Data Mining and Machine Learning – SIT307

Machine Learning Challenge (Group 8)

Full Name	Student ID	Contribution
Betty Yuliani	220214569	5
Daniel Agbay	220224024	5
Ekam Behl	219408012	5
Gia Phu Tran	220344366	5
Owen Tsao	220242653	5

DATASET NAME: HEART DISEASE PREDICTION

BURWOOD CAMPUS

CITATION STYLE: HARVARD

1 Table of Contents

2 INTRODUCTION

2.1 Exploratory Data Analysis Conclusions

The heart disease prediction dataset was collected over a cardiovascular study to predict the risk of heart disease development. With over 4000 records and 14 attributes of heart disease determinants including demographic aspects, behavioral aspects, medical history, and current medical conditions, data analysis and visualization were performed rigorously to test out the hypothesis. The observations derived includes:

- No obvious correlation between education level and tobacco usage, where patients with level 3 education has the least cigarettes consumption
- Males have higher frequency of coronary heart disease development
- Systolic and diastolic pressure have a clear positive correlation
- The correlation between age and cholesterol has a steady positive trend
- Glucose and diabetes are highly correlated
- Patients aged 30-40 have higher tendency to smoke
- The lower the education level, the higher the frequency of heart disease development within the samples
- There is a higher average consumption of cigarettes for patients with heart disease than patients without heart disease
- No relationship between BMI and heart disease
- Having more medical issues doesn't necessarily increase the risk of developing heart disease
- The older the patient is, the higher the frequency of heart disease sufferer within the samples

2.2 Machine Learning Problem Formulation

2.2.1 Problem Statement

As the independent variable of the heart disease dataset is a binary attribute, the main aim of building the machine learning model is to solve the **classification problem** for whether a patient has the risk of developing heart disease within 10 years' time (1 for yes and 0 for no).

2.2.2 Algorithms Deployment

Numerous machine learning algorithms were utilized to recognize patterns that occur across the dataset according to (Agrawal, et al., 2020). The algorithms that were deployed are as follows.

1. **Logistic Regression**

   Logistic regression is one of the statistical analysis models that is used in supervised machine learning, and is often known as a logit model according to (IBM, n.a). It represents the relationship between attributes with natural logarithmic function through the concept of odd ratios and probabilities, to generate predictive analysis and modelling.

2. **Decision Tree Classifier**

Unlike other supervised learning algorithms, this model can be used to solve regression and classification problems according to (Singh C, 2022). This is achieved through the use of simple learning decision rules on the training data, where a tree representation model is created. For every attribute, comparison is done starting from the root of the tree, where the result shall determine which branch of the tree to follow. Therefore, this model is also used to predict the value of the target variable ("TenYearCHD") by learning its decision rules that's inferred from the training data provided.

3. **KNN Algorithm**

   As KNN algorithm delivers very precise predictions, this algorithm can compete with the most accurate models. As a result, the KNN method may be used for applications that need high accuracy but do not require a human-readable model, such as the heart disease prediction dataset (IBM, 2021).

4. **Random Forest Classifier**

   Random forest classifier provides an effective way to handle missing data and solves the issue of overfitting. Often, in prediction modeling, the goal is to reduce the number of variables needed to obtain a prediction which reduces the burden of data collection (Speiser, 2019). As such, this classifier helps in deciding variables that are important to be used to infer the data and prove the hypothesis true. For instance, if the attribute 'age' is proven to contribute to the development of heart disease, this attribute can then be used to support the hypothesis.

5. **Support Vector Machine**

   SVM can produce higher accuracy as compared to other techniques as it works well with high dimensional dataset as said in (Ganhi, 2018). The heart disease dataset, having 14 attributes, will be able to make the best out of this machine learning model.

6. **XG Boost**

   The main purpose of using XG boost was to increase the model performance and execution speed (Brownlee, 2016). On top of that, it can generally result in deeper but optimized trees (due to the incremental training). The XG boost model also decreases the bias and helps prevent overfitting.

7. **Naïve Bayes**

   Naïve Bayes is excellent for a fast and easy prediction with less variables and data by assuming independence between all attributes in the dataset. As the observations outlined in section 1.1 suggested some correlations between variables, Naïve Bayes may be beneficial to be used to allow an accurate prediction (Ray, 2017).

2.2.3 Accuracy improvement

Simply deploying algorithms on a dataset may not always yield a very predictive model in the first iteration. Often, the model needs to undergo several improvement processes to get better predictions. A few methods to improve a machine learning's model are as follows.

- **Feature Selection**

  Feature selection is known to be a method that can increase the predictive power of machine learning algorithms by selecting the important features and eliminating redundant and irrelevant features. This process was done in two steps.

- **Feature Scaling**

  Scaling features in a dataset is known to be effective in improving a machine learning model as it smooths the flow of gradient descent and allow algorithms to reach the minima of their cost function quickly. Without feature scaling, algorithms may be biased towards the feature which has values higher in magnitude.

