SIT307 – Data Mining & Machine Learning Group Assignment 3 (T1 2020)

# Dataset Name: Heart Disease Dataset from UCI (8)

Group No.: 6 On Campus/cloud: CLOUD

| STUDENT ID | STUDENT FULL NAME | Individual contribution * |
|---|---|---|
| 217218863 | TIM REN | 5 |
| 218515745 | YANG LYU | 5 |
| 218512951 | YIU WONG | 5 |
| - | - | - |

* 5 – Contributed significantly, attended all meetings
4 – Partial contribution, attended all meetings
3 – Partial contribution, attended few meetings
1 – No contribution, attended few meetings
0 – No contribution, did not attended any meetings

## Section 1: Brief Summary & ML Problem Formulation

| Data Set Name | Heart Disease Dataset From UCI |
|---|---|
| Data Size | 18 KB |
| Date of Release | September 11, 2018 |
| No. Of Attributes | 13 |
| No. Of Data Records | 457 |
| Data Source Provider | https://www.kaggle.com/sonumj/heartdisease-dataset-from-uc |
| Data Privacy | Publicly available |

### 1.1 Review last experiment

The heart disease dataset is extracted from the UCI, and the data were collected from Hungary, Switzerland, and the VA Long Beach (The United States). Our main observations focus on the importance of gender and age, and given this two-variable, we made two hypotheses. First, is that gender and age are statistically significant effect on heart disease. Second is that poor eating habits lead to increased cholesterol, high blood sugar and high sugar and high blood pressure can result in diagnosed heart disease. After conducted our main observations, we concluded that the first hypothesis is correct since it is inevitable that age and gender are statistically significant effect on heart disease. For the second hypothesis, we cannot find evidence from the dataset that proves poor eating habits lead to increased cholesterol, high blood sugar and high sugar and high blood pressure can result in diagnosed heart disease because there are no data about eating habits in the dataset.

### 1.2 Current tasks

In this experiment, we plan to apply Machine Learning methods on our dataset to get a better understanding, and further prove our first hypothesis is correct, or even a different result from previous. For Machine learning project, we wish to build a model to process the patient's dataset and predict the chance of occurrence of heart disease. Then the model is investigated using ML explain ability tools and techniques.
Analysis different algorithms, then we will decide the best model/algorithm that we are going to use to perform machine learning.

