Heart Disease Prediction using Machine Learning Algorithms

Yunchao Yang

September 27, 2021

1 Introduction

For this assignment, I am going to study the prediction of cardiovascular disease. Facing high risk of heart disease is a shadow in my life since I was diagnosed with brugada syndrome, I am especially interested in studying the pattern of heart disease.

Heart disease (cardiovascular disease) occurs when blood flow to the heart muscle is suddenly blocked. According to statistics from the World Health Organization, 17.9 million people die from heart attacks every year. Medical research shows that the human lifestyle is the main cause of this heart problem. In addition, there are many key factors that warn that the person may/may not have a heart attack.

The data set contains medical information about some patients, which indicates whether the person has a low or high chance of having a heart attack. Use this information to explore the data set and use different machine learning models to classify the target variable and find an algorithm that fits the data set.

The growth of medical data collection provides new opportunities for doctors to improve patient diagnosis. In recent years, practitioners have increased their use of computer technology to improve decision support. In the healthcare industry, machine learning is becoming an important solution for assisting patient diagnosis. Machine learning is an analysis tool used when tasks are large and difficult to program, such as converting medical records into knowledge, epidemic prediction, and genomic data analysis

1.1 Goal:

The goal of this study is to predict whether a patient should be diagnosed with Heart Disease. This is a binary outcome. • Positive (+) = 1, patient diagnosed with Heart Disease • Negative (-) = 0, patient not diagnosed with Heart Disease

Experiment with various Classification Models & see which yields greatest accuracy.

1.2 Datasets

The dataset is from the UCI database, which collects from 4 different clinical hospitals. The reason why choosing this dataset is because of its reliability, based on the credential of the four clinics. The original datasets contains 76 features. Researchers shrunk the feature space to 13 features, which will be used in the following discussion.

1.2.1 Features

1. age: age in years
2. sex: sex (1 = male; 0 = female)
3. cp: chest pain type • Value 1: typical angina • Value 2: atypical angina • Value 3: non-anginal pain • Value 4: asymptomatic
4. restbp: resting blood pressure (in mm Hg on admission to the hospital)
5. chol: serum cholestoral in mg/dl
6. fbs: (fasting blood sugar > 120 mg/dl) (1 = true; 0 = false)
7. restecg: resting electrocardiographic results • Value 0: normal • Value 1: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV) • Value 2: showing probable or definite left ventricular hypertrophy by Estes' criteria
8. maxhr: maximum heart rate achieved
9. exang: exercise induced angina (1 = yes; 0 = no)
10. oldpeak = ST depression induced by exercise relative to rest
11. slope: the slope of the peak exercise ST segment • Value 1: upsloping • Value 2: flat • Value 3: downsloping
12. mv: number of major vessels (0-3) colored by flourosopy
13. thal: thallium heart scan • Value 3: normal • Value 6: fixed defect • Value 7: reversable defect
14. diagnosis: severerity of heart disease (angiographic disease status)
    - Value 0: Absent
    - Value 1: Level 1
    - Value 2: Level 2
    - Value 3: Level 3
    - Value 4: Level 4

1.3 Implementated Approaches

The heart disease prediction is been carried out using Naive Bayes Decision Tree Classifier, k-Nearest Neigbors (kNN), Gradient Boosting Classifier(GBP) , Support Vector Machines (SVM) and Artificial Neural Networks (ANN).

1) Import Packages
2) Exploratory Data Analysis
3) Data split
4) Machine Learning models evaluation
5) Ensembling
6) Conclusion

The dataset used in the research was the "Heart Disease Dataset" of the UCI Machine Learning Repository

1.4 Libraries Used / Requirements

- numpy == 1.16.5
- pandas == 1.0.3
- matplotlib == 3.4.3
- Seaborn == 0.11.2
- Sklearn == 0.22

2 Exploratory Data Analysis (EDA)

First, we analyze the target variable, which is the goal of this machine learning projects. The output is a binary variable based on the diagnose results, it will be set to 1 if the patient is diagnosed to have heart diseases, regardless of levels, otherwise the output is set to be 0. The distribution of the output is plotted in Figure 7.

