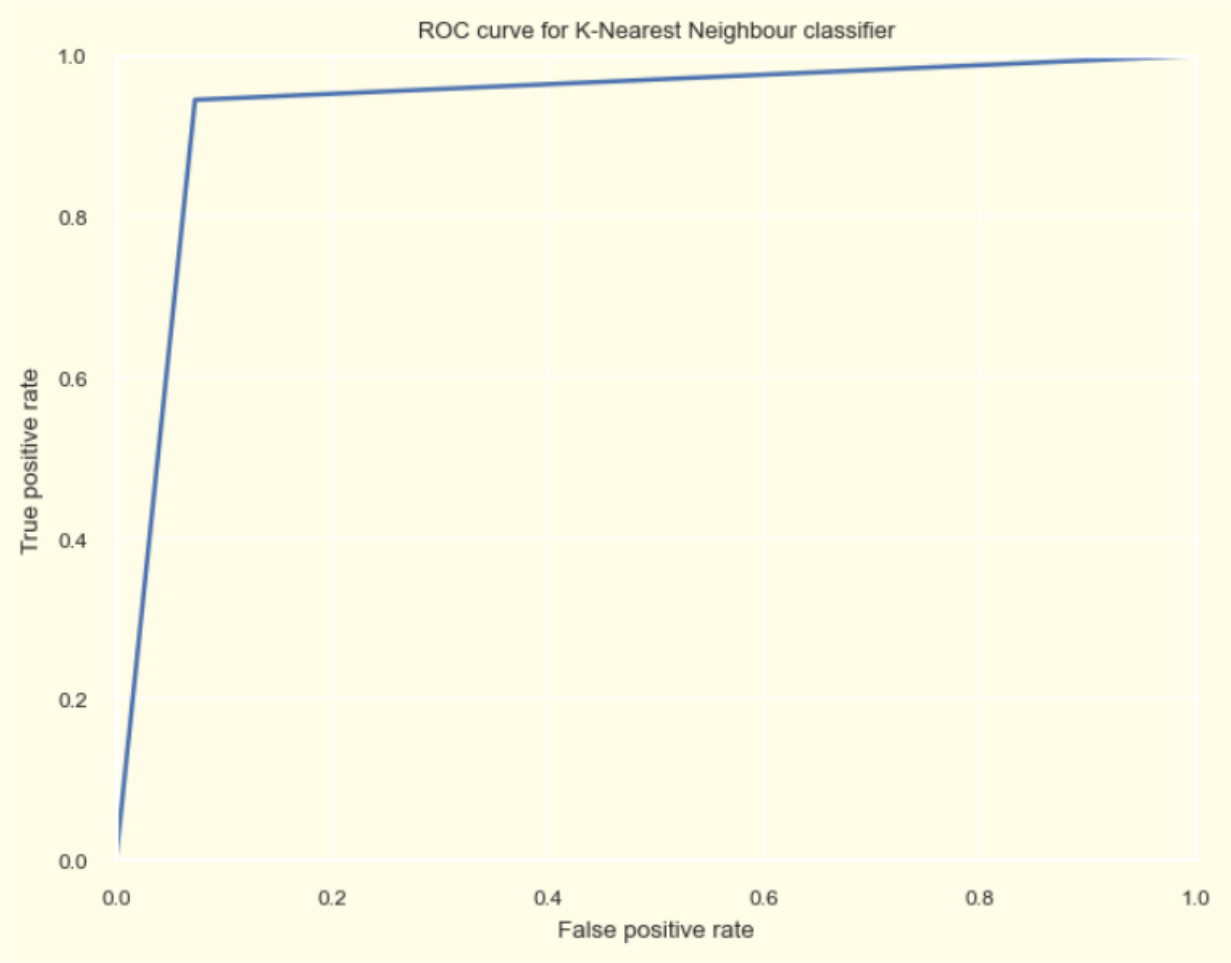

Figure 27: ROC Curve of K-Nearest Neighbor

Support Vector Machine(SVM)

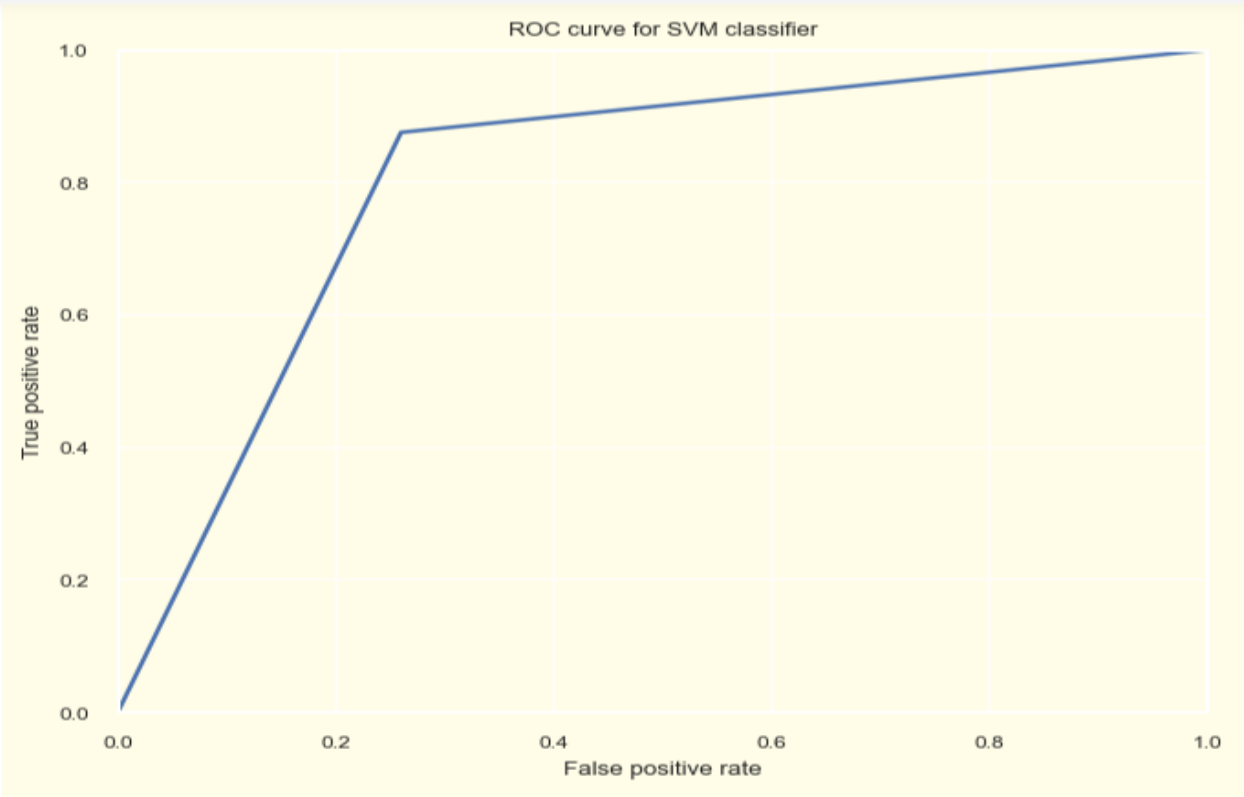

Figure 28: ROC Curve of Support Vector Machine(SVM)

Confusion Matrix

Stacking CV:

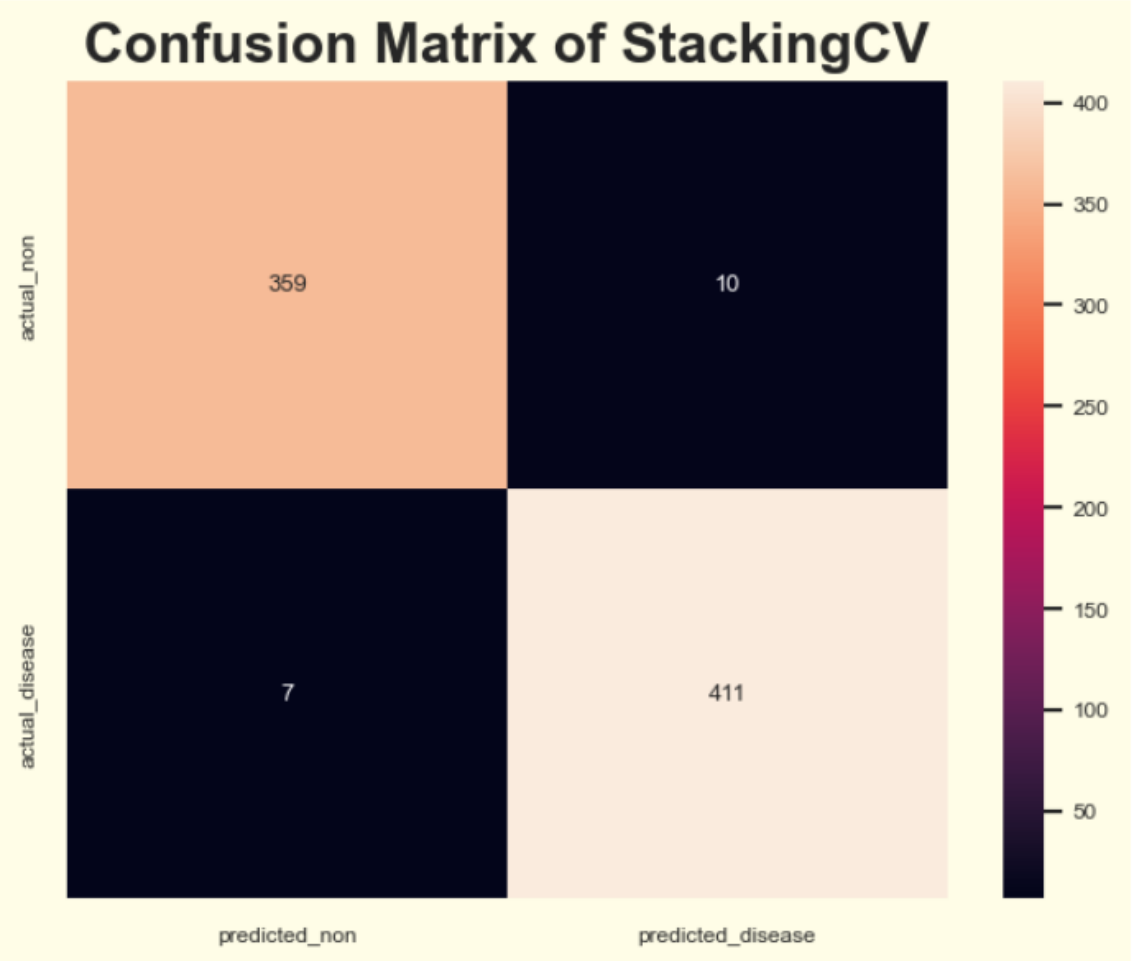

Figure 29: Confusion Matrix of Stacking CV

Random Forest:

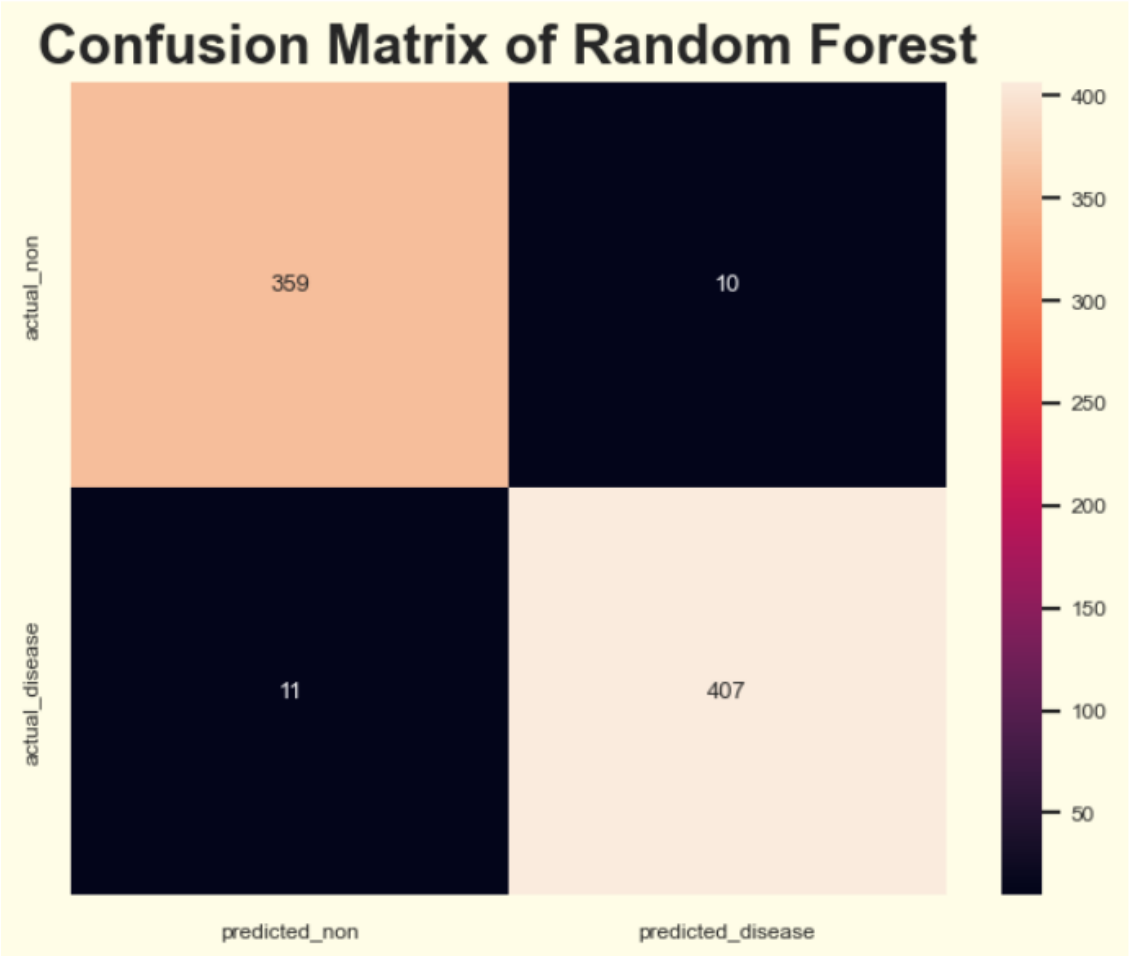

Figure 30: Confusion Matrix of Random Forest

K-Nearest Neighbor:

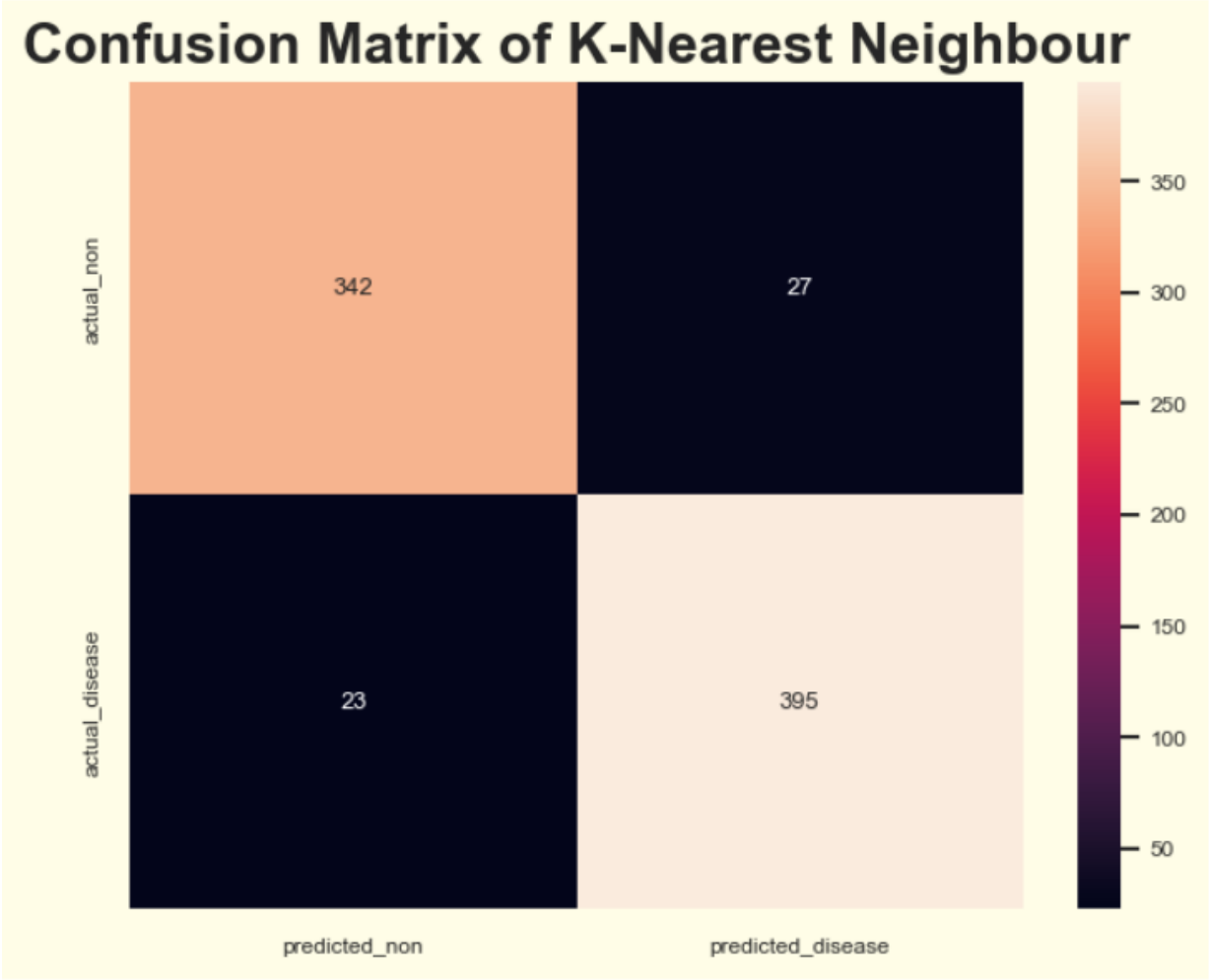

Figure 31: Confusion Matrix of KNN

Support Vector Machine(SVM):