- **Ensemble Learning**

  Ensemble is a machine learning model that combines the predictions from two or more models. It helps to achieve better predictive performance as the model reduces the variance component of the prediction error by adding bias, also known as bias-variance trade off. The idea of ensemble learning is depicted with the following figure.

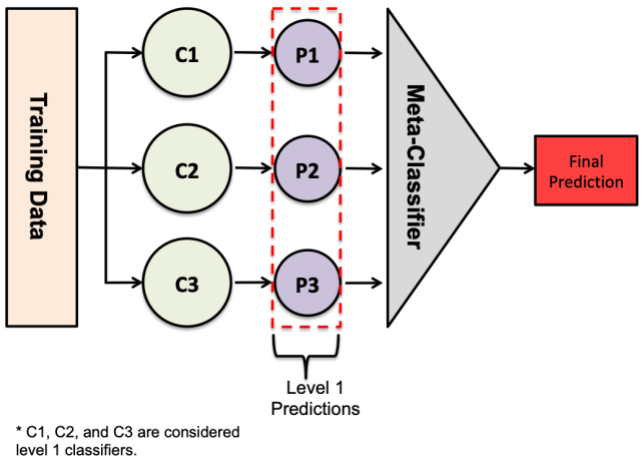

Figure 1: Idea of ensemble learning

2.2.4 Machine Learning Flowchart

Figure 2 depicts our process of building a model for the heart disease prediction dataset.

The dataset will follow the pre-processing stage to perform data cleaning where null values are either removed or imputed, and outliers are handled. The models will then be trained with different machine learning algorithms, with *train dataset* which is a subset of the cleaned dataset. The models will then undergo the evaluation process where its predicted output with *test data* is compared with the actual output.

After the evaluation process, the models will perform several accuracy improvement processes; where it is re-structured by being trained with the dataset that has been modified through different model enhancement methods.

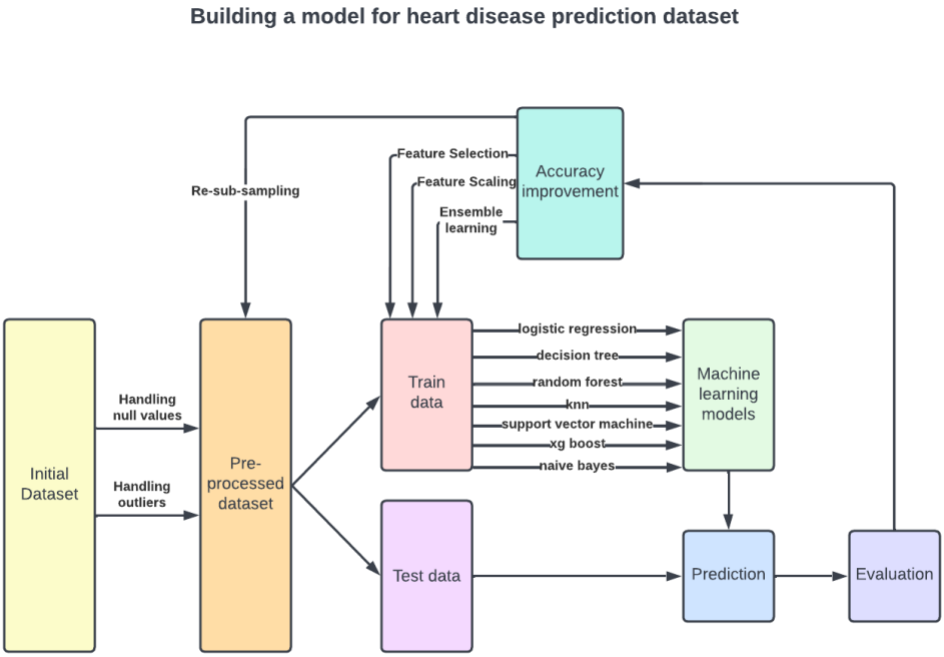

Figure 2: Flowchart of machine learning process

3 RESULTS AND DISCUSSIONS

3.1 Machine Learning Model Analysis

In the first part of the machine learning process, the models mentioned in section 1.2.3 were trained with the heart disease dataset that were cleaned. The outcome was then evaluated with accuracy and confusion.

As the independent variable ("TenYearCHD") were highly unbalanced with most patients in the dataset having no potential of developing coronary heart disease, *True Positive* and *True Negative* values will be significantly lower than True Negative and False Negative. For this reason, precision and recall score are not used as these performance matrices focus on the *True Positive* values.