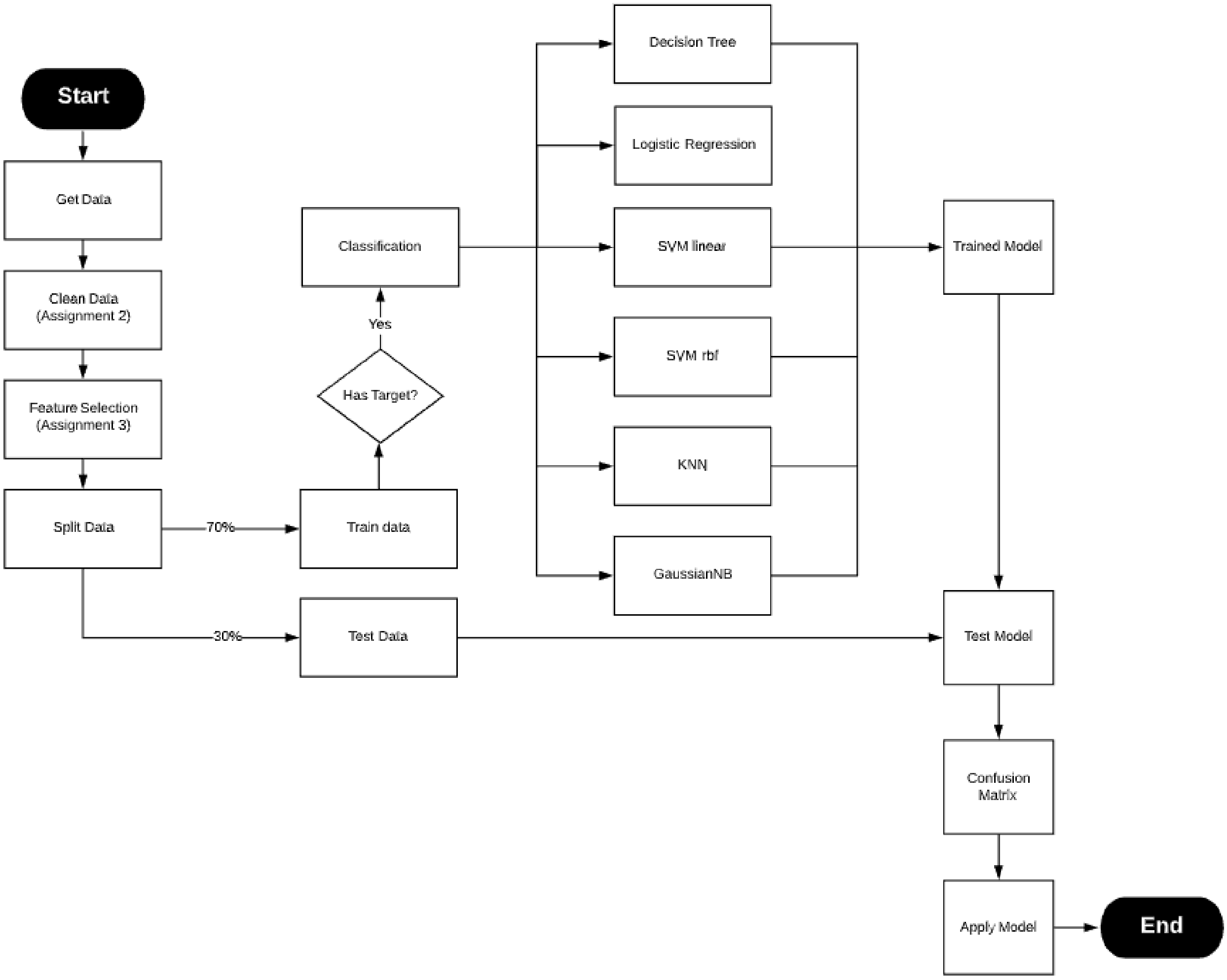
### 1.3 Model for our dataset

After analysing the pros and cons of various models, our team decided to use classification as our model. We will apply different type of classifiers to the data set and analysis the score, then pick the most accurate one as our classifier. According to Brownlee (2020), Classification refers to predictive modelling problems, where category labels are predicted for a given input data example. Since the goal of this project is to use machine learning to predict whether a patient has heart disease, our team believes that using the classification model is an ideal solution.

Classifiers we will apply and reasons we choose them:

- Decision Tree - The decision tree needs less process for data preparation during pre-processing compared to other classifiers/algorithms. Another great benefit from the decision tree is that missing values in the data will not influence the process of conducting a decision tree to any considerable extent.
- Logistic Regression – this classifier performs well when the dataset is linearly separable, which is not suit for our dataset. However, this classifier can measure logistic regression and its direction of association (negative or positive)
- SVM – our group is not familiar with the heart disease data, and SVM is perform very well when we have no idea on the data, it is even work well with unstructured and semi structured
- K Nearest Neighbor – The classifier is called “lazy learner”, which means it does not learn anything in the training period; it stores the training dataset. It learns from it only at the tie of making real-time predictions, which form the K Nearest Neighbor classifier much faster than other classifiers that require training. Another advantage is that it is very easy to implement.
- GaussianNB – it is a simple classifier. A Naïve Bayes classifier will converge quicker than discriminative classifiers like logistic regression, which need less training data. When we have experimented these classifiers, we apply the classifier in a simulation that we created to predict the chance for a human being can get heart disease.

### 1.4 Machine Learning Flowchart

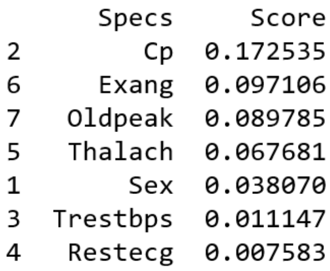


The above Flow chart shows the overall process of this Machine Learning project.

## Section 2: Results and Discussion (max 7 pages)

### 2.1 Feature selection

In this experiment, we will be continuing using data we have cleaned from assignment 1 to perform out machine learning tasks.
To perform a machine learning experiment, we need to find the best features; hence, we drop num (diagnosis) and place columns from the dataset and use num as model validation feature.
Best 7 features:



```
       Specs     Score
2         Cp  0.172535
6      Exang  0.097106
7    Oldpeak  0.089785
5    Thalach  0.067681
1        Sex  0.038070
3   Trestbps  0.011147
4    Restecg  0.007583
```

### 2.2 Data splitting and pipeline

We put these best features into a dataset for our machine learning experiments, and then split the dataset into 30% for test data, and 70% for train data.

