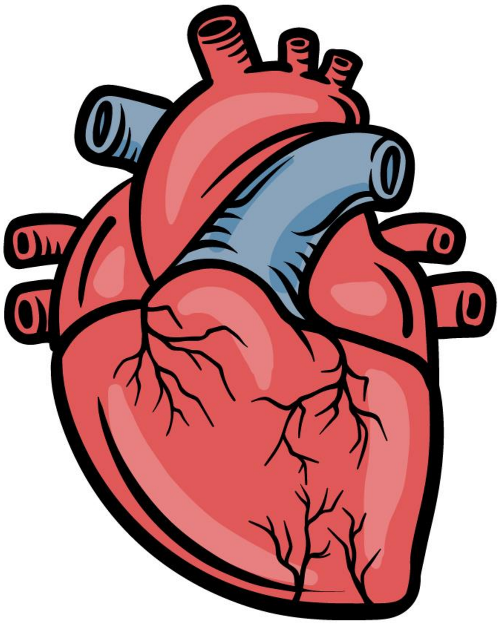


COMP5310 –
PROJECT STAGE 3

Predictive Model & Final Report

Section 1: Setup

The research questions this report will explore is encapsulated from the Heart Disease Prediction dataset. Here we will explore:

- What is the most effective classification model for predicting heart disease using the dataset, and how does its performance compare to other popular machine learning algorithms?
- What are the most important factors in predicting the presence of heart disease, and how can these factors be incorporated into a predictive model?
- Can we enhance the interpretability of the classification model for heart failure prediction by incorporating explainable rule-based models?

The null and alternative hypotheses involve all three research questions, but to highlight an example:

$$H_0: There\ are\ no\ significant\ factors\ in\ predictng\ presence\ of\ heart\ disease$$

$$H_1: There\ are\ significant\ factors\ in\ predicting\ presence\ of\ heart\ disease$$

The appropriate approach involves constructing and evaluating various matching learning models to determine the most accurate one for predicting results. Various classification models have been implemented including kNN, Decision Tree, Support Vector Machine (SVM), Naïve Bayes and Random Forest. Measuring classification effectiveness includes Accuracy, Precision, Recall and F1-score. Initial inspection of the data is deployed via data transformation and EDA, with machine learning models following afterwards and a comparison showcasing the most effective techniques to predict presence of heart disease.

Section 2.1: Approach

I propose to employ a machine learning approach using various classification algorithms to develop a prediction model. The goal is to accurately predict the presence of heart disease based on the available features in the dataset. To achieve this, I would pre-process the dataset by handling missing values, normalising numerical values, and encoding categorical variables if necessary. The aforementioned techniques are commonly used in medical prediction tasks and have shown promising results in similar studies. In order to establish a benchmark for comparison, considering a simple baseline model, such as a Naïve Bayes classifier or a basic logistic regression model. These models serve as a starting point to assess the performance of more complex algorithms. Comparing the results of the benchmark model with the performance of the proposed models would provide insights into the effectiveness of the advanced machine learning techniques in predicting heart disease.

Additionally, exploring feature selection techniques to identify the most relevant features that contribute significantly to the prediction task is useful. This is because the analysis would assist in understanding the importance of different variables and potentially improve the model's performance by reducing dimensionality and removing irrelevant features. To instantiate a high performing model, hyperparameter tuning is essential in controlling the behaviour of a machine learning model during the learning process. Advantages of hyperparameter tuning goes beyond just attaining model efficiency, we can employ more complexity and flexibility of a model, whether that revolves around striking the right balance between underfitting (high bias) and overfitting (high variance).