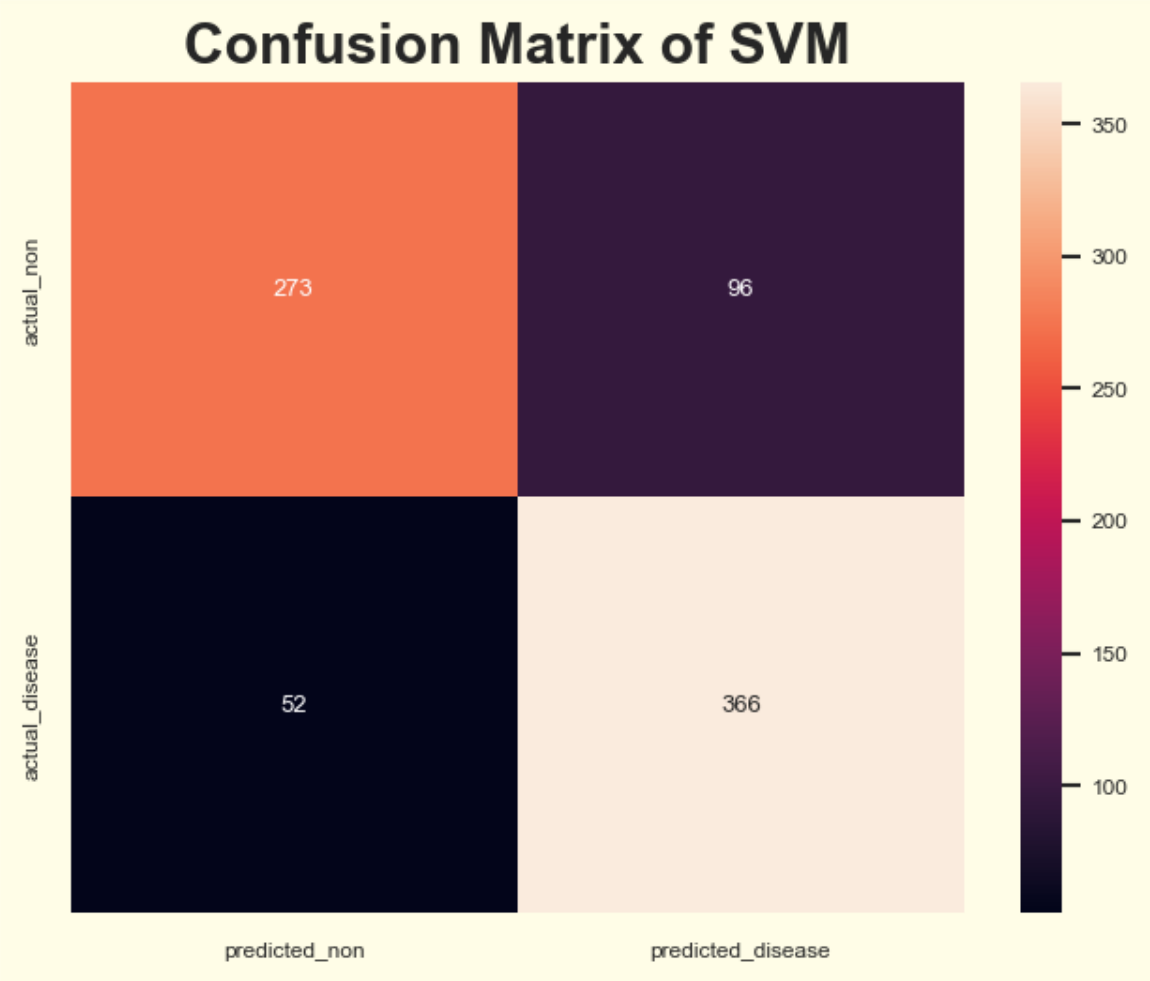

Figure 32: Confusion Matrix of SVM

The confusion matrix for each of the models above helps us to summarize on how each model performed based on the testing data. The table below shows us the description of the confusion matrix of heatmap.

<table>
<tr><td rowspan="2"></td><td colspan="3">TP = True Positive TN = True Negative</td></tr>
<tr><td colspan="3">FP = False Positive FN = False Negative</td></tr>
<tr><td rowspan="2">Actual</td><td>actual_non</td><td>TP: An individual that is not having heart disease and correctly identified by the algorithm.</td><td>FN: An individual that is not having heart disease but the algorithm said is diagnosed with heart disease.</td></tr>
<tr><td>actual_disease</td><td>FP: An individual that is diagnosed heart disease but the algorithm said not having heart disease.</td><td>TN: An individual that is diagnosed with heart disease and correctly identified by the algorithm.</td></tr>
<tr><td rowspan="2"></td><td></td><td>predicted_non</td><td>predicted_disease</td></tr>
<tr><td colspan="3">Predicted</td></tr>
</table>

The TP and TN indicates how many times the algorithm has been correctly classified. According to the confusion matrix above, we can see that the number of correctly classified by K-Nearest Neighbor, Random Forest and Stacking CV are 342+395=737, 359+407=766, 359+411=770 respectively. With these numbers, we can conclude that the StackingCV classifier is the most suitable classification algorithm to be used for our heart disease prediction application.

Classification Report: K-Nearest Neighbor, Random Forest and Stacking CV

Suppot Vector Machine Classification Report

	precision	recall	f1-score	support
0	0.84	0.74	0.79	369
1	0.79	0.88	0.83	418
accuracy			0.81	787
macro avg	0.82	0.81	0.81	787
weighted avg	0.81	0.81	0.81	787

K-Nearest Neighbour Classification Report

	precision	recall	f1-score	support
0	0.94	0.93	0.93	369
1	0.94	0.94	0.94	418
accuracy			0.94	787
macro avg	0.94	0.94	0.94	787
weighted avg	0.94	0.94	0.94	787

Random Forest Classification Report

	precision	recall	f1-score	support
0	0.97	0.97	0.97	369
1	0.98	0.97	0.97	418
accuracy			0.97	787
macro avg	0.97	0.97	0.97	787
weighted avg	0.97	0.97	0.97	787

Stacking CV Classification Report

	precision	recall	f1-score	support
0	0.98	0.97	0.98	369
1	0.98	0.98	0.98	418
accuracy			0.98	787
macro avg	0.98	0.98	0.98	787
weighted avg	0.98	0.98	0.98	787

Figure 33: Classification Report of KNN, RF and SCV

Precision

The proportion of properly predicted positive observations to the total anticipated positive observations.

Recall

The ratio of properly predicted observations to the total number of observations in the actual class.

F1 Score

Showing us the weighted average of Precision and Recall.

The formula for precision, recall and F1 score is shown below:

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$\text{F1} = 2 \times \frac{Precision * Recall}{Precision + Recall}$$

Figure 34: Formula for precision, recall and F1 score

From figure 34, Support Machine Vector(SVM) has the worst precision, recall and F1 score and K-Nearest Neighbor (KNN) will be the second last even though it is higher than the SVM. Therefore, SVM and KNN will be excluded from being used in our heart disease prediction application . In contrast, both models of Random Forest (RF) and Stacking CV (SCV) have a high value when it comes to the precision, recall and F1 value. The SCV model has a slight advantage when compared to RF with a slight increase by 0.01 in terms of precision, recall and F1 value. With this, it is able to conclude that the SCV model has the highest accuracy out of all the models therefore it is the most suitable model to be used for our heart disease prediction application and Random Forest will be the second choice.

3.2.5 User Interface of Heart Disease Prediction System

Source Code of Writing pkl file:

```
import pickle as pkl

#Save Model

# StackingCV
pkl.dump(scv,open("final_scv_model.p","wb"))

#Random Forest
pkl.dump(RF_clf, open("final_rf_model.p","wb"))

#SVM
pkl.dump(Knn_clf,open("final_knn_model.p","wb"))

#KNN
pkl.dump(svm_model,open("final_svm_model.p","wb"))
```

Figure 35: Source code for saving model into pkl file

First, we will be storing the models that will be displayed in the User Interface (UI). The models that we have chosen to store are the 2 models with the highest accuracy which are the StackingCV classifier model and the Random Forest classifier model. The models are being stored in a pkl format which enables us to load the model into the heart-app.py. After storing the model into pkl file:

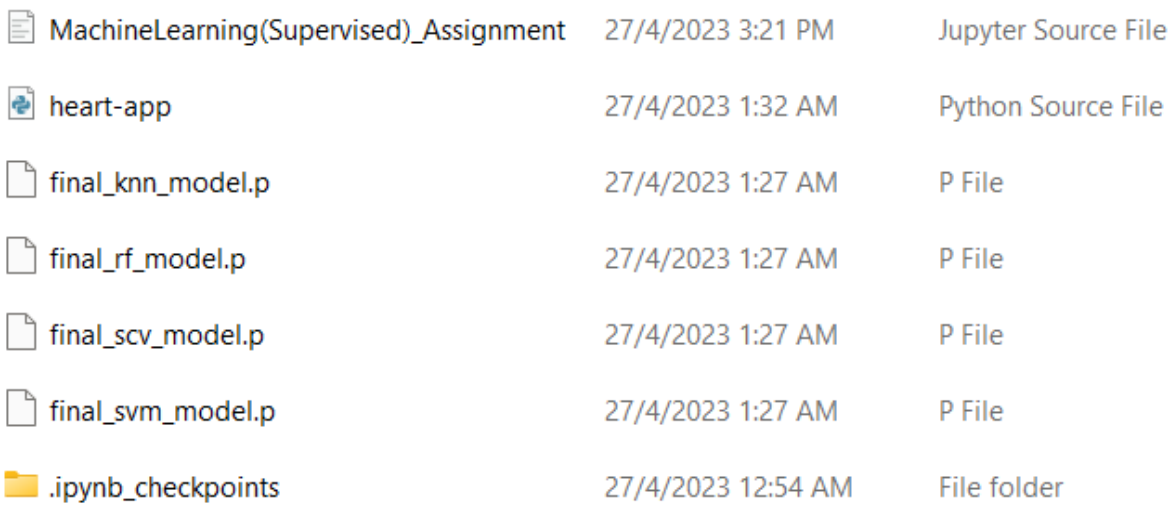

Figure 36: File of the models have been created

The Random Forest Classifier Model ,StackingCV Classifier Model, K-Nearest Neighbor (KNN) and Support Vector Machine (SVM) pkl file are final_rf_model.p, final_scv_model.p, final_knn_model.p and final_svm_model.p respectively.