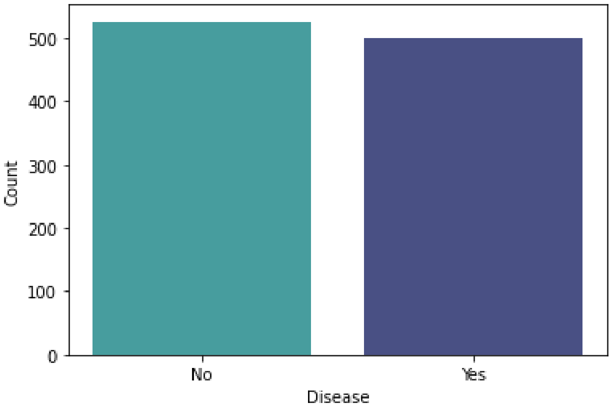


Figure 1: Counts of target output

Total 1025 samples are divided into two catogories: training and testing samples.

```
output from pandas dataframe
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       1025 non-null   int64
 1   sex       1025 non-null   int64
 2   cp        1025 non-null   int64
 3   trestbps  1025 non-null   int64
 4   chol      1025 non-null   int64
 5   fbs       1025 non-null   int64
 6   restecg   1025 non-null   int64
 7   thalach   1025 non-null   int64
 8   exang     1025 non-null   int64
 9   oldpeak   1025 non-null   float64
 10  slope     1025 non-null   int64
 11  ca        1025 non-null   int64
 12  thal      1025 non-null   int64
 13  target    1025 non-null   int64
dtypes: float64(1), int64(13)
memory usage: 112.2 KB
```

By further exploring the data, we exported the descriptions of the data. It is seen that the mean value of variables vary significantly from 0.3 to 246. This indicates that a scaling is required to account for the importance of the numerical values.

```
Out[]:                age          sex           cp     trestbps        chol  \
        count  1025.000000  1025.000000  1025.000000  1025.000000  1025.00000
        mean     54.434146     0.695610     0.942439   131.611707   246.00000
        std       9.072290     0.460373     1.029641    17.516718    51.59251
        min      29.000000     0.000000     0.000000    94.000000   126.00000
        25%      48.000000     0.000000     0.000000   120.000000   211.00000
        50%      56.000000     1.000000     1.000000   130.000000   240.00000
        75%      61.000000     1.000000     2.000000   140.000000   275.00000
        max      77.000000     1.000000     3.000000   200.000000   564.00000

                       fbs      restecg      thalach        exang      oldpeak  \
        count  1025.000000  1025.000000  1025.000000  1025.000000  1025.000000
        mean      0.149268     0.529756   149.114146     0.336585     1.071512
        std       0.356527     0.527878    23.005724     0.472772     1.175053
        min       0.000000     0.000000    71.000000     0.000000     0.000000
        25%       0.000000     0.000000   132.000000     0.000000     0.000000
        50%       0.000000     1.000000   152.000000     0.000000     0.800000
        75%       0.000000     1.000000   166.000000     1.000000     1.800000
        max       1.000000     2.000000   202.000000     1.000000     6.200000

                     slope           ca         thal       target
        count  1025.000000  1025.000000  1025.000000  1025.000000
        mean      1.385366     0.754146     2.323902     0.513171
        std       0.617755     1.030798     0.620660     0.500070
        min       0.000000     0.000000     0.000000     0.000000
        25%       1.000000     0.000000     2.000000     0.000000
        50%       1.000000     0.000000     2.000000     1.000000
        75%       2.000000     1.000000     3.000000     1.000000
        max       2.000000     4.000000     3.000000     1.000000
```

The review the first 5 rows of data is displayed below.

```
Out[]:    age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  slope  \
       0   52    1   0       125   212    0        1      168      0      1.0      2
       1   53    1   0       140   203    1        0      155      1      3.1      0
       2   70    1   0       145   174    0        1      125      1      2.6      0
       3   61    1   0       148   203    0        1      161      0      0.0      2
       4   62    0   0       138   294    1        1      106      0      1.9      1

          ca  thal  target
       0   2     3       0
       1   0     3       0
       2   0     3       0
       3   1     3       0
       4   3     2       0
```

Comparing positive and negative patients, we can see there are vast differences in means for many of our 13 Features. From examining the details, we can observe that positive patients experience heightened maximum heart rate achieved (thalach) average. In addition, positive patients exhibit about 1/3rd the amount of ST depression induced by exercise relative to rest (oldpeak).