Then we put all the algorithms into a classifier with a pipeline to get accuracy for each algorithm

- Decision Tree – 72.66%
- Logistic Regression – 74.22%
- SVM – 67.97%
- SVC – 58.59%
- K Nearest Neighbor – 63.28%
- GaussianNB – 82.03%

### 2.3 Classification

#### 2.3.1 Decision Tree

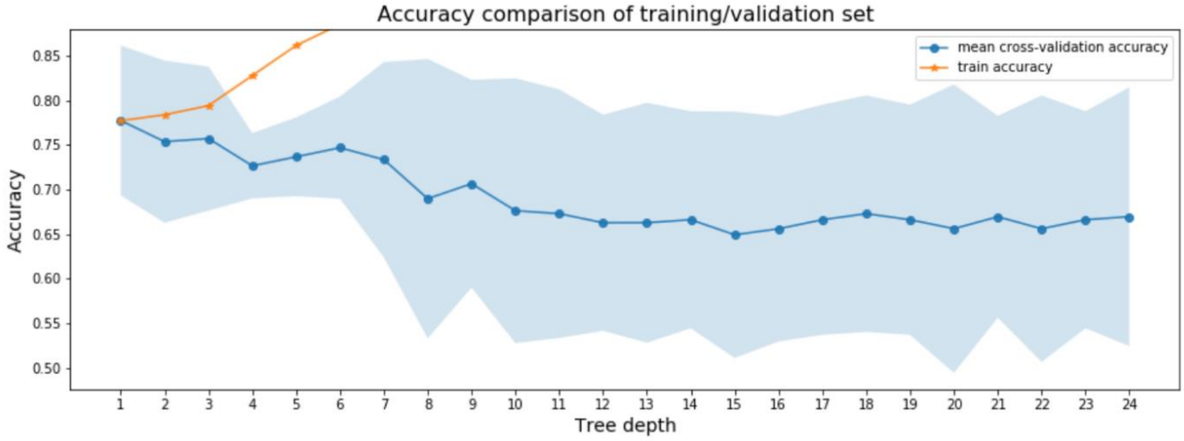
A decision tree uses a tree structure to model a hierarchical path of decisions and it serves as a classifier to predict if a patient might have cancer based on a set of labelled data. The first step is to split the data to train data (70%) and test data (30%). Then we used DecisionTreeClassifer from sklearn library to build the decision tree.
First, we used a classification tree with max depth equals to 5, and we got:

- Training accuracy – 85.81%
- Testing accuracy – 78.13%

Then we perform cross validation to get average accuracy:

- Mean accuracy – 70.29%

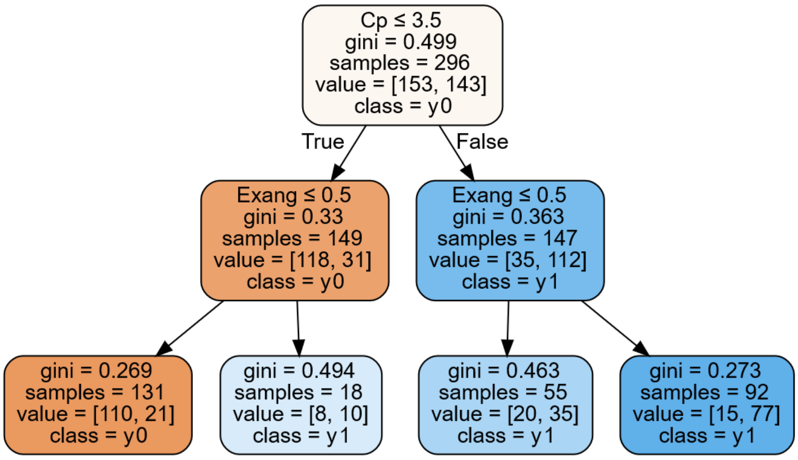
The training accuracy and testing accuracy are higher than the mean accuracy, which tells us the max depth should be lower than 5.
Thus, we conduct a graph that shows accuracy per decision tree depth on training data



Then we use another algorithm to find the best depth from the tree. In this case, the depth 1 tree achieves the best mean cross-validation accuracy 77.72316 +/- 4.20058% on training dataset.