Section 2.2: Logistic Regression

Logistic regression is a calculation used to predict a binary outcome, for instance a patient having heart disease or not. The predictors are analysed to determine the binary outcome with the results falling into one of two categories. These variables can be categorical or numeric, but the dependent variable is always categorical. This method calculates the probability of dependent variable Y, given independent variable X:

$$P(Y = 1|X)\ or\ P(Y = 0|X)$$

This supervised learning technique will allow us to compute a classification problem to the heart disease dataset, considering all the variables we use, we can predict their effectiveness on the outcome variable, based on evaluation metrics, that will be covered in the results section.

Section 2.3: kNN Classification

K-Nearest Neighbours (kNN) classification is a machine learning technique used for classifying data points into different classes based on their feature values. This algorithm determines the class label of a new, unlabelled data point by considering the classes of its K nearest neighbours in the feature space. This method assumes that the data points with similar features tend to belong to the same class. KNN assigns a class label to a new, unlabelled data point based on the class labels of its k nearest neighbours in the feature space. The number k represents the user-defined parameter that determines the number of neighbours to consider. We will analyse the choice of the parameter k since it is crucial and can impact the algorithm’s performance. A small k value can lead to more flexible decision boundaries but may be sensitive to noise, while a large k value can provide smoother decision boundaries but may overlook local patterns.

Section 2.4: Decision Tree Classifier

A decision tree is a supervised learning technique can order classes on a precise level, which is ideal for a classification problem we are exploring here. The hierarchical arrangement of decision nodes and leaf nodes can be easily understood and visualised. This allows us to explain and communicate the reasoning behind the classification outcomes. The heart data set in this report explores various numerical and categorical features and although we have undergone data pre-processing to handle different data types, decision trees can be performed without extensive pre-processing and thus able to handle both numerical and categorical features. Furthermore, decision trees are relatively robust to outliers and missing values, this is due to the tree structure’s ability to split data into different branches, isolating extreme values. Missing values can also be accommodated by evaluating alternative paths based on available features. The random forest classifier explanation has been omitted to save space.

Section 2.5: Support Vector Machine (SVM)

A Support Vector Machine (SVM) is an unsupervised algorithm used to train and classify data within degrees of polarity, taking it to a degree beyond simply predicting the outcome variable and predictors. SVM is a classification technique that assigns a hyperplane that best separates data points. The goal of SVM is to maximise the margin, which is the distance between the decision boundary and the nearest data points of each class. These nearest data points are called support vectors, and they play a crucial role in determining the position of the decision boundary. SVM is commonly used for handling complex decision boundaries, including non-linear and high-dimensional data. To avoid problems of overfitting we can counter these issues by generalising unseen data and avoid memorising noise or outliers in the training set using SVM technique.

Section 2.6: Naïve Bayes

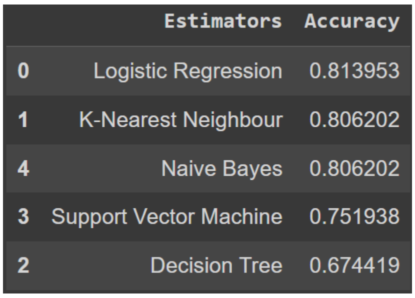
The Naïve Bayes technique is a probabilistic algorithm used for classification tasks. It is based on the Bayes’ theorem and assumes that the features in the data are conditionally independent of each other given the class label. We calculate the probability of a data instance belonging to a specific class based on the observed feature values. It calculates the posterior probability using Bayes’ theorem, which states that the posterior probability of a class given the observed features is proportional to the prior probability of the class multiplied by the likelihood of the features given the class. Naïve Bayes makes a strong assumption of feature independence, meaning it assumes that the presence or absence of a particular feature does not affect the presence or absence of other features. This assumption simplifies the calculations and allows Naïve Bayes to work efficiently even with large feature spaces.