2.1 Analyses of selected features

The distribution of patient diagnosed with or without heart disease among male and female are displayed below. It can be seen that the imbalance in this dataset, there are more male samples than females, given that male and female in the general population are equal. Thus this could be an significant defect of this datasets.

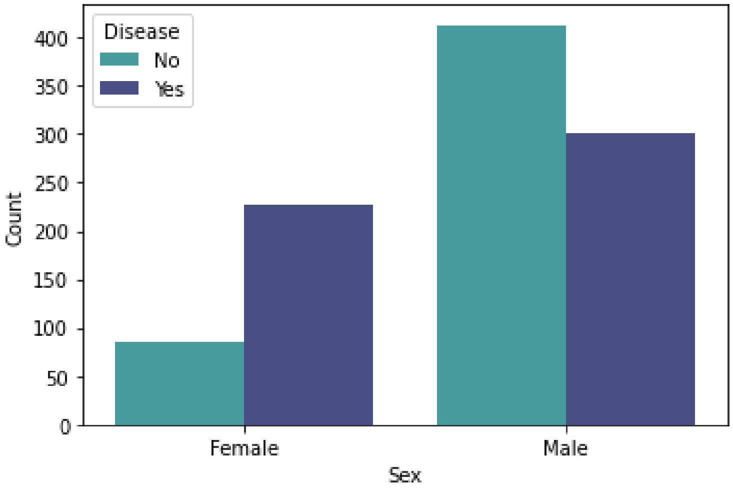


Figure 2: Counts of target output

The patient distribution among different chese pain types feature is plotted below. It can be seen a strong relationship between the chese pain and heart disease.

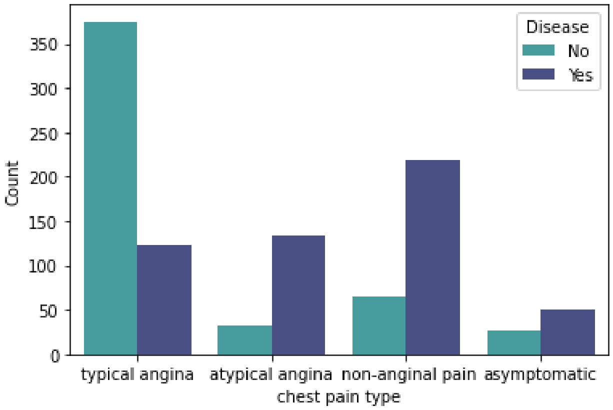


Figure 3: Distribution of samples for chese pain type

2.2 Correlation matrix

The correlation matrix among the pair of each other features are plotted below. It is seen that the positive relations are observed in most of the paird variables.

2.3 Correlation to Target

We are more interested in how each features impact the target variable. We plot the relationship between the target and other variables in the plot and table 1. We can see there is a positive correlation between chest pain (cp) & target (our predictor).

This makes sense since, the greater amount of chest pain results in a greater chance of having heart disease. Cp (chest pain), is a ordinal feature with 4 values: Value 1: typical angina ,Value 2: atypical angina, Value 3: non-anginal pain , Value 4: asymptomatic.

In addition, we see a negative correlation between exercise induced angina (exang) & our predictor. This makes sense because when you excercise, your heart requires more blood, but narrowed arteries slow down blood flow. Pairplots are also a great way to immediately see the correlations between all variables. But you will see me make it with only continuous columns from our data, because with so many features, it can be difficult to see each one. So instead I will make a pairplot with only our continuous features.

To prepare dataset for modeling, there are three steps : assign, split, and normalize, are performed.

2.4 split

The dataset is split into traing and testing data.