3.1.1 Logistic Regression

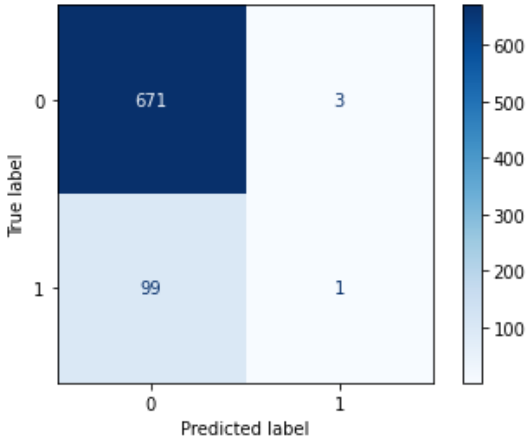

Figure 3: Confusion Matrix results of Logistic Regression model

By observing the confusion matrix of Logistic Regression in *Figure 1*, we notice that 671 out of 674 patients without heart disease were successfully classified. However, only 1% of the positive values were correctly identified by the model. This makes our model accuracy to be 86.82%.

```
#Accuracy score for this model
lr_accuracy = accuracy_score(lr_pred, y_test)*100
print("Accuracy for Logistic Regression Model: " + str(
```

```
Accuracy for Logistic Regression Model: 86.82170542635659%
```

Figure 4: Accuracy of logistic regression model

The accuracy result proves that this model is suitable for our classification problem to predict the binary outcome. However, it is believed that this accuracy can still be improved to a value higher than 86%.

3.1.2 Decision Tree

The decision tree was used for its simplicity and its availability to be applied to either categorical or numerical data, whereas this method were applied to the categorical variables in the dataset. The dataset will be used to improve the output to identify the ideal split from the root node by measuring the impurity of a node in the decision tree. It begins with a single node and branches to potential outcomes using a greedy search of the properties of each node. The decision tree model is depicted as follows.

```
Decision tree in text format :
|--- age <= 48.50
|   |--- cigsPerDay <= 9.50
|   |   |--- glucose <= 233.50
|   |   |   |--- bloodPressure <= 96.00
|   |   |   |   |--- male <= 0.50
|   |   |   |   |   |--- glucose <= 119.50
|   |   |   |   |   |   |--- BMI <= 39.03
|   |   |   |   |   |   |   |--- age <= 39.50
|   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- age >  39.50
|   |   |   |   |   |   |   |   |--- bloodPressure <= 24.50
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- bloodPressure >  24.50
|   |   |   |   |   |   |   |   |   |--- BMI <= 27.41
|   |   |   |   |   |   |   |   |   |   |--- BMI <= 27.05
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 5
|   |   |   |   |   |   |   |   |   |   |--- BMI >  27.05
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- BMI >  27.41
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- BMI >  39.03
|   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- glucose >  119.50
|   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |--- male >  0.50
|   |   |   |   |   |--- totChol <= 258.50
|   |   |   |   |   |   |--- glucose <= 59.50
|   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- glucose >  59.50
|   |   |   |   |   |   |   |--- bloodPressure <= 37.25
|   |   |   |   |   |   |   |   |--- BMI <= 28.65
```

Figure 5: Decision tree in text format

The results show a better prediction outcome for the positive values, where 10% of the patients with heart disease were successfully identified. However, as there is a high amount of false positive, and lesser true negative as compared to logistic regression, the accuracy score is standing at only 83.720930%.

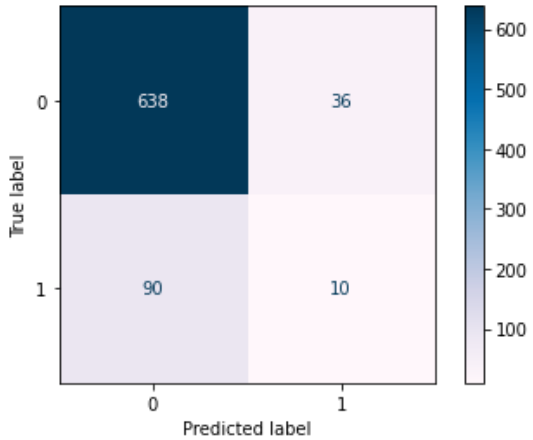

Figure 6: Confusion Matrix results of Decision Tree model

3.1.3 K-Nearest Neighbor

When using KNN algorithm, it is essential to find a suitable K value that could yield the best outcome. This was done with the brute force approach where different k values were tested, and was then then evaluated by calculating the error rate. The number of iterations is limited to the square root of dataset size. With the post-cleaning dataset, the K value that yields the minimum error is 22.

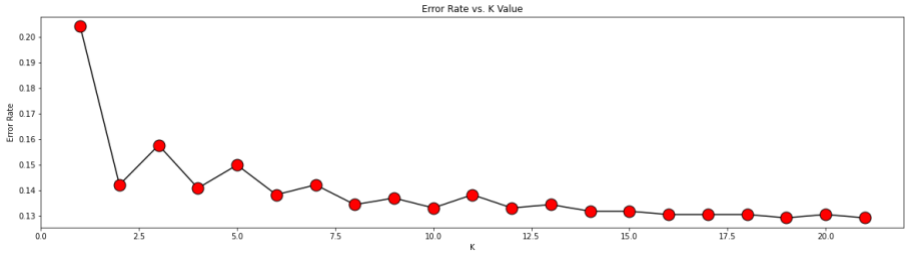

Figure 7: Error rate of different k values

The confusion matrix below shows a fairly similar result to logistic regression, where only 1% of the positive heart disease patient was correctly identified. Up until this point, we can conclude that this result is reasonable due to the unbalanced independent variable which has underfitted the model. Upon multiple run times, the *true positive* value oscillates between 1 to 10, and hence why the accuracy score of 87.209302% was yielded for this model.