- Training accuracy – 77.70%
- Testing accuracy – 71.09%

Last step for decision tree, we export the decision tree graph:



### 2.3.2 Logistic Regression

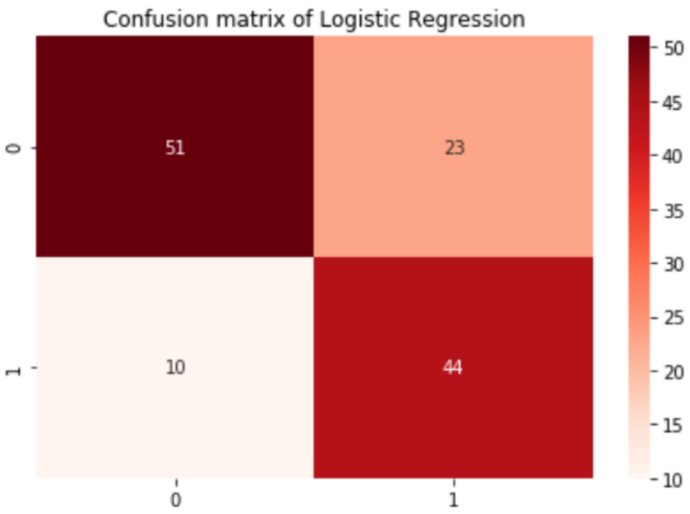
Frist we conduct the accuracy report:

- accuracy score for Logistic Regression – 74.22%
- Precision of predictions – 65.67%
- Recall of predictions – 81.48%
- F1-score of predictions – 72.73%

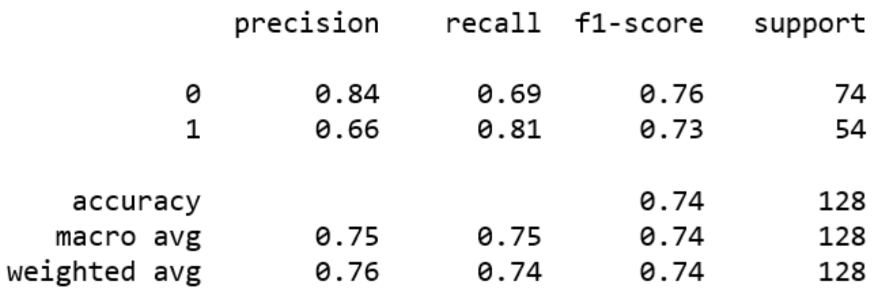
Next, we perform the cross-validation algorithm to get mean accuracy:

- Mean accuracy – 75.38%

Then we create a confusion matrix to summarize of prediction results:


Confusion matrix of Logistic Regression

Last, we conduct a classification report:


```
              precision    recall  f1-score   support

           0       0.84      0.69      0.76        74
           1       0.66      0.81      0.73        54

    accuracy                           0.74       128
   macro avg       0.75      0.75      0.74       128
weighted avg       0.76      0.74      0.74       128
```

- Recall – 69% for positive and 81% for negative; positive has a moderate recall, which means many classes are not correctly recognized, and negative has a high recall, which means many classes is correctly recognized.
- Precision – 84% for positive and 66% for negative; positive has a high precision, which means the classifier is returning accurate result, and negative has a moderate precision, which means many results are not correct.

Positive has high precision and moderate recall, which means we miss a lot of positive examples but those we predict as positive are indeed positive.

Negative has moderate precision and high recall, which means most of the positive examples are correctly recognized but there are a lot of false positives.

### 2.3.3 SVM (kernel = 'linear', C=1)

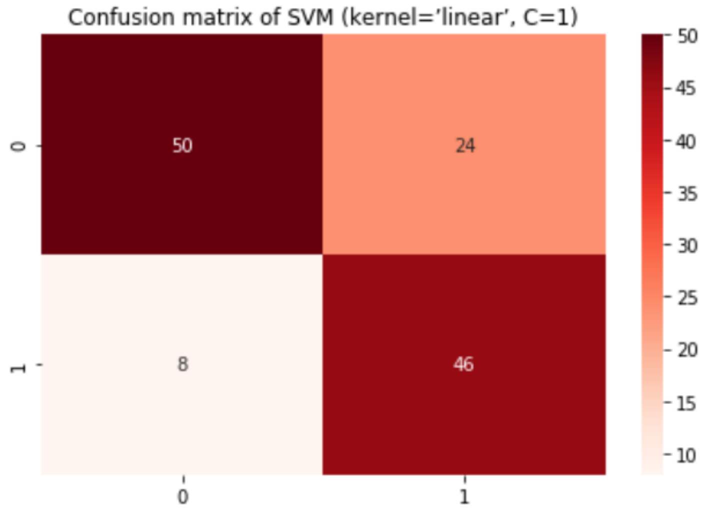
Frist we conduct the accuracy report:

- accuracy score for SVM – 75.00%
- Precision of predictions – 65.71%
- Recall of predictions – 85.18%
- F1-score of predictions – 74.19%