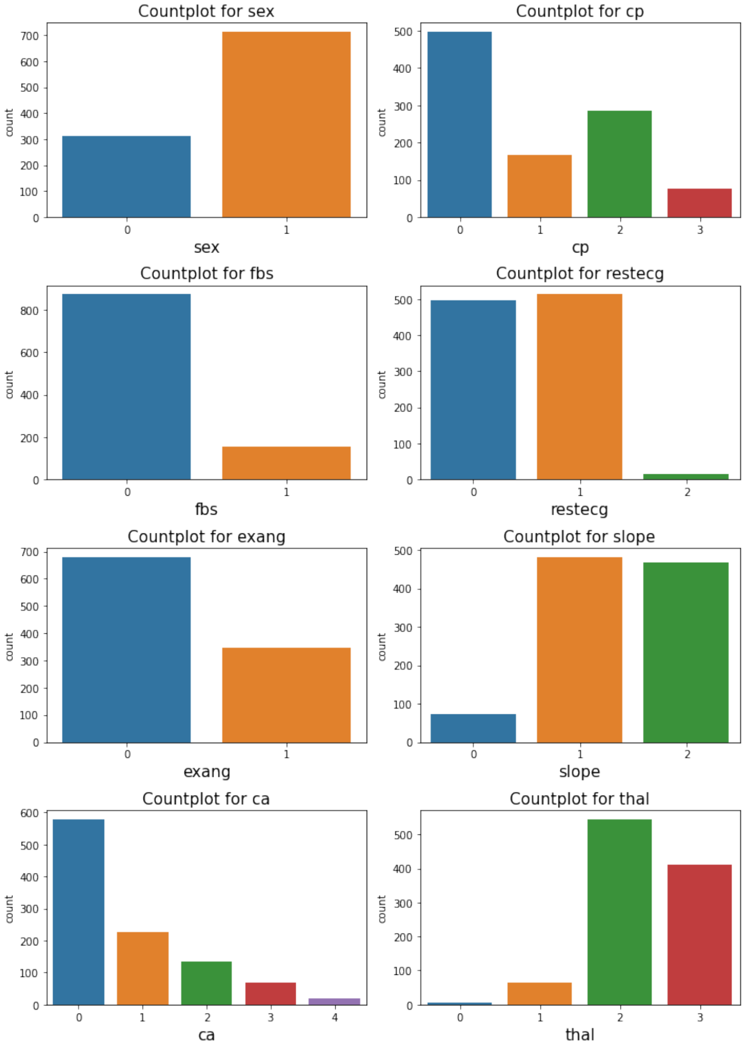


Figure 4: Distribution of samples for different features

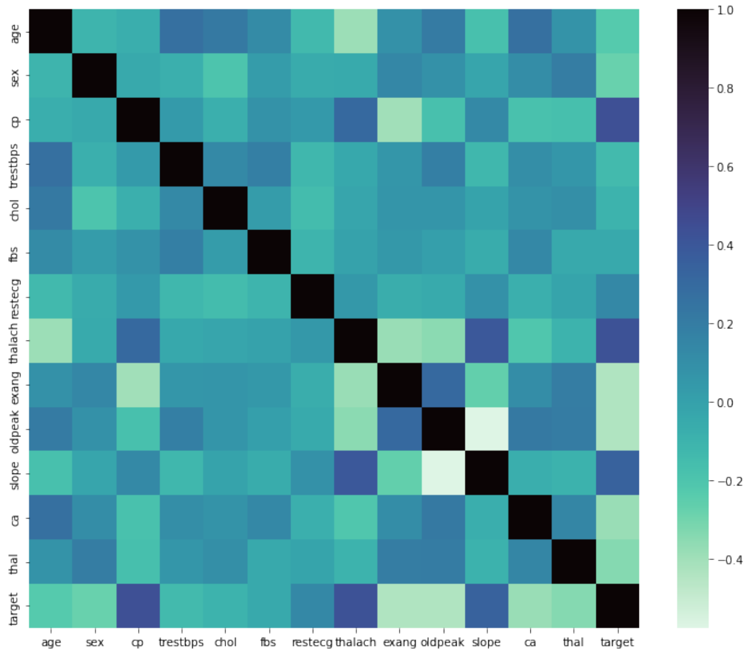


Figure 5: Correlation coefficient matrix different features

Table 1: correlation between target and other variables

feature	correlation coefficient
oldpeak	0.438441
exang	0.438029
cp	0.434854
thalach	0.422895
ca	0.382085
slope	0.345512
thal	0.337838
sex	0.279501
age	0.229324
trestbps	0.138772
restecg	0.134468
chol	0.099966
fbs	0.041164