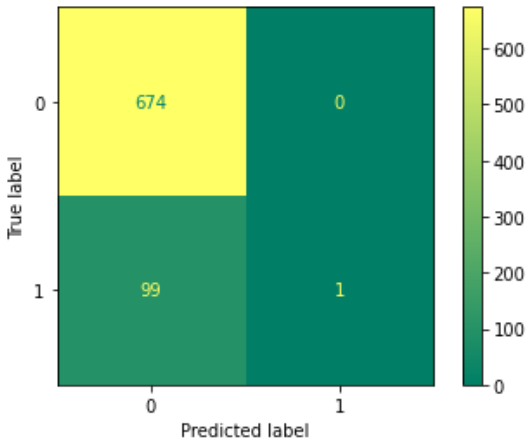

Figure 8: Confusion matrix of kNN algorithm

3.1.4 Random Forest

Random forest classifier is a simplistic algorithm, where it constructs numerous decision trees. Theoretically, this algorithm should reduce the number of incorrect predictions. However, it yielded the same accuracy as KNN algorithm, with an accuracy of 87.209302%.

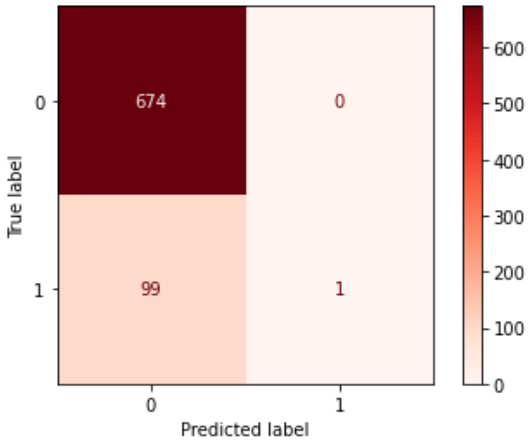

Figure 9: Confusion matrix of random forest

3.1.5 Support Vector Machine

In addition to linear classification, using Support Vector Classifier to do non-linear classification effectively by employing the kernel technique, which involves implicitly mapping their inputs into high-dimensional feature spaces. In this example, SVC performs well with an accuracy score of 87.08% but the True Positive value is 0 which raise questions about the application of the model.

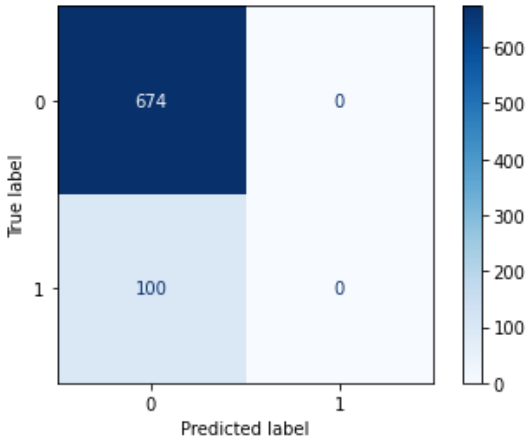

Figure 10: Confusion matrix of SVM

3.1.6 Extreme Gradient Boost Classifier

The Gradient Boosting Decision Trees (GDBT) is a decision tree ensemble learning algorithm used for classification and regression that is slightly like random forest. A gradient-boosted trees model is constructed in the same stage-wise manner as decision tree, but it generalises the other approaches by enabling optimisation of any differentiable loss function. When this classifier is trained with the dataset, none of its predictions yielded positive results, which raises concerns about the model's applicability.

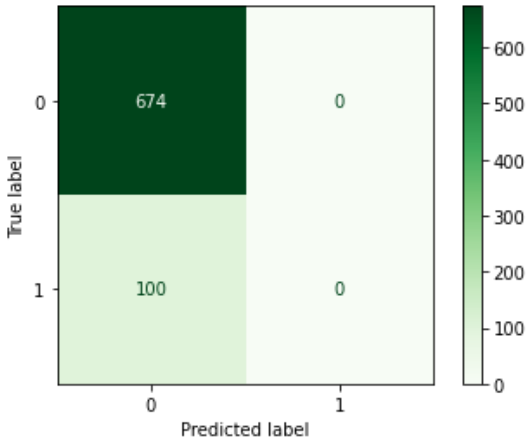

Figure 11: Confusion matrix of XGboost

3.1.7 Naïve Bayes

The Naive Bayes classifier is a type of probabilistic classifier that is based on the famous Bayes' theorem. By training a model with this algorithm, an accuracy score of 85.92% was yielded, and it has the ROC-AUC score of 0.56, which is the best in all our models.