Section 3: Results

Initial inspection of the dataset after completing pre-processing tasks, such as removing outliers, one-hot encoding to transform categorical features to numeric and scanning for any missing values, we begin with Logistic Regression. We fit this classification technique with the training data and predict the outcome. We obtain an accuracy score of 0.814 (3 d.p) indicating a relatively high score for the logistic regression model, however we can extend our analysis by interpreting the confusion matrix as shown in appendix 1.1. The true positive, false positive, false negative and true negative are 26, 12, 12 and 79 respectively. We conclude that true negative has the highest value, which alludes that the model predicts that a patient does not have heart disease and this becomes true. Type 1 and type 2 errors are low with 12 which means we do not incorrectly predict heart disease. Now calculating other evaluation metrics such as precision, recall and F1-score can be computed from the confusion matrix values. Therefore, we obtain 0.684 score for all the metrics stated above. This indicates that we correctly predict the classes, whether they are positive or negative only 68% of the times based on these metrics, besides accuracy which indicates 81% score. Although not very promising results, we can extend this 2-class classification problem using other classifiers to potentially obtain greater accuracy.

Following on, kNN classification is implemented on the heart dataset. To optimise the kNN model, choosing the right k value is essential because it determines the number of nearest neighbours that will be considered when making predictions for a new data point. Manually selecting a k value may not result in the most optimal evaluation metrics in our classification report. Before we calculate these metrics, we produce a plot of the k-values between a certain range and select the value which obtained the greatest accuracy, displayed in appendix 1.2. As shown, the highest accuracy of 0.806 is obtained from a k value of 3, thus we will be computing the evaluation metrics using k = 3 neighbours. Appendix 1.3 showcases the confusion matrix of the kNN classification model which we have deployed using k value of 3 with very similar results to logistic regression. The precision score is 0.658, recall score of 0.676 and F1-score of 0.667. Analogous to the previous logistic model, we obtain metrics which are slightly lower, indicating kNN is less effective, based on all accuracy, precision, recall and F1-score.

The decision tree classifier has been implemented as another classification technique for the heart dataset. Here we seem to be obtaining much more different results compared to logistic regression and kNN classification, with an accuracy score of 0.674. Using a random state of 42 is chosen randomly, however following on for the other classification models, we have kept this value consistent. The confusion matrix is shown in appendix 1.4 with precision, recall and F1-score of 0.464, 0.684 and 0.553. Comparatively, these scores are much lower than their corresponding counterparts. An explanation for these lower values can be due to decision trees generally overfitting the training data, especially when the model captures noise or idiosyncrasies of the training data, resulting in poor generalisation to unseen data. Another reason may be due to decision trees being insensitive to the scale of the features, which means they can handle datasets without requiring explicit feature scaling. In contrast, kNN and logistic regression can be affected by the scale of the features. If the features have different scales, kNN and logistic regression might benefit from appropriate feature scaling, which can improve their performance compared to decision trees in such cases. To maintain a concise report, we have omitted further deep analysis of the other classification methods mentioned in the approach section. To answer the research question of identifying the best estimators using classification algorithms we have compared a multitude of evaluation metrics including precision, recall and F1-scores, however we will be focusing on the accuracy scores as shown below:



	Estimators	Accuracy
0	Logistic Regression	0.813953
1	K-Nearest Neighbour	0.806202
4	Naive Bayes	0.806202
3	Support Vector Machine	0.751938
2	Decision Tree	0.674419

Here we observe that the logistic regression model performs the best with 0.814, followed by kNN and Naïve Bayes showing the same accuracy scores. Logistic regression can perform well in terms of accuracy scores in certain scenarios due to several factors. These including the linear separability which allows the data to be well separated by a linear decision boundary, logistic regression can accurately classify the instances, resulting in higher accuracy scores. Furthermore, regularisation such as L1 or L2 can prevent overfitting and logistic regression allows for this. Helping reduce the impact of irrelevant or noisy features, preventing he model from fitting the noise in the data. By avoiding overfitting, logistic regression can generalise well to unseen data, leading to higher accuracy scores. We have also implemented a Grid Search method on logistic regression to determine the best possible accuracy score based on hyperparameter tuning and we see that the accuracy score of 0.814 is the greatest performance metric we can achieve from emulating a classification task on the heart dataset.