User Interface of Heart Disease Prediction Application:

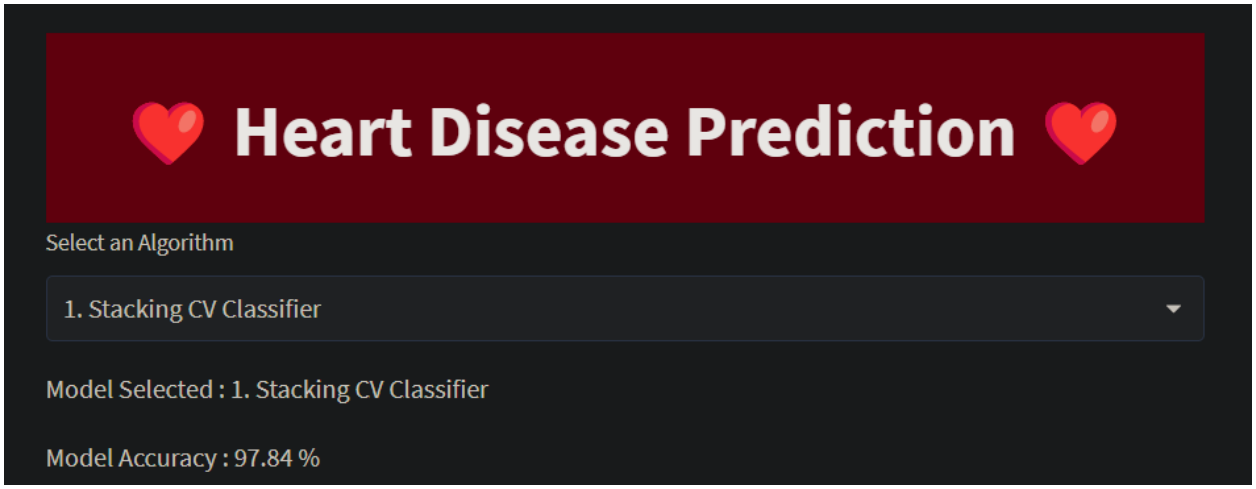

Figure 37: User Interface of Heart Disease Prediction Application

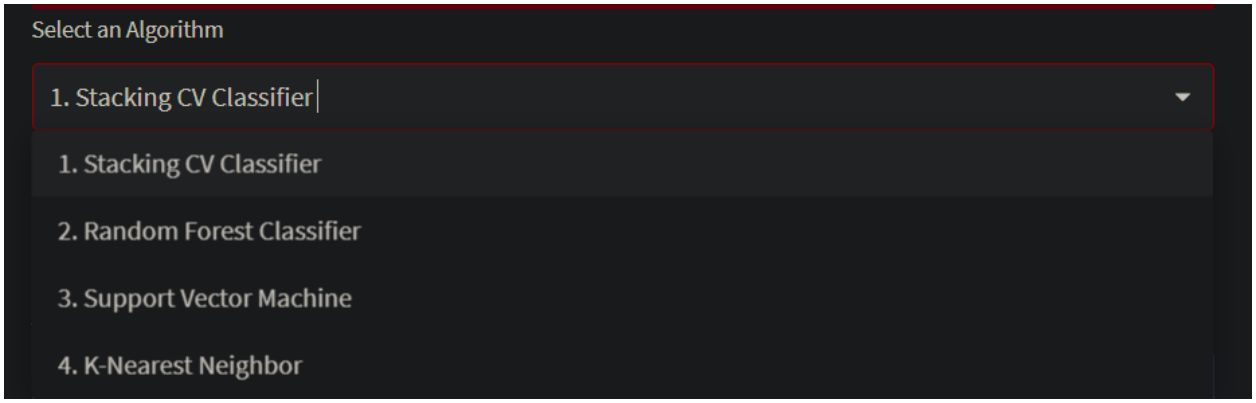

Figure 38: Model Selection Option

In our heart disease prediction application, it allows our users to choose which algorithm to use when predicting whether they are prone to heart diseases or not.

3.3. System flowchart/activity diagram

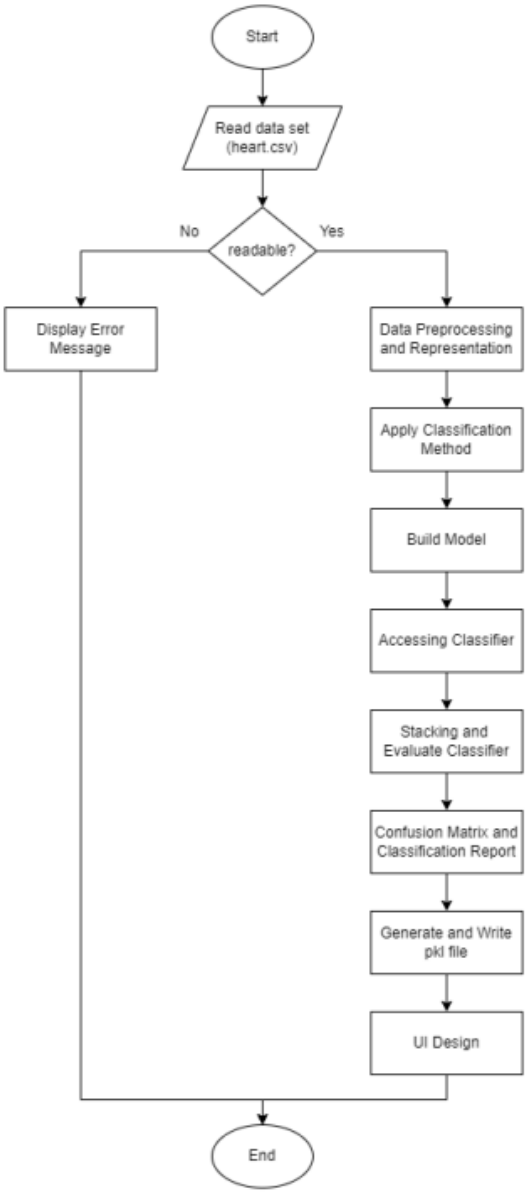

Figure 39: Module Flowchart

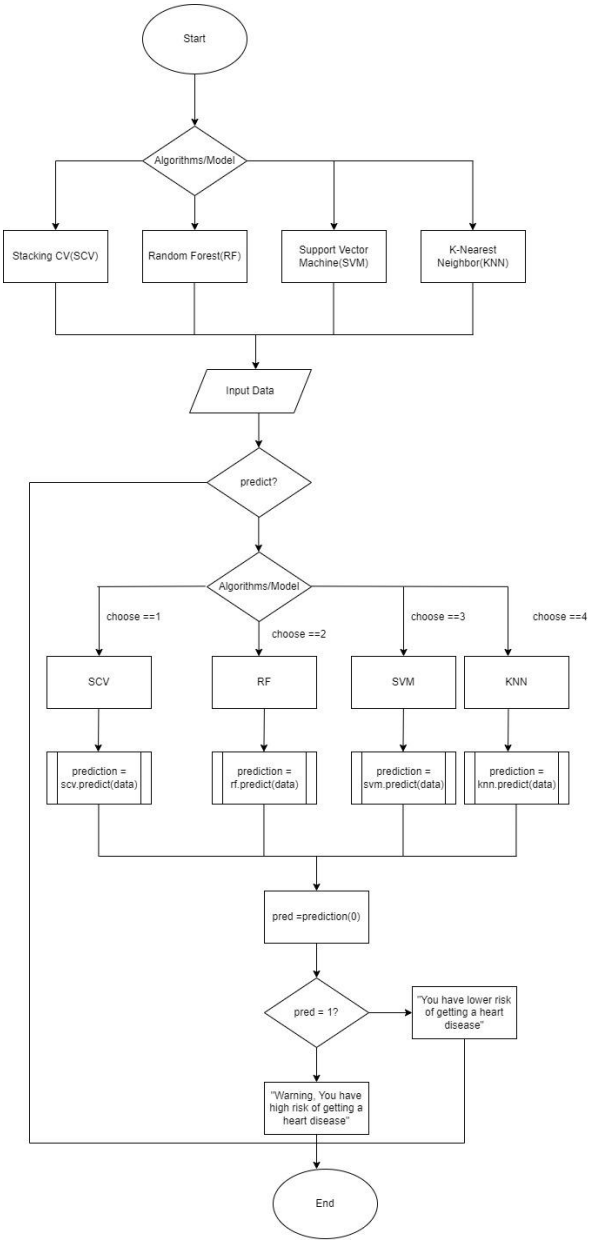

Figure 40: System flowchart

3.4. Proposed test plan/hypothesis

Step 1: Select the data from the dataset heart.csv that I chose to input and test in the project.