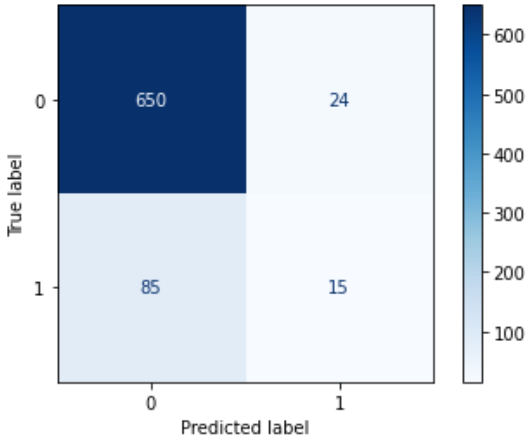

Figure 12: Confusion matrix of naïve bayes model

3.1.8 Models Summary

The table below summarizes the accuracy yielded from the different machine learning algorithms used in the heart disease dataset. The model(s) with the best accuracy are *random forest* and *KNN* where their accuracy stands at 87.2%.

Models	Accuracy %
Logistic Regression	86.821705
Decision Tree	83.720930
Random Forest	87.209302
Support Vector Machine	87.080103
K Nearest Neighbor	87.209302
XG Boost	87.080103
Naive Bayes	85.917313

Table 1: Models Summary Table

AUC and ROC curves are effective in measuring classification performance. The ROC curve depicts true positive against false positive rates across multiple thresholds; the area under the ROC curve is shortened as AUC to determine a classifier's performance. A higher AUC score suggests that it is the superior model. The curve depicts the two parameters, true positives (TPR) and false positives (FP) (FPR). TPR is computed as TP/(TP+FN) and FPR as FN/(FN+TP). The figure below assesses the performance of the categorisation models.

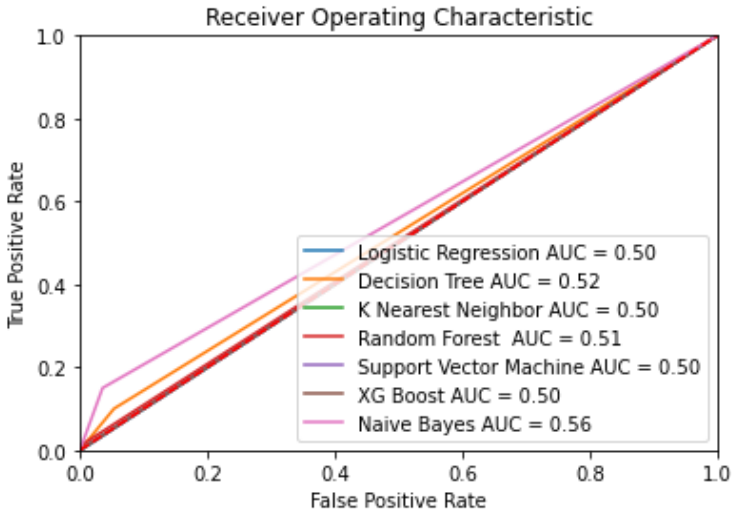

Figure 13: Comparison Between Models

Even though Naïve Bayes has a relatively low accuracy, it is still concluded to be the model that best suit our post-cleaning dataset, for it has the greatest ROC score and highest number of *true positive*. However, it is believed that the ROC and accuracy of other models can be improved through several accuracy improvement processes like feature selection, feature scaling, and cross validation, which will be discussed in the next section.

3.2 Accuracy Improvement Processes

As seen in section 2.1.8, deploying multiple machine learning algorithms on the pre-processed dataset results in an accuracy of just 87.2%. Several accuracy boosting methods were used in hope that a higher highest accuracy among all models can be obtained.

3.2.1 Feature Selection

The process of feature selection was performed in 3 steps. Firstly, the attribute “sysBP” and “diaBP” were merged into a new attribute labelled as “bloodPressure” to better reflect the condition of a patient.

Next, attributes with highly unbalanced data are removed, as introducing such features into the machine learning model may introduce biasness. Looking at the data distribution, 'BPMeds', 'prevalentStroke', 'diabetes' was removed.

Lastly, the k highest score for each attribute was computed. The bar chart below depicts the descending order of attributes that are most relevant to the target variable. It also gives a very important information as it further proves the findings from assignment 1 to be right, where “BMI” does not contribute to the development of heart disease.