One other method we have not included in our comparison table is the random forest classifier. Constructing this model requires us to understand what the best number of estimators is to obtain the highest accuracy as we have tuned in the previous methods. As displayed in appendix 2.1, we observe how the accuracy is affected by the increasing number of estimators from the random forest classifier. According to this plot we observe the highest accuracy of 0.775 for 13 estimators, thus we utilise that in our random forest model. When comparing to the table above, random forest classifier performs better than SVM and decision tree, however less effective compared to logistic regression, kNN and Naïve Bayes methods. To extend our analysis of evaluating these models, in appendix 2.2, we have implemented a ROC curve which is useful for comparing performance of binary classification models. This plot displays the true positive rate (TPR) on the y-axis against the false positive rate (FPR) on the x-axis. This ROC curve can choose an appropriate classification threshold based on predicting the heart disease for various patients. Based on the left corner of the curve, visually the performance is not very high, and this is reinforced from the Area Under the Curve (AUC) which is 0.677. Although the AUC is greater than 0.5, based on the value we achieved, the classifier achieves poor separation between positive and negative cases.

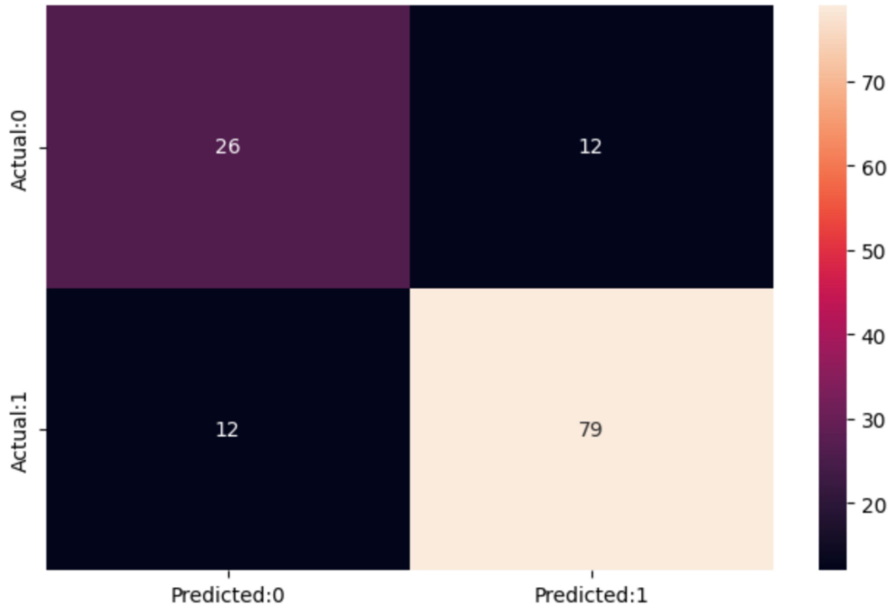
To complete our analysis, answering the research question of determining the most significant factors in determining heart disease and reject or favour the null hypothesis, we compute the feature importance of the dataset. The feature importance of the dataset can be visualised as shown in appendix 2.3. The plot showcases that ExerciseAngina predictor performs the best with a feature importance score of 0.28 with FastingBS, RestingECG being the numeric variables with having 0 importance scores. Therefore, based on this plot and the values, we confirm the most important features that will determine HeartDisease outcome.