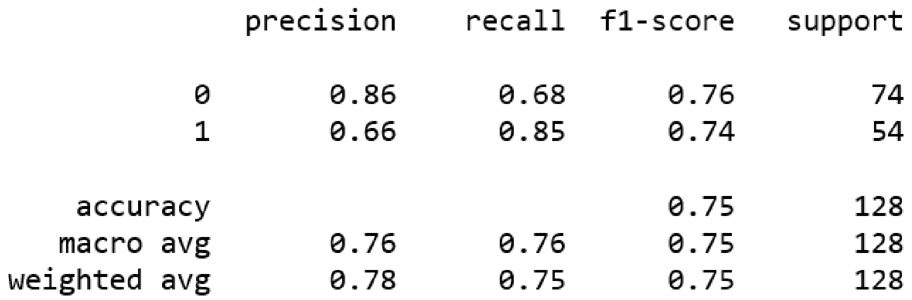
Next, we perform the cross-validation algorithm to get mean accuracy:

- Mean accuracy – 75.38%

Then we create a confusion matrix to summarize of prediction results:



Last, we conduct a classification report:



```
              precision    recall  f1-score   support

           0       0.86      0.68      0.76        74
           1       0.66      0.85      0.74        54

    accuracy                           0.75       128
   macro avg       0.76      0.76      0.75       128
weighted avg       0.78      0.75      0.75       128
```

- Recall – 68% for positive and 85% for negative; positive has a moderate recall, which means many classes are not correctly recognized, and negative has a high recall, which means many classes is correctly recognized.
- Precision – 86% for positive and 68% for negative; positive has a high precision, which means the classifier is returning accurate result, and negative has a moderate precision, which means many results are not correct.

Positive has high precision and moderate recall, which means we miss a lot of positive examples but those we predict as positive are indeed positive.

Negative has moderate precision and high recall, which means most of the positive examples are correctly recognized but there are a lot of false positives.

### 2.3.4 SVC (kernel='rbf', C=1, gamma = 'auto')

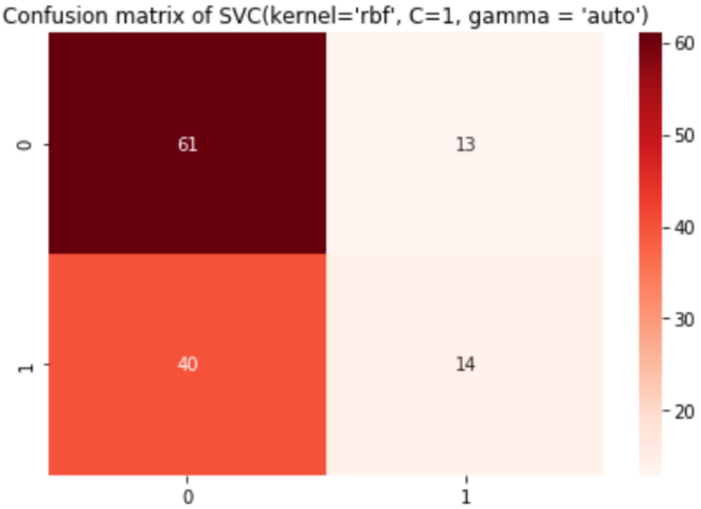
Frist we conduct the accuracy report:

- accuracy score for SVC – 58.59%
- Precision of predictions – 51.85%
- Recall of predictions – 25.93%
- F1-score of predictions – 34.57%

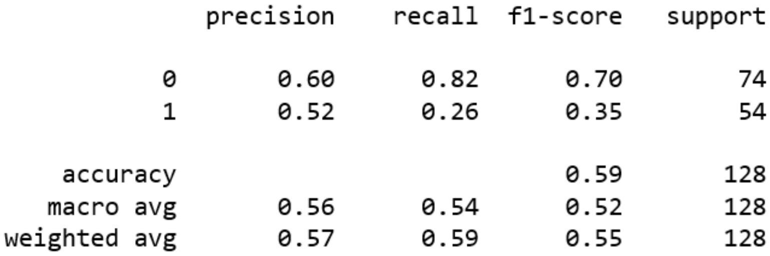
Next, we perform the cross-validation algorithm to get mean accuracy:

- Mean accuracy – 56.82%

Then we create a confusion matrix to summarize of prediction results:



Last, we conduct a classification report:



```
              precision    recall  f1-score   support

           0       0.60      0.82      0.70        74
           1       0.52      0.26      0.35        54

    accuracy                           0.59       128
   macro avg       0.56      0.54      0.52       128
weighted avg       0.57      0.59      0.55       128
```

- Recall – 82% for positive and 26% for negative; positive has a strong recall, which many classes is correctly recognized, and negative has a low recall, which means a lot of classes are not correctly recognized.
- Precision – 60% for positive and 52% for negative; positive has a moderate precision, which means many results are not correct, and negative has a low precision, which means many results are not correct.