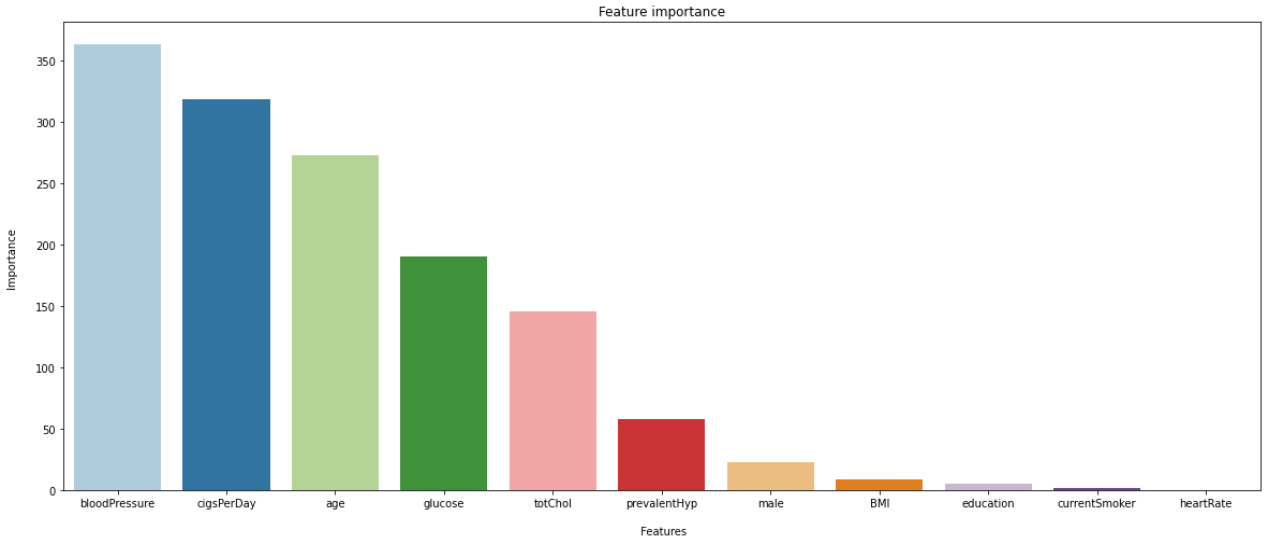

Figure 14: feature importance

Evaluating from the graph, features after “male” were pruned and removed as they have relatively low k scores as compared to the other attributes. On top of that, as blood pressure and hypertension were linearly correlated as stated in section 1, “prevalentHyp” was also removed.

The new dataset with just 6 features were then used to re-train the models. Upon this step, slight improvement of accuracy score was seen on *logistic regression* and *decision tree*. it is also apparent that, from figure 15, the ROC curves were improved, where only SVM and XGBoost algorithm remains their AUC score at 0.5.

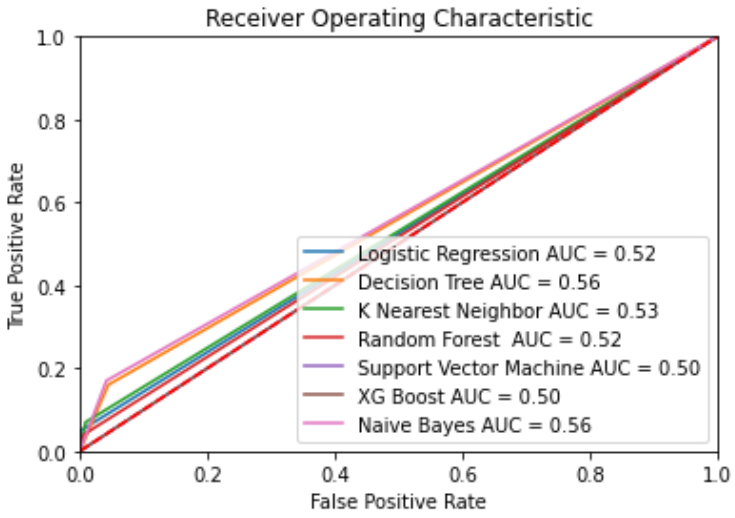

Figure 15: AUC and ROC graph after feature selection

We were also able to obtain a slightly higher accuracy of 87.596% from the *logistic regression* model. This number is a 0.3% increase from the previous highest accuracy prior to feature selection. Given the characteristics of our dataset, this slim increase of accuracy is magnificent.

Logistic Regression	87.596899
Decision Tree	85.271318
Random Forest	87.080103
Support Vector Machine	87.080103
K Nearest Neighbor	87.209302
XG Boost	87.080103
Naive Bayes	85.658915

Figure 16: Accuracy summary after feature selection

3.2.2 Feature Scaling

During the pre-processing step, the outliers in some features were kept for they contain meaningful values. For this reason, the *robust scaler* was chosen as it scales features with statistics robust to outliers, where the median is removed, and data is scaled according to the quantile range. *Standard scaler*, on the other hand, removes the mean and scales data according to unit variance; this may not be optimal given the characteristics of our dataset after the pre-processing step. Upon this step, the values in our dataset are seen as follows.

	cigsPerDay flo…	bloodPressure f…	age int64	glucose float64	totChol float64	male int64	prevalent
4042	0	74	44	193	208	1	
1545	0	51	48	90	193	0	
1578	30	39	38	88	255	0	
438	20	48	36	70	226	1	
1549	0	44.5	40	117	213	1	

Figure 17: dataset after feature scaling

The re-training of our models upon this step yielded the following results. As seen in figure 12, the ROC curves improved from the previous state, where the SVM algorithm no longer have 0.5 AUC score.