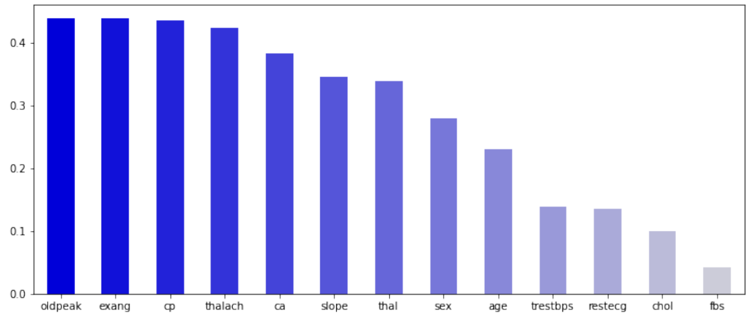


Figure 6: Correlation coefficient matrix different features

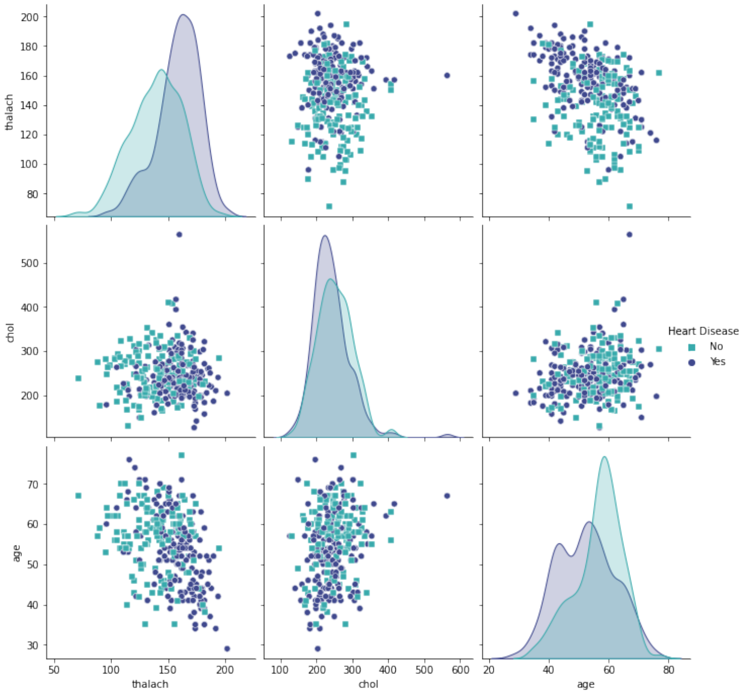


Figure 7: Correlation coefficient matrix different features

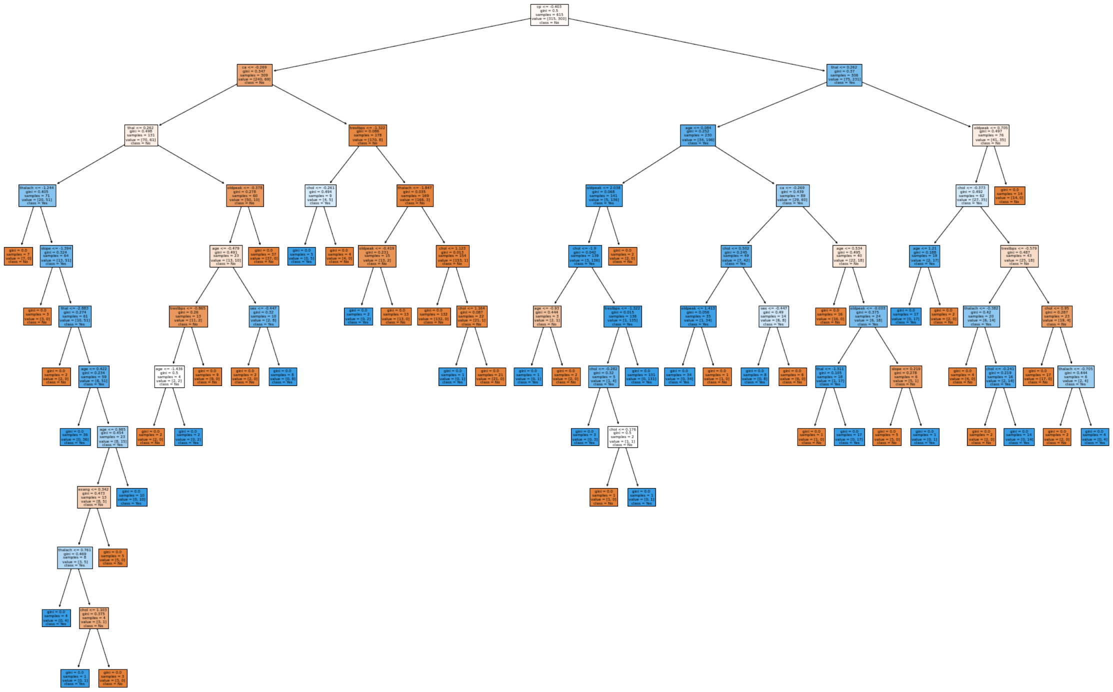

cp <= -0.403 gini = 0.5 samples = 615 value = [315, 300] class = No
ca <= -0.269 gini = 0.347 samples = 309 value = [240, 69] class = No
thal <= 0.262 gini = 0.37 samples = 306 value = [75, 231] class = Yes
thal <= 0.262 gini = 0.498 samples = 131 value = [70, 61] class = No
trestbps <= -1.322 gini = 0.088 samples = 178 value = [170, 8] class = No
age <= 0.084 gini = 0.252 samples = 230 value = [34, 196] class = Yes
oldpeak <= 0.705 gini = 0.497 samples = 76 value = [41, 35] class = No
thalach <= -1.244 gini = 0.405 samples = 71 value = [20, 51] class = Yes
oldpeak <= -0.378 gini = 0.278 samples = 60 value = [50, 10] class = No
chol <= -0.261 gini = 0.494 samples = 9 value = [4, 5] class = Yes
thalach <= -1.847 gini = 0.035 samples = 169 value = [166, 3] class = No
oldpeak <= 2.038 gini = 0.068 samples = 141 value = [5, 136] class = Yes
ca <= -0.269 gini = 0.439 samples = 89 value = [29, 60] class = Yes
chol <= -0.373 gini = 0.492 samples = 62 value = [27, 35] class = Yes