To conclude, from all the stages of this project, I have learnt how to analyse a large dataset with various classification models, understanding their mechanisms and comparing them, reinforcing my own knowledge. In addition, the skills, and key learnings I have achieved from this includes data visualisations such as the violin plots that I have utilised for the first time and interpreted and being more proficient in python programming. I have also learnt valuable concepts in machine learning, especially how to implement Naïve Bayes which my knowledge was construed to theoretical understandings, instead of implementation. After completing this project, I would still not recommend this solution to the problem, because I think more data pre-processing, such as handling outliers in a more effective manner was needed. Comparison of the evaluation metrics was constructed well, however there are some key questions we require to ask for further validation, such as: how does the AUC curve compare to different classifiers? What happens if we construct hyperparameter tuning for all the models using Grid Search then compare the accuracies? Would that change the results? Finally, feature importance does not necessarily answer the research question entirely, it is merely a exploratory data analysis technique which should have been utilised in previous stages of the project, but for the sake of answering one of the research questions, it was created.

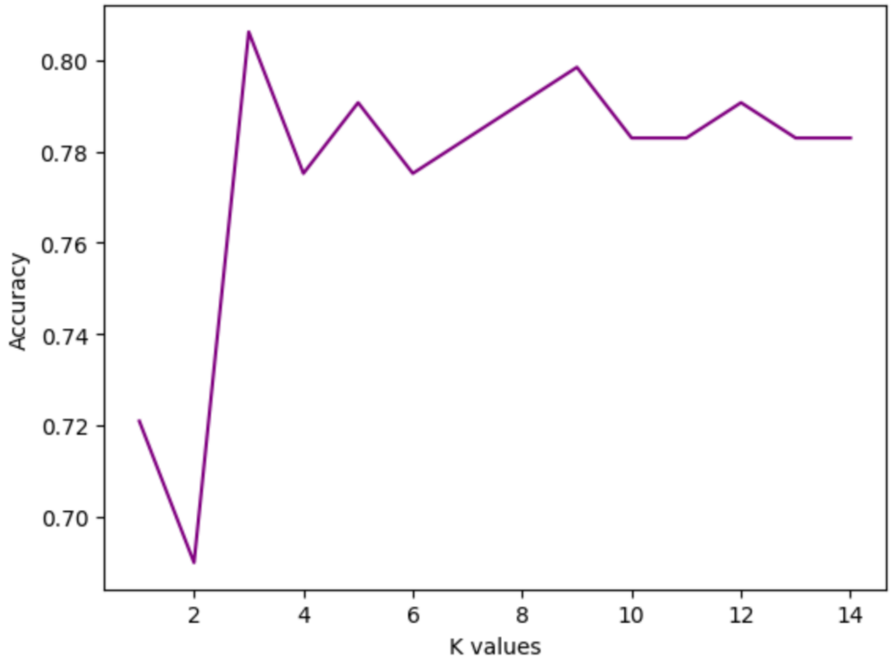
Further analysis was required in order to reject or favour the null hypothesis.

Appendix

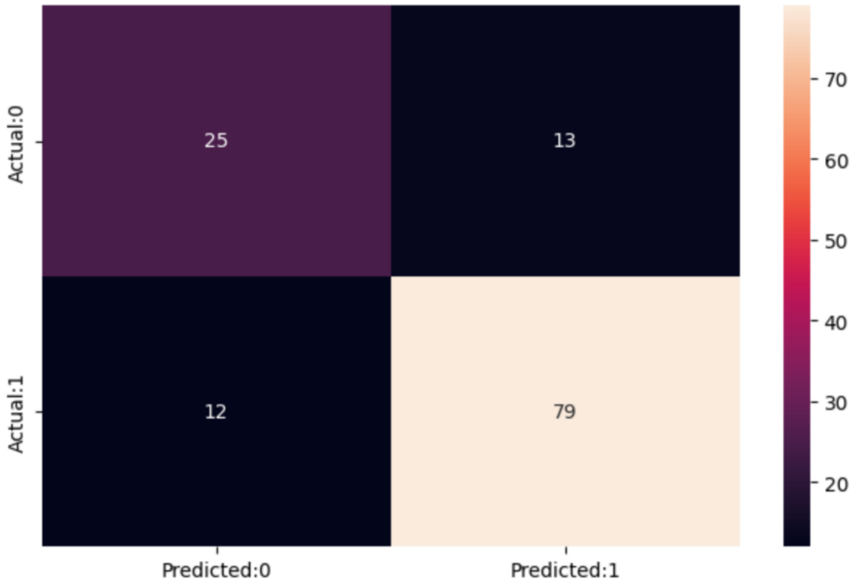
Appendix 1.1: Confusion matrix of Logistic Regression