Selected Data:

	Data										
S	Age	Sex	Chest Pain	RestBp	Choles	FastBs	RestEeg	MaxHr	Exagina	Oldpeak	Stslope
S1	58	Male	Atypical Angina	136	164	No	ST-T Wave abnormality	99	Yes	2.00	Flatsloping
S2	70	Male	Asymptomatic	170	192	No	ST-T Wave abnormality	129	Yes	3.00	Downsloping
S3	52	Male	Atypical Angina	140	100	No	Nothing to note	138	Yes	0.00	Upsloping
S4	42	Male	Non-Anginal Pain	160	147	No	Nothing to note	146	No	0.00	Upsloping
S5	48	Female	Asymptomatic	138	214	No	Nothing to note	108	Yes	1.50	Flatsloping
S6	59	Female	Asymptomatic	130	338	Yes	ST-T Wave abnormality	130	Yes	1.50	Flatsloping
S7	45	Female	Atypical Angina	130	237	No	Nothing to note	170	No	0.00	Upsloping
S8	48	Female	Atypical Angina	120	284	No	Nothing to note	120	No	0.00	Upsloping

Step 2: State the hypothesis for these selected input data

H1)	S1 will be diagnosed as a heart disease patient after predicting using the K-Nearest Neighbor Model.
H2)	S2 will be diagnosed as a heart disease patient after predicting using the K-Nearest Neighbor Model.
H3)	S3 will be diagnosed as a non heart disease patient after predicting using the K-Nearest Neighbor Model.
H4)	S4 will be diagnosed as a non heart disease patient after predicting using the K-Nearest Neighbor Model.
H5)	S5 will be diagnosed as a heart disease patient after predicting using the K-Nearest Neighbor Model.
H6)	S6 will be diagnosed as a heart disease patient after predicting using the K-Nearest Neighbor Model.
H7)	S7 will be diagnosed as a non heart disease patient after predicting using the K-Nearest Neighbor Model.
H8)	S8 will be diagnosed as a non heart disease patient after predicting using the K-Nearest Neighbor Model.

Step 3: Input the selected data in the UI application to do prediction by choosing the K-Nearest Neighbor algorithm as shown below.

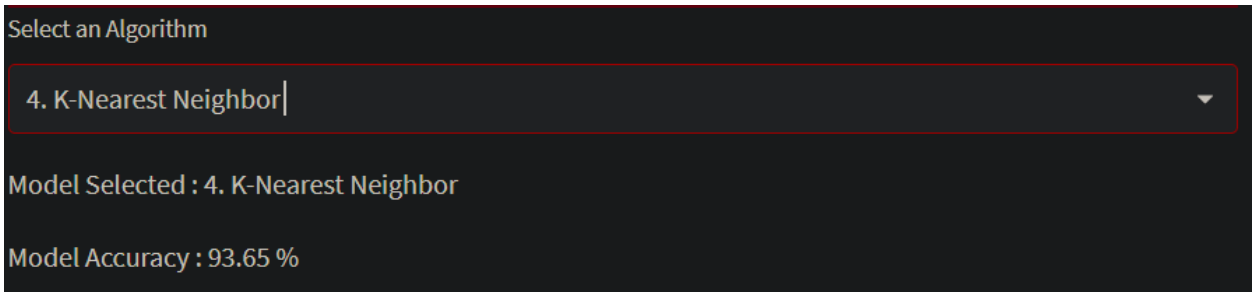

Figure 41: : User Interface after selecting K-Nearest Neighbor (KNN) Model

4. Result

4.1. Results

K-Nearest Neighbor Model

Result after predict S1:

Figure 42: Results after predict S1

Result after predict S2:

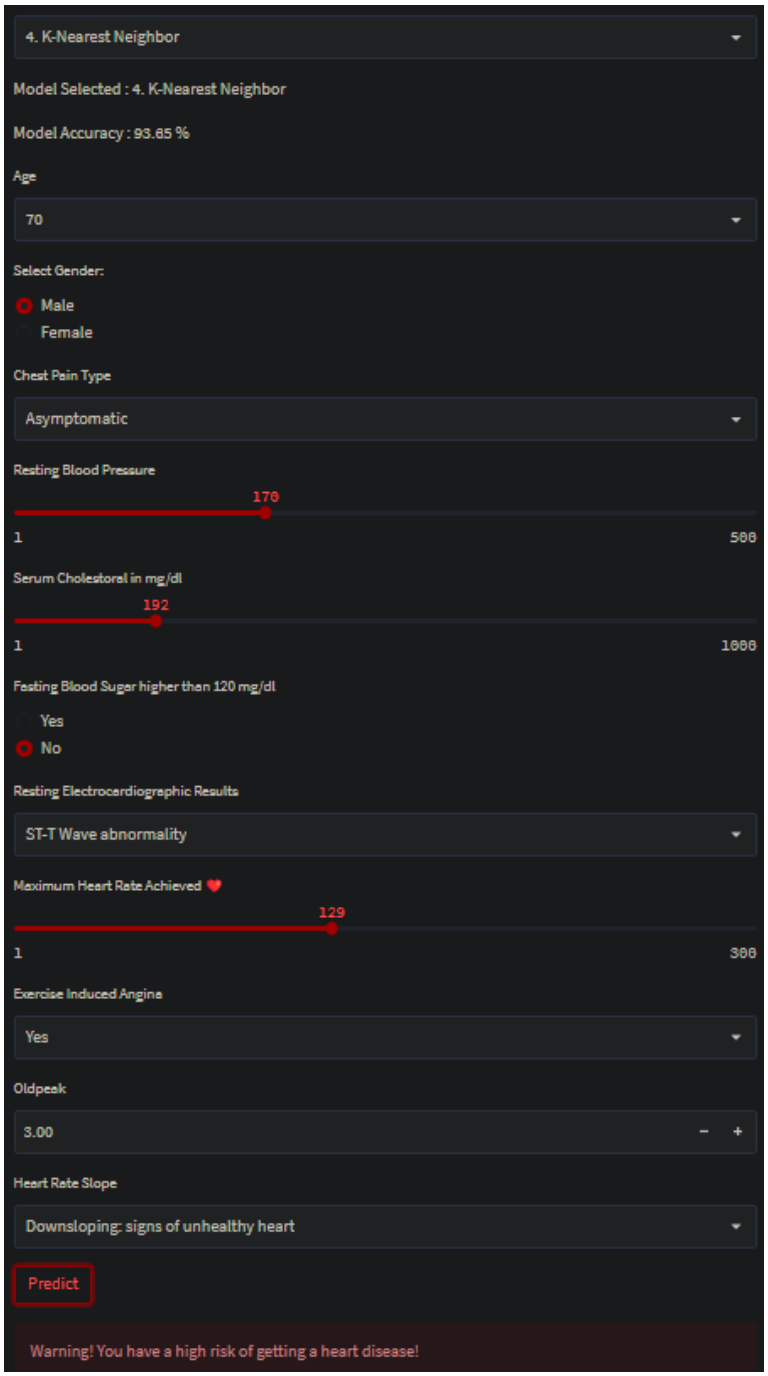

Figure 43: Results after predict S2

Result after predict S3:

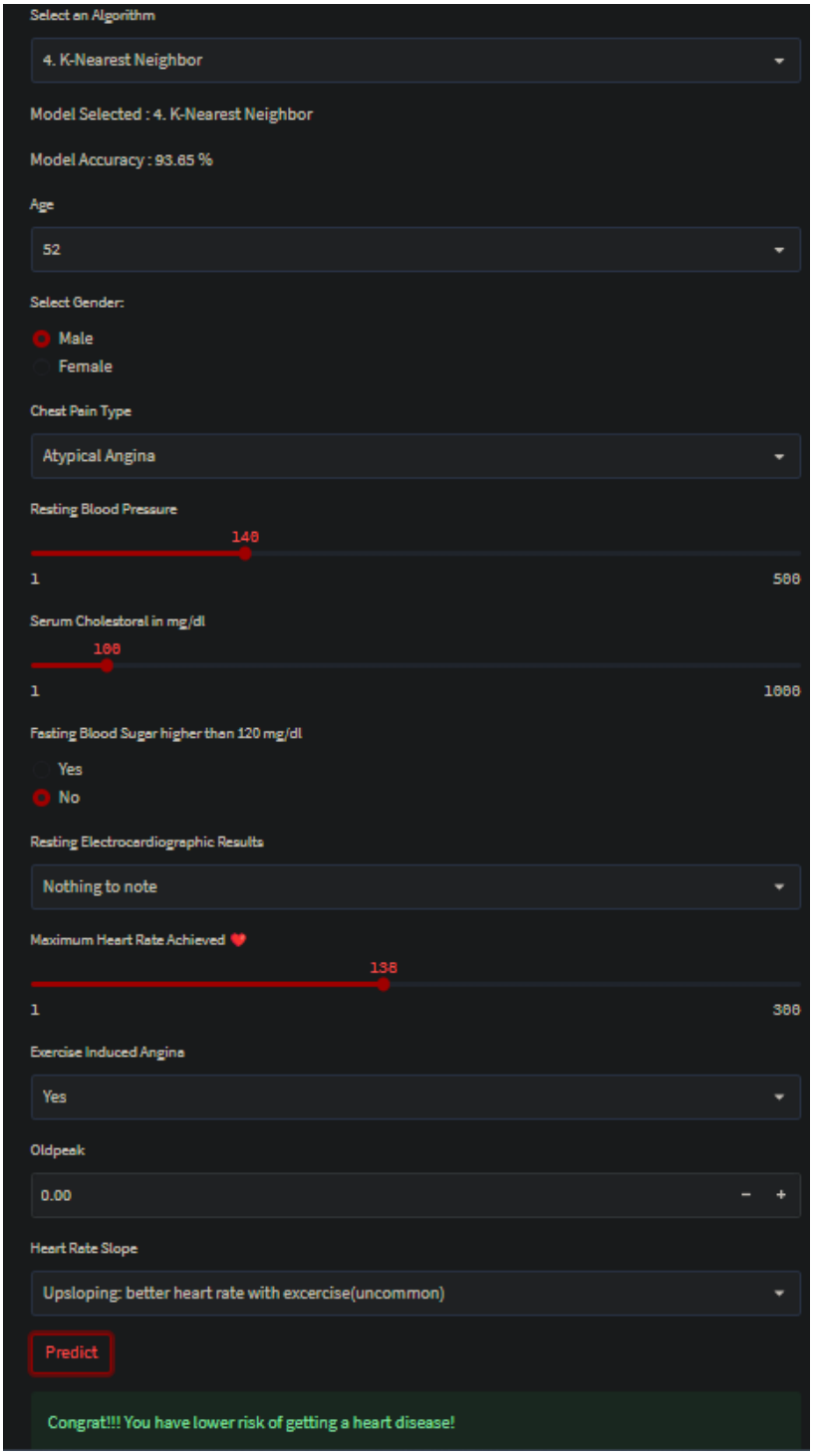

Figure 44: Results after predict S3

Result after predict S4:

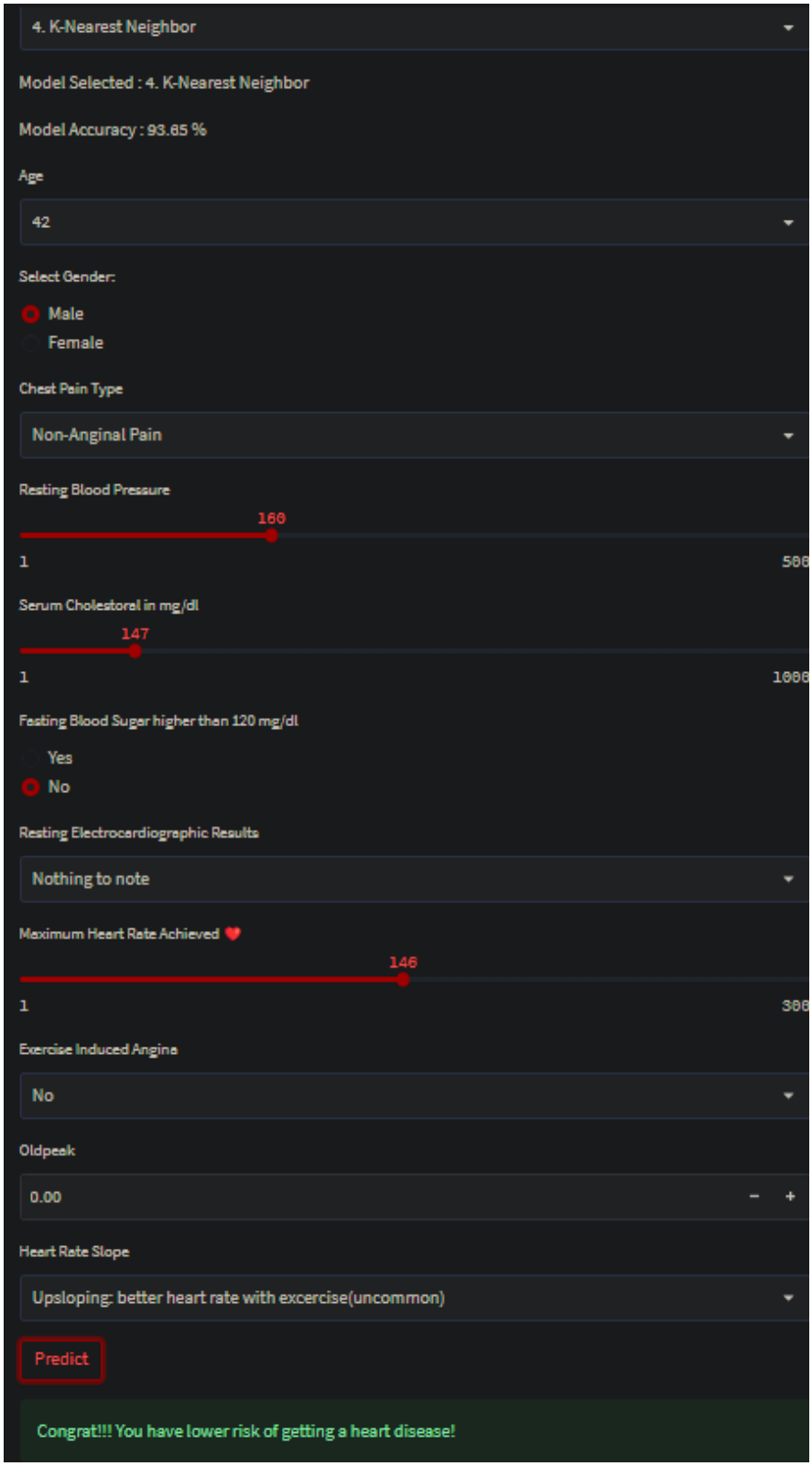

Figure 45: Results after predict S4

Result after predict S5:

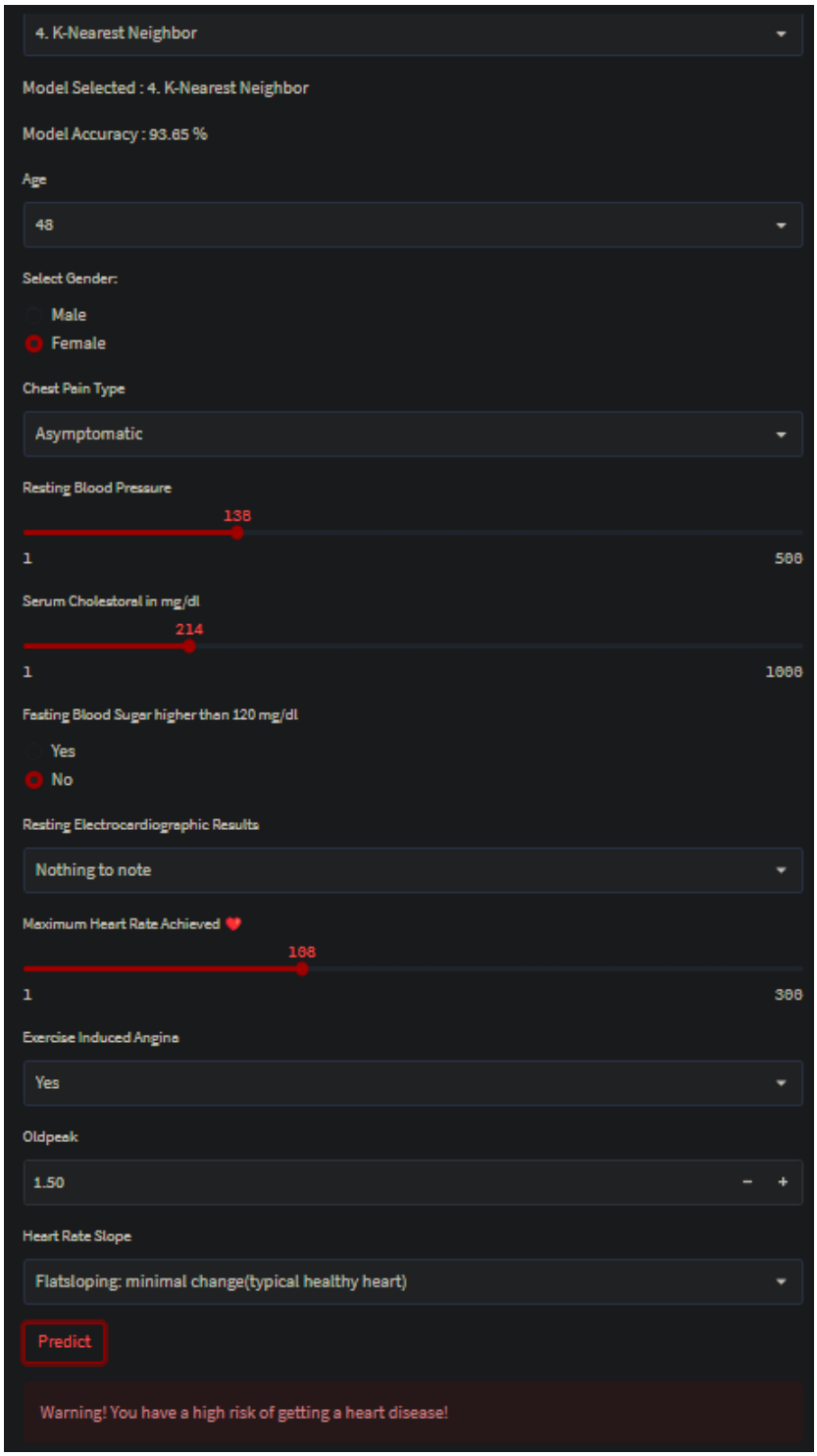

Figure 46: Results after predict S5

Result after predict S6:

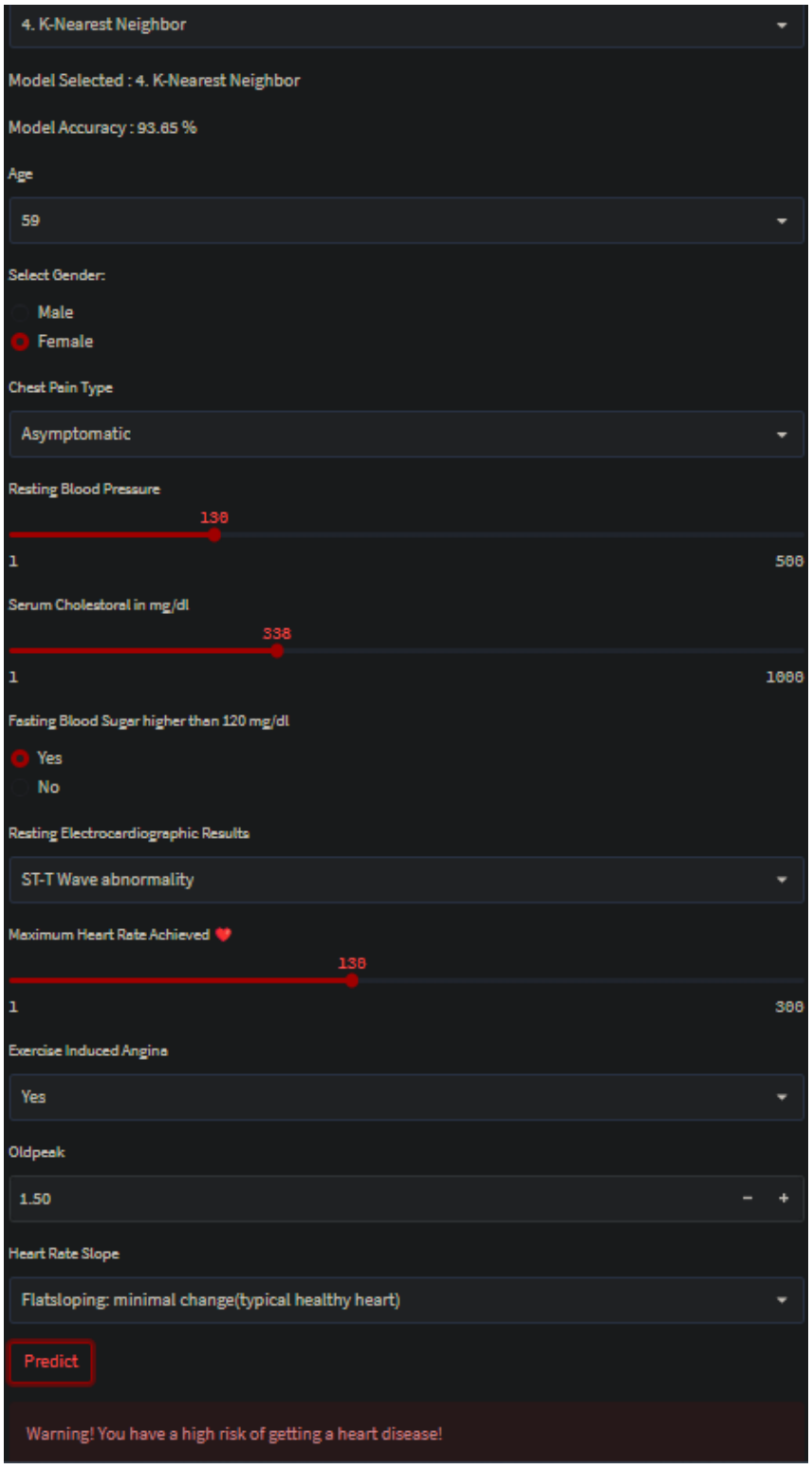

Figure 47: Results after predict S6

Result after predict S7:

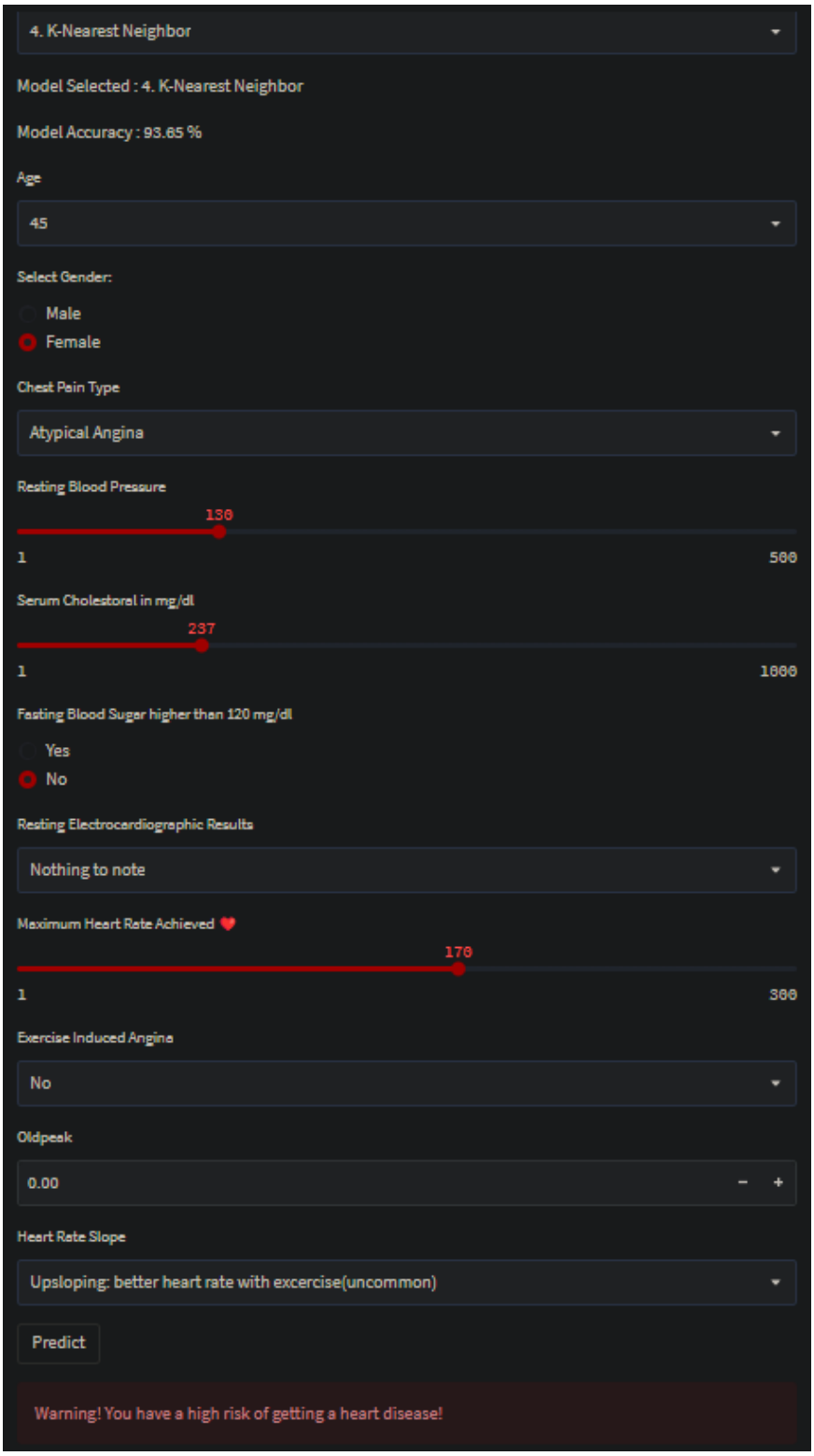

Figure 48: Results after predict S7

Result after predict S8:

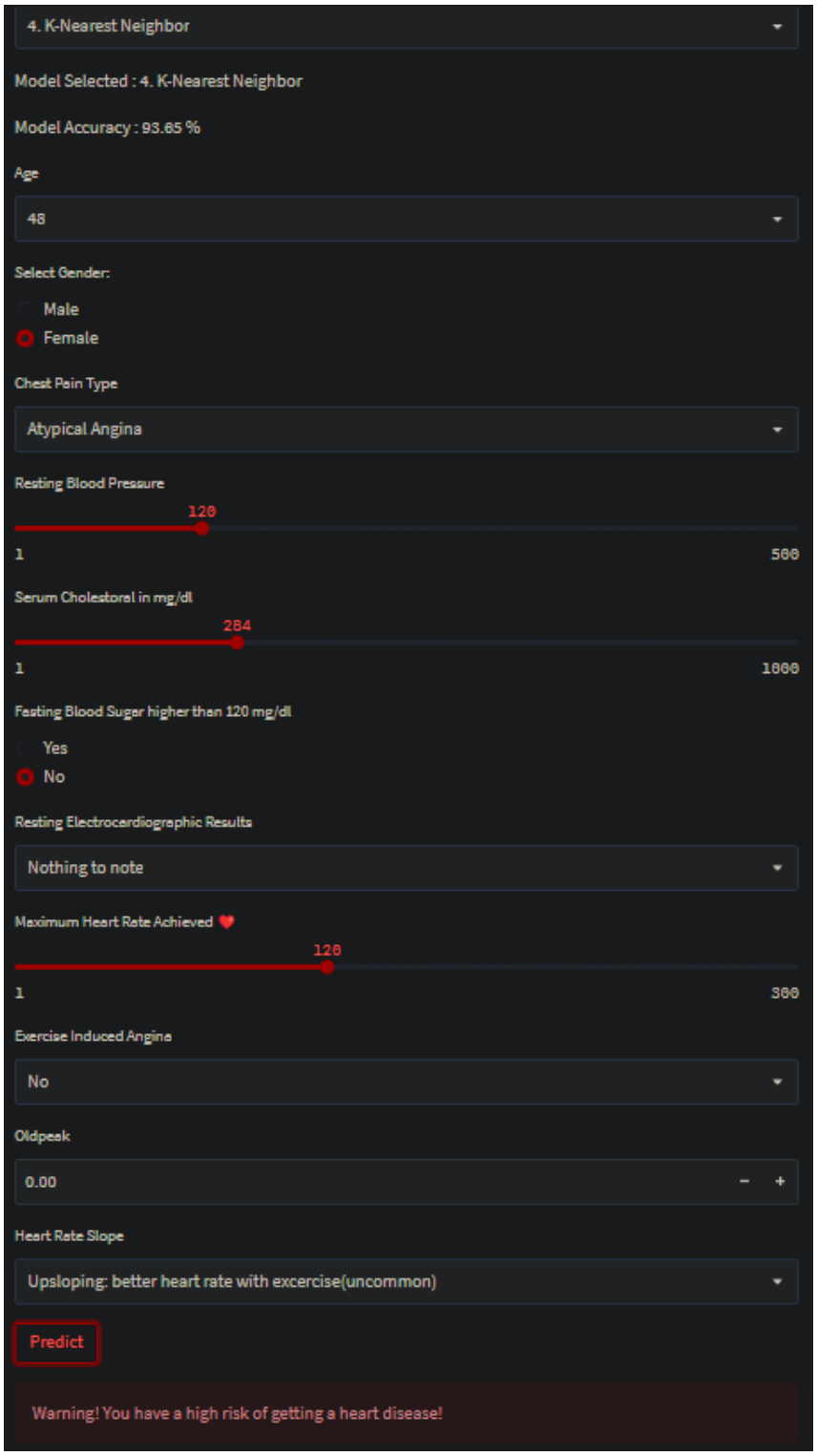

Figure 49: Results after predict S8

4.2. Discussion/Interpretation

In this section, we will be recording the result of the testing hypotheses and make a conclusion out of it.