Positive has moderate precision and high recall, which means most of the positive examples are correctly recognized but there are a lot of false positives.

Negative has moderate precision and low recall, which means there are a lot of false positives and missed a lot of positive examples

### 2.3.5 K Nearest Neighbor

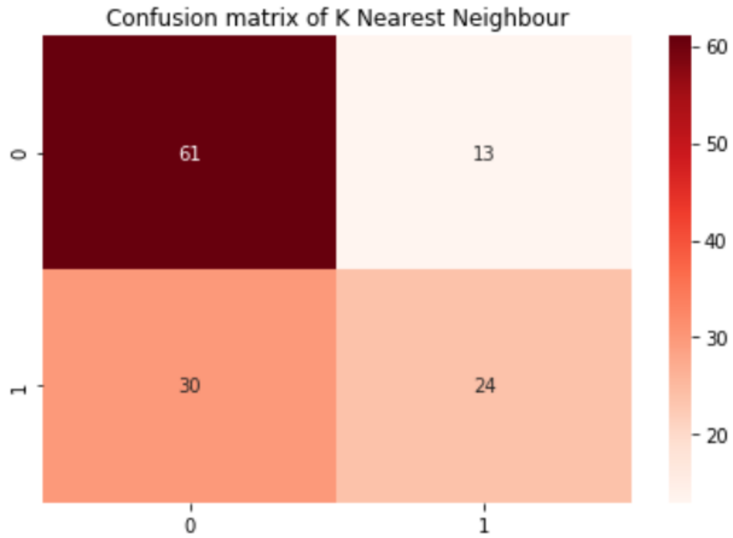
Frist we conduct the accuracy report:

- accuracy score for K Nearest Neighbor – 66.41%
- Precision of predictions – 64.86%
- Recall of predictions – 44.44%
- F1-score of predictions – 52.75%

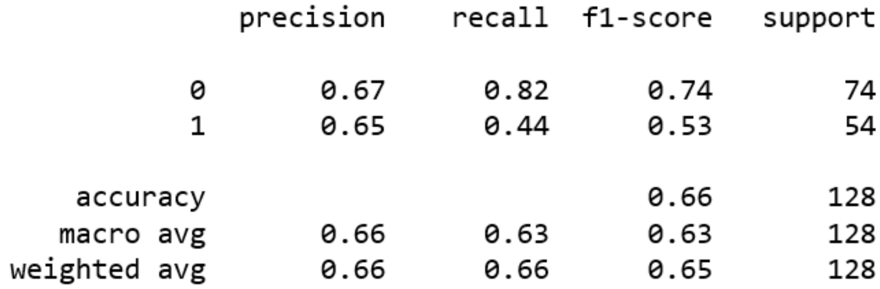
Next, we perform the cross-validation algorithm to get mean accuracy:

- Mean accuracy – 56.80%

Then we create a confusion matrix to summarize of prediction results:



Last, we conduct a classification report:



```
              precision    recall  f1-score   support

           0       0.67      0.82      0.74        74
           1       0.65      0.44      0.53        54

    accuracy                           0.66       128
   macro avg       0.66      0.63      0.63       128
weighted avg       0.66      0.66      0.65       128
```

- Recall – 82% for positive and 44% for negative; positive has a strong recall, which means many classes are correctly recognized, and negative has a low recall, which means many classes are not correctly recognized.
- Precision – 67% for positive and 65% for negative, both have a moderate precision, which means many results they are returning are not correct.

Positive has moderate precision and high recall, which means most of the positive examples are correctly recognized but there are a lot of false positives.
Negative has moderate precision and low recall, which means there are a lot of false positives and missed a lot of positive examples