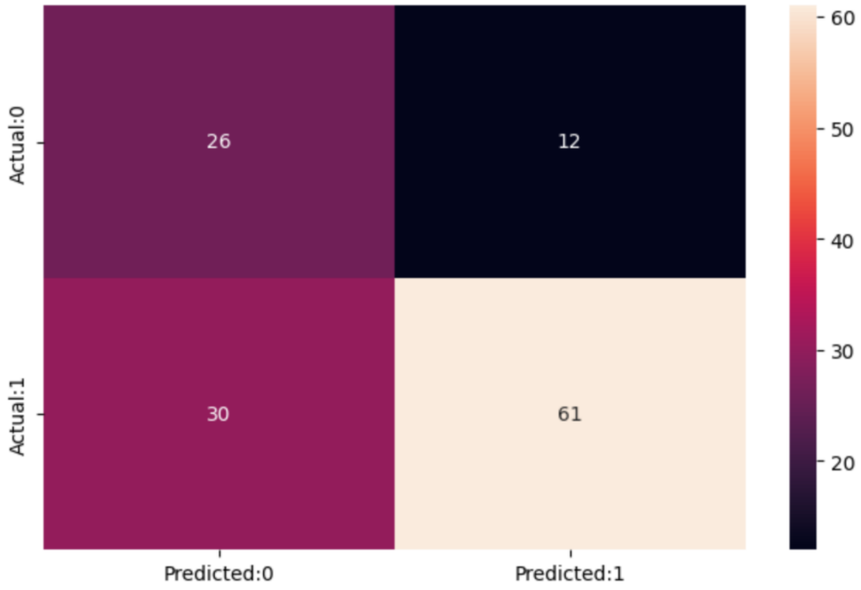
Appendix 1.2: Plot to determine k-values based on accuracy achieved using kNN



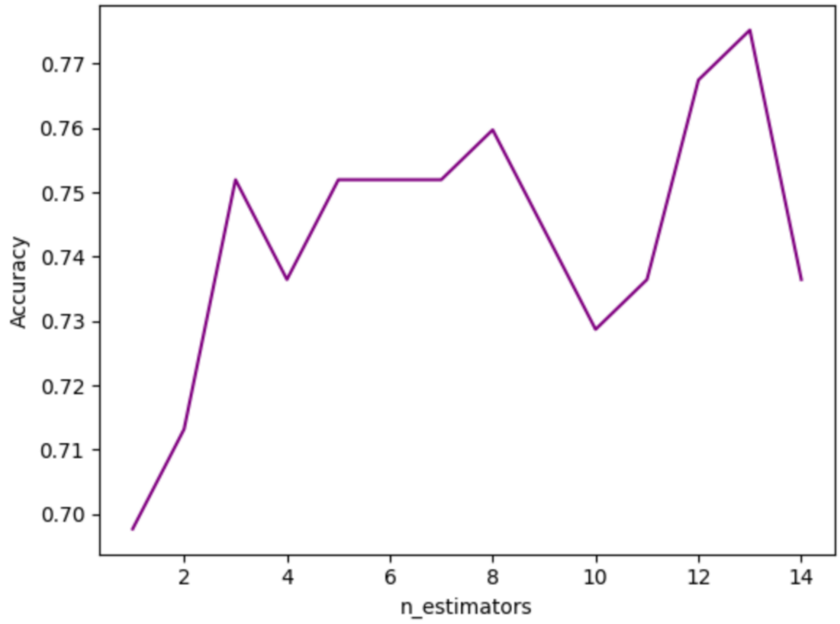
Appendix 1.3: Confusion matrix of kNN model