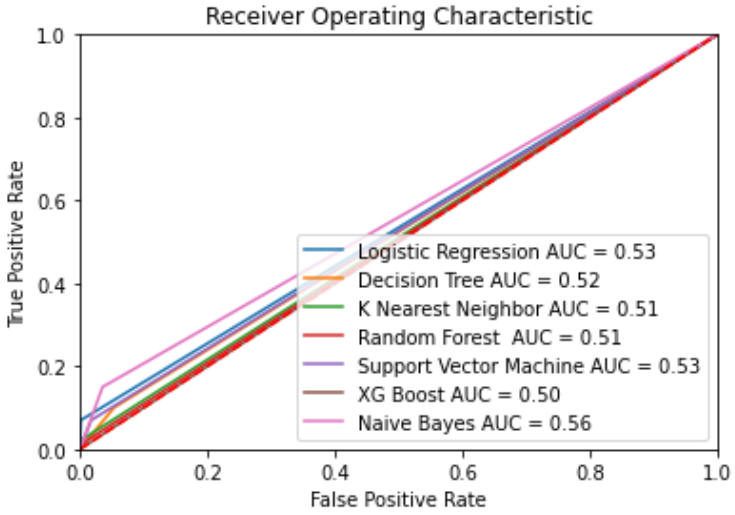

Figure 18: Receiver Operating Characteristic

The accuracy scores on the other hand, is depicted as follows. We were able to obtain another 0.3% increase on *logistic regression*. However, feature scaling actually worked worse on decision tree as it has a decrease of accuracy by 2%.

```
Logistic Regression        87.855297
Decision Tree              83.720930
Random Forest              87.209302
Support Vector Machine     86.434109
K Nearest Neighbor         87.338501
XG Boost                   87.080103
Naive Bayes                85.917313
```

Figure 19: Accuracy summary after feature scaling

3.2.3 Stack CV Classifier

Stacking is an ensemble learning technique to combine multiple classification models via a meta-classifier. The *StackingCVClassifier* extends the standard stacking algorithm (implemented as *StackingClassifier*) using cross-validation to prepare the input data for the level-2 classifier. The *Logistic Regression*, *K Nearest Neighbour*, *Support Vector Machine*, *XG Boost* and *Random Forest* were stacked, with the meta-classifier of *Logistic Regression*.

The algorithm was able to compute an output within 4.5 seconds, with a result of 87.46% accuracy score. However, this number is relatively low compared to the accuracy yielded by the *logistic regression* model after feature scaling.

```
[58]
X_train, X_test, y_train, y_test = train_test_split(fin
from mlxtend.classifier import StackingCVClassifier
scv=StackingCVClassifier(classifiers=[xgb,knn,svm,lr,rf
scv.fit(X_train,y_train)
scv_pred = scv.predict(X_test)
scv_accuracy = accuracy_score(y_test, scv_pred)*100
print("Accuracy of SCV: " + str(scv_accuracy) + "%")

Accuracy of SCV: 87.46770025839793%
```

Figure 20: Accuracy score of SCV

3.2.4 Cross Validation

Cross-validation is used to test models using limited training data, to generate output on model correctness, and to prevent overfitting. The *k-fold cross-validation approach* was deployed to perform this method.

The accuracy summary below shows the significant improvement of performance on all models after the adoption of cross-validation score, where the highest accuracy score stands at 88.37%. There is also a close to 4% increase in *decision tree*'s accuracy, as it was standing at 83% prior to cross validation.

```
Logistic Regression       88.372093
Decision Tree             87.209302
Random Forest             88.242894
Support Vector Machine    88.372093
K Nearest Neighbor        88.372093
XG Boost                  88.372093
Naive Bayes               85.658915
```

Figure 21: Accuracy summary after cross validation

By evaluating the ROC shape of some models, it is also apparent that more *true positive* values were identified.