### 2.3.6 GaussianNB

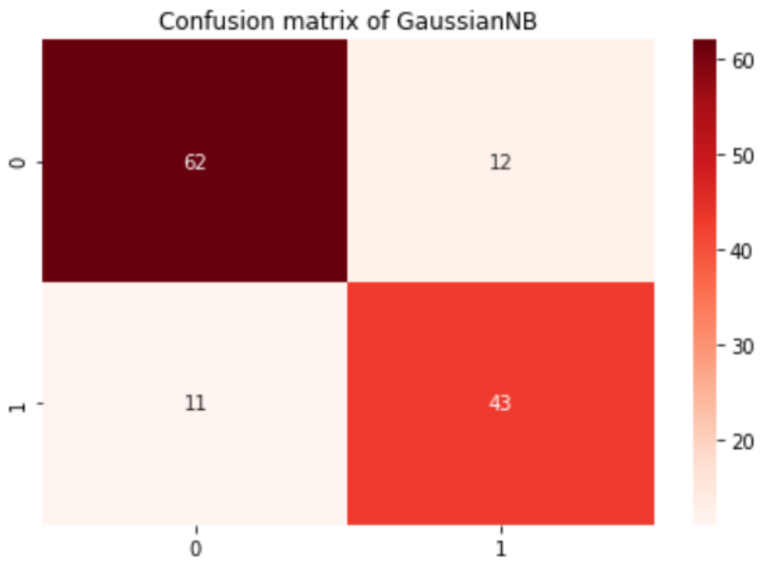
Frist we conduct the accuracy report:

- accuracy score for GaussianNB – 82.03%
- Precision of predictions – 78.18%
- Recall of predictions – 79.63%
- F1-score of predictions – 78.80%

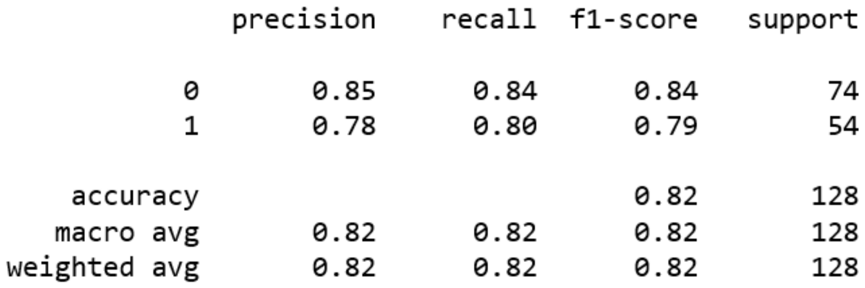
Next, we perform the cross-validation algorithm to get mean accuracy:

- Mean accuracy – 76.41%

Then we create a confusion matrix to summarize of prediction results:



Last, we conduct a classification report:



```
              precision    recall  f1-score   support

           0       0.85      0.84      0.84        74
           1       0.78      0.80      0.79        54

    accuracy                           0.82       128
   macro avg       0.82      0.82      0.82       128
weighted avg       0.82      0.82      0.82       128
```

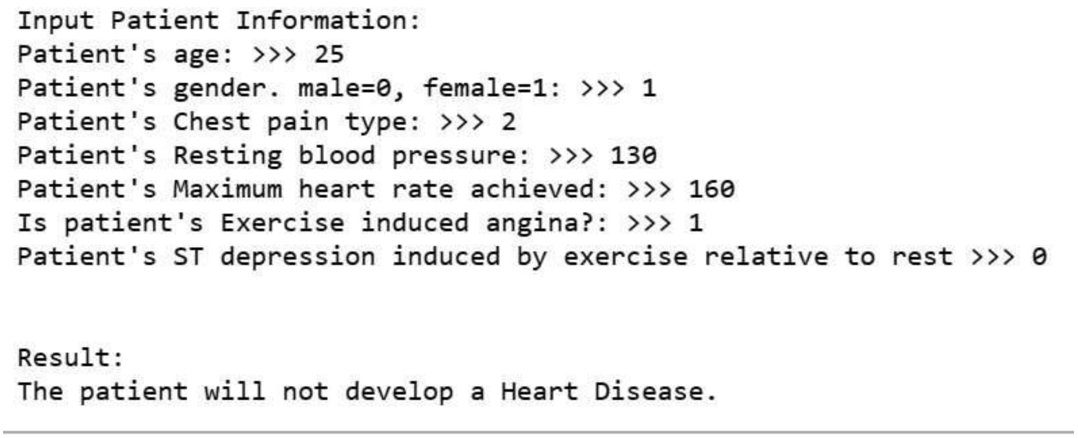
- Recall – 84% for positive and 80% for negative; both has a strong recall, which means many classes are correctly recognized.
- Precision – 85% for positive and 78% for negative, both has a strong precision, which means most results they are returning are correct.

Positive has strong precision and strong recall, which means most of the positive examples are correctly recognized and those we predict as positive are indeed positive.

Negative has strong precision and strong recall, which means most of the positive examples are correctly recognized and those we predict as positive are indeed positive.

### 2.4 Applying model in simulation

Finally, we have experimented all models, and now we are going to apply the model in a simulation that we created to predict the chance for a human being can get heart disease.