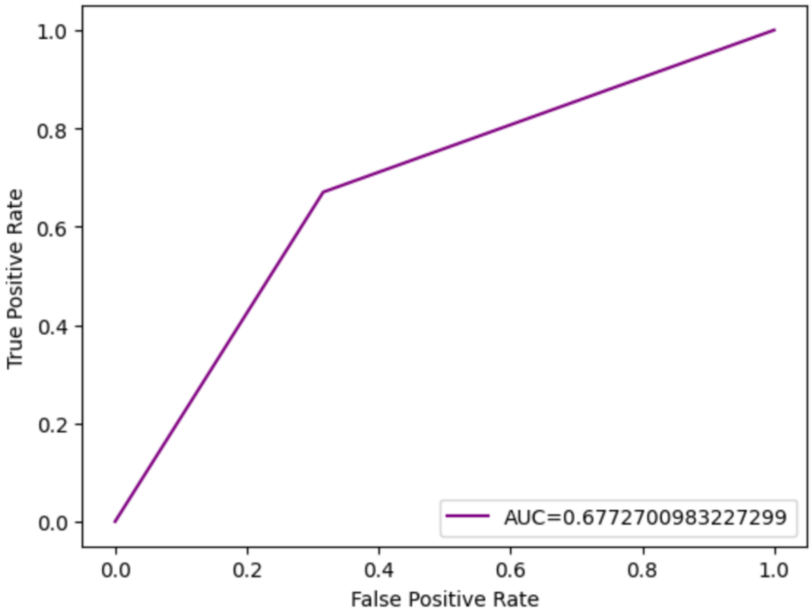
Appendix 1.4: Confusion matrix of decision tree classifier



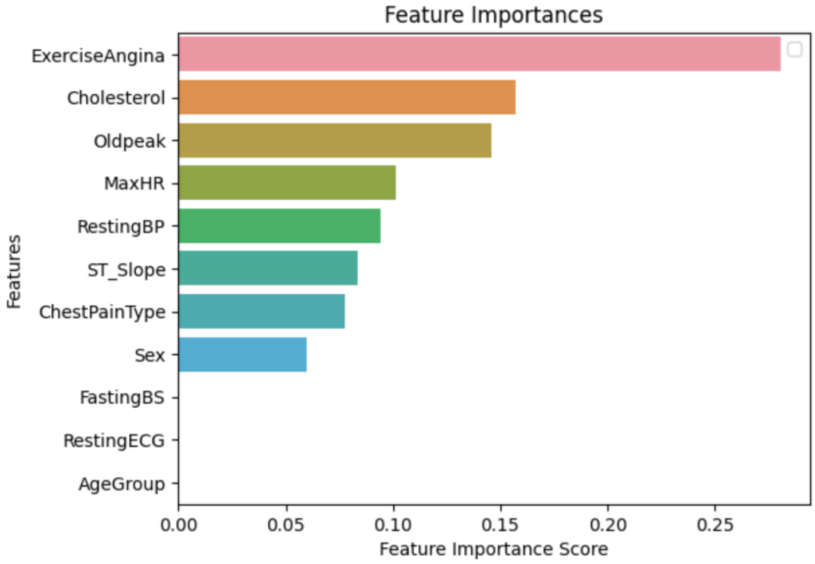
Appendix 2.1: Accuracy vs number of estimators for a random forest classifier



Appendix 2.2: ROC Curve of classifiers



Appendix 2.3: Plot for feature importance in heart dataset


Feature Importances
ExerciseAngina
Cholesterol
Oldpeak
MaxHR
RestingBP
ST_Slope
ChestPainType
Sex
FastingBS
RestingECG
AgeGroup
Features
Feature Importance Score