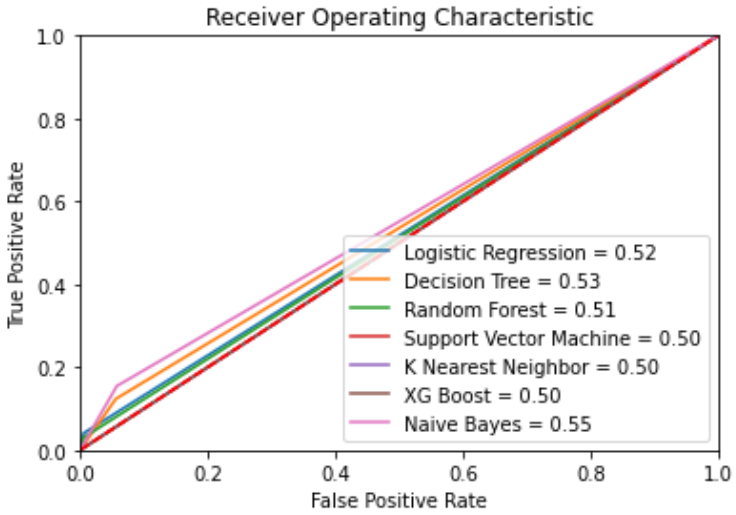

Figure 22: ROC and AUC curve after cross validation

3.2.5 Accuracy Summary

Models	Accuracy Before	Accuracy After Feature Selection	Accuracy After Feature Scaling	Cross Validation
Logistic Regression	86.821705	87.596899	87.855297	88.372093
Decision Tree	83.720930	85.271318	83.720930	87.209302
Random Forest	87.209302	87.080103	87.209302	88.242894
Support Vector Machine	87.080103	87.080103	86.434109	88.372093
K Nearest Neighbor	87.209302	87.209302	87.338501	88.372093
XG Boost	87.080103	87.080103	87.080103	88.372093
Naive Bayes	85.917313	85.658915	85.917313	85.658915

Table 2: Model Summary before and after improvement

4 Conclusion

In this paper, seven classifier models and four improvement methods have been deployed to find the best model for predicting a person's risk of coronary heart disease.

During the first iteration, the classifiers were trained with the raw pre-processed data where only null values and outliers were handled. The accuracy score stands relatively low, 87.2% at maximum. At this stage, it is believed that the Naïve Bayes algorithm best suit the dataset as it yielded a higher AUC score compared to the other models.

The second iteration removed redundant features that are:

- Highly unbalanced
- Having low k-values
- Having dependencies with other features

The classifiers were then re-trained with the newly modified dataset and we were able to obtain a new highest accuracy, 87.5% by the *logistic regression* model. The third iteration scaled the features with *robust scaler*, and the *logistic regression* model was again yielding a higher accuracy results, at 87.8%. At this iteration, it is obvious that *decision tree* classifier is the least favourable for the heart disease prediction, as its accuracy fluctuates only between 83% to 85%.

During the forth iteration, Stack CV Classifier were used, in hope to combine multiple classification models which performed great during the previous iterations to obtain a better accuracy score. Surprisingly, the result stands at only 87.46% accuracy, which is lower than the accuracy we had during the third iteration.

The last iteration deployed the cross-validation subsampling method to restructure the model with an entirely different train dataset and it yielded surprising results; the accuracy of most models stood at 88.4%, which is a 0.6% increase from the third iteration. Even with the *decision tree* classifier, which had an accuracy of 85% at maximum out of all attempts, is now standing at 87%.

Several key takeaways to conclude:

- Feature selection and feature scaling do not always improve accuracy, it depends on the characteristics of the dataset and algorithms used.
- Decision tree models tend to overfit data and hence is prone to sampling errors.
- Logistic regression models can be significantly improved with feature selection and feature scaling.
- Stacked Classifier does not always yield a better accuracy.
- An accuracy increase of as low as 0.3% may be magnificent, depending on the dataset,
- Due to the unbalanced target variable, the AUC score tend to stay low as the model is having trouble in distinguishing positive and negative classes.

Other procedures and implementation should be thoroughly explored and examined to have a better understanding of whether there is a specific component or attributes among the most significant ones that determine the effectiveness of the model.

5 References

Agrawal, A., Gans, J. & Goldfarb, A., 2020. *How to Win with Machine Learning.* [Online]
Available at: https://hbr.org/2020/09/how-to-win-with-machine-learning
[Accessed 10 May 2022].
Brownlee, J., 2016. *A Gentle Introduction to XGBoost for Applied Machine Learning.* [Online]
Available at: https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/
[Accessed 18 May 2022].
Ganhi, R., 2018. *Support Vector Machine - Introduction to Machine Learning Algorithms.* [Online]
Available at: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47
[Accessed 18 May 2022].
IBM, 2021. *Usage of KNN.* [Online]
Available at: https://www.ibm.com/docs/en/db2oc?topic=knn-usage
[Accessed 16 May 2022].
IBM, n.a. *What is logistic regression?.* [Online]
Available at: https://www.ibm.com/topics/logistic-regression
[Accessed 11 May 2022].
Ray, S., 2017. *6 Easy Steps to Learn Naive Bayes Algorithm with codes in Python and R.* [Online]
Available at: https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/
[Accessed 18 May 2022].
Singh C, N., 2022. *Decision Tree Algorithm, Explained.* [Online]
Available at: https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html
[Accessed 11 May 2022].
Speiser, J. L., 2019. *A Comparison of Random Forest Variable Selection Methods for Classification Prediction Modeling.* [Online]
Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7508310/
[Accessed 17 may 2022].