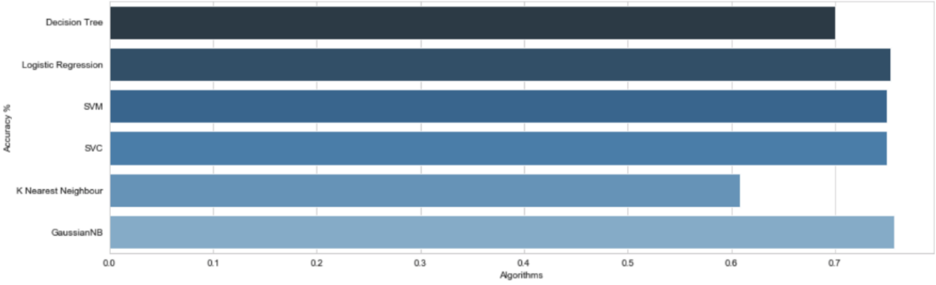


```
Input Patient Information:
Patient's age: >>> 25
Patient's gender. male=0, female=1: >>> 1
Patient's Chest pain type: >>> 2
Patient's Resting blood pressure: >>> 130
Patient's Maximum heart rate achieved: >>> 160
Is patient's Exercise induced angina?: >>> 1
Patient's ST depression induced by exercise relative to rest >>> 0


Result:
The patient will not develop a Heart Disease.
```

## Section 3: Conclusions

By building the model and running different machine learning classifiers on our dataset, it helps us to predict the chances of patients having heart disease based on selected features that have the most predictive value. We have trained and tested our dataset using Classification model with six different classifiers. Among them, the GaussianNB classifier gives the highest mean accuracy at 82.03%, while the K Nearest Neighbor classifier has the lowest mean accuracy at around 60%. After comparing the cons and pros of each machine learning classifier, we could select the one that best fits in our purpose of predicting heart disease. It is helpful for the prediction of heart disease in the future.



We have some suggestions to improve the performance of our machine learning model to achieve better results for predicting the chances of heart disease for a patient. Firstly, we can increase our training data size, as we only have 457 rows of patients' data for both training and testing. To upscale our dataset, we can search for more open-source data, compare and add them to our existing dataset. If given more patients' data for training, it is highly possible that we can achieve better performance inaccuracy in some of the training classifiers that require larger dataset. To further elaborate on this, we could create a learning curve to see how different the size of the training dataset can affect the performance of each machine learning classifier. Once we have created the learning curve graph, we could then use the best size of the data for a different classifier, and in this way, we could further increase our accuracy in predicting heart disease.

Secondly, we could limit the range of the selected features. Our current model is using seven best features; however, some of the features have low predictive value compared to others. By testing and comparing the different number of the best features in our machine learning model, we could then select the classifier with the number of best features that yields the highest accuracy.

Thirdly, changing the standard configuration for the algorithms used in our machine learning classifier. We are currently using the built-in standard configuration for each algorithm from the library. If we tune the parameters of the input of the algorithm, we might have a chance to improve the performance of our data classifier.

In summary, using a machine learning model to predict the chances of heart disease is very promising. Even with our generally small amount of data, we could achieve mean accuracy around 80%. By implementing more changes to our data and algorithm, we might achieve even better performance. It will be very beneficial if this kind of machine learning model could be used for patients in the future and help patients to prevent and detect heart disease earlier.

## Section 4: References

Brownlee, J., 2020. 4 Types Of Classification Tasks In Machine Learning. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/types-of-classification-in-machine-learning/> [Accessed 27 May 2020].

The Heart Foundation. 2017. Key Statistics: Heart Disease in Australia. [online] Available at: <https://www.heartfoundation.org.au/About-us/Australia-Heart-Disease-Statistics> [Accessed 15 April 2020].

WHO. 2002. Cardiovascular Death And Disability Can Be Reduced More Than 50 Percent. [online] Available at: <https://www.who.int/mediacentre/news/releases/pr83/en/> [Accessed 27 May 2020].

Microsoft. 2016. Training and Testing Data Sets. [online] Available at: <https://docs.microsoft.com/en-us/analysis-services/data-mining/training-and-testing-data-sets?redirectedfrom=MSDN&view=asallproducts-allversions&viewFallbackFrom=sql-server-ver15> [Accessed 27 May 2020].

Browiee,J.,2017. How much Training Data is Required for Machine Learning?[online] Available at:<https://machinelearningmastery.com/much-training-data-required-machine-learning/> [Accessed 25 May 2020].