Hypothesis:

H1)	S1 will be diagnosed as a heart disease patient after predicting using the K-Neartest Neighbor model (KNN).
H2)	S2 will be diagnosed as a heart disease patient after predicting using the K-Neartest Neighbor model (KNN).
H3)	S3 will be diagnosed as a non heart disease patient after predicting using the K-Neartest Neighbor model (KNN).
H4)	S4 will be diagnosed as a non heart disease patient after predicting using the K-Neartest Neighbor model (KNN).
H5)	S5 will be diagnosed as a heart disease patient after predicting using the K-Neartest Neighbor model (KNN).
H6)	S6 will be diagnosed as a heart disease patient after predicting using the K-Neartest Neighbor model (KNN).
H7)	S7 will be diagnosed as a heart disease patient after predicting using the K-Neartest Neighbor model (KNN).
H8)	S8 will be diagnosed as a heart disease patient after predicting using the K-Neartest Neighbor model (KNN).

In Figure 42, we predicted the input data as:

S1	58	Male	Atypical Angina	136	164	No	ST-T Wave abnormality	99	Yes	2.00	Flatsloping

The result of the inputted data is “Warning! You have a high risk of getting a heart attack!”. Therefore, we are not rejecting H1 and can conclude that S1 is prone to heart disease after predicting using the K-Nearest Neighbor (KNN) model.

In Figure 43, we predicted the input data as:

S2	70	Male	Asymptomatic	170	192	No	ST-T Wave abnormality	129	Yes	3.00	Downsloping

The result of the inputted data is “Warning! You have a high risk of getting a heart attack!”. Therefore, we are not rejecting H2 and can conclude that S2 is prone to heart disease after predicting using the K-Nearest Neighbor (KNN) model.

In Figure 44, we predicted the input data as:

S3	52	Male	Atypical Angina	140	100	No	Nothing to note	138	Yes	0.00	Upsloping

The result of the inputted data is “You have lower risk of getting a heart disease!”. Therefore, we are not rejecting H3 and can conclude that S3 is prone to heart disease after predicting using the K-Nearest Neighbor (KNN) model.

In Figure 45, we predicted the input data as:

S4	42	Male	Non-Anginal Pain	160	147	No	Nothing to note	146	No	0.00	Upsloping

The result of the inputted data is “You have lower risk of getting a heart disease!”. Therefore, we are not rejecting H4 and can conclude that S4 is not prone to heart disease after predicting using the K-Nearest Neighbor (KNN) model.

In Figure 46, we predicted the input data as:

S5	58	Male	Atypical Angina	136	164	No	ST-T Wave abnormality	99	Yes	2.00	Flatsloping

The result of the inputted data is “Warning! You have a high risk of getting a heart attack!”. Therefore, we are not rejecting H5 and can conclude that S5 is prone to heart disease after predicting using the K-Nearest Neighbor (KNN) model.

In Figure 47, we predicted the input data as:

S6	59	Female	Asymptomatic	130	338	Yes	ST-T Wave abnormality	130	Yes	1.50	Flatsloping

The result of the inputted data is “Warning! You have a high risk of getting a heart attack!”. Therefore, we are not rejecting H6 and can conclude that S6 is prone to heart disease after predicting using the K-Nearest Neighbor (KNN) model.

In Figure 48, we predicted the input data as:

S7	45	Female	Atypical Angina	130	237	No	Nothing to note	170	No	0.00	Upsloping

The result of the inputted data is “Warning! You have a high risk of getting a heart attack!”. Therefore, we are rejecting H7 and can conclude that the output for S7 was not accurate as the K-Nearest Neighbor (KNN) model has some flaws in it as the accuracy score was slightly lower than the StackCV model.

In Figure 49, we predicted the input data as:

S8	48	Female	Atypical Angina	120	284	No	Nothing to note	120	No	0.00	Upsloping

The result of the inputted data is “Warning! You have a high risk of getting a heart attack!”. Therefore, we are rejecting H8 and can conclude that the output for S8 was not accurate as the K-Nearest Neighbor (KNN) model has some flaws in it as the accuracy score was slightly lower than the StackCV model.

According to the above results, we can conclude that although not all the outcomes from the K-Nearest Neighbor (KNN) model are exactly matched with the hypothesis that was stated in Section 3.4, it is no doubt that K-Nearest Neighbor (KNN) model is still a reliable model to use. Even though the K-Nearest Neighbor (KNN) classifier has a high accuracy score, it will still have a chance of predicting wrongly.

5. Discussion and Conclusion

5.1. Achievements

Throughout this assignment, we have studied different types of machine learning algorithms and how they are being applied into our heart disease prediction application. By learning multiple algorithms, we are able to differentiate between the benefits and flaws of each algorithm. The algorithms that were used in our assignment are K-Nearest Neighbor, Random Forest and StackingCV. We had given an evaluation of these algorithms based on their precision and performance. The evaluation was based on the extraction of a dataset consisting of 3932 data from Kaggle. All the classifier models had demonstrated an excellent performance, however there is an algorithm that stands out the most among all the other models which is StackingCV, with a 97.84% prediction accuracy.

5.2. Limitations and Future Works

Here are the few limitations that we are facing during this project. First one is inconsistencies of data quality. Machine learning algorithms rely on large amounts of data to train models. However, the quality of the data can greatly impact the accuracy of the models. If the data is incomplete, inconsistent, or biased, the machine learning algorithms will produce inaccurate results. Efforts can be made to improve the quality of the data that is used to train machine learning models. This includes methods such as data cleaning and augmentation, as well as the use of more diverse and representative datasets.

Second, lack of Interpretability. Deep learning models can be very complex and difficult to interpret. It can be challenging to understand how a decision was made or to identify the specific features that contributed to the output. This makes it hard to identify errors or to ensure that the model is behaving ethically.Researchers can work on developing more interpretable machine learning algorithms, as well as tools and techniques for explaining the decisions made by complex models. This can include techniques such as feature attribution and visualization.

Third, limited generalization. Machine learning models often struggle to generalize well to new, unseen data. This is particularly true in cases where the training data is not representative of the population the model will be used on. This can result in models that work well in the lab, but fail in real-world settings. Researchers can work on developing algorithms that are better able to generalize to new, unseen data. This can include methods such as transfer learning, as well as techniques for identifying and addressing biases in the training data.

Last limitation we are facing is over-reliance on data. Machine learning algorithms often require vast amounts of data to produce accurate results. This can make it difficult to use machine learning in cases where data is scarce, or where collecting data is time-consuming or expensive.Researchers can work on developing algorithms that are more efficient and effective at learning from small amounts of data. This can include techniques such as active learning and reinforcement learning, as well as the use of synthetic data to supplement real-world datasets.

Reference & Source

1. Brownlee, J. (2014). *A Gentle Introduction to Scikit-Learn*. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/a-gentle-introduction-to-scikit-learn-a-python-machine-learning-library/#:~:text=Scikit%2Dlearn%20was%20initially%20developed.

2. Kaggle (2022). *Kaggle: Your Home for Data Science*. [online] Kaggle.com. Available at: https://www.kaggle.com/.

3. Donges, N. (2021). *A Complete Guide to the Random Forest Algorithm*. [online] Built in. Available at: https://builtin.com/data-science/random-forest-algorithm.

4. JavaTpoint (2021). K-Nearest Neighbor(KNN) Algorithm for Machine Learning - Javatpoint. [online] www.javatpoint.com. Available at: https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning.

5. Khan Academy. (n.d.). Machine learning algorithms (article). [online] Available at: https://www.khanacademy.org/computing/ap-computer-science-principles/data-analysis-101/x2d2f703b37b450a3:machine-learning-and-bias/a/machine-learning-algorithms#:~:te [Accessed 27 Apr. 2023].

6. Sunil, R. (2019). *Understanding Support Vector Machine algorithm from examples (along with code)*. [online] Analytics Vidhya. Available at: https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/.

7. Jindal, H. et al. (2021) "Heart disease prediction using machine learning algorithms," in IOP Conference Series: Materials Science and Engineering. IOP Publishing Ltd. doi:10.1088/1757-899X/1022/1/012072.

8. FTM. 2021. [online] Available at: <https://www.freemalaysiatoday.com/category/nation/2021/11/16/heart-diseases-remain-the-top-killer-in-malaysia/>

9. CodeBlue (2020). *Nearly Half Of Malaysians Lack Health Coverage Beyond Public Care*. [online] CodeBlue. Available at: https://codeblue.galencentre.org/2020/06/02/nearly-half-of-malaysians-lack-health-coverage-beyond-public-care/.