gini = 0.0 samples = 14 value = [14, 0] class = No
gini = 0.0 samples = 7 value = [7, 0] class = No
slope <= -1.394 gini = 0.324 samples = 64 value = [13, 51] class = Yes
age <= -0.479 gini = 0.491 samples = 23 value = [13, 10] class = No
gini = 0.0 samples = 37 value = [37, 0] class = No
gini = 0.0 samples = 5 value = [0, 5] class = Yes
gini = 0.0 samples = 4 value = [4, 0] class = No
oldpeak <= -0.419 gini = 0.231 samples = 15 value = [13, 2] class = No
chol <= 1.123 gini = 0.013 samples = 154 value = [153, 1] class = No
chol <= -1.9 gini = 0.042 samples = 139 value = [3, 136] class = Yes
gini = 0.0 samples = 2 value = [2, 0] class = No
chol <= 0.502 gini = 0.245 samples = 49 value = [7, 42] class = Yes
age <= 0.534 gini = 0.495 samples = 40 value = [22, 18] class = No
age <= 1.21 gini = 0.188 samples = 19 value = [2, 17] class = Yes
trestbps <= -0.579 gini = 0.487 samples = 43 value = [25, 18] class = No
gini = 0.0 samples = 3 value = [3, 0] class = No
thal <= -2.883 gini = 0.274 samples = 61 value = [10, 51] class = Yes
trestbps <= -0.865 gini = 0.26 samples = 13 value = [11, 2] class = No
sex <= -0.447 gini = 0.32 samples = 10 value = [2, 8] class = Yes
gini = 0.0 samples = 2 value = [0, 2] class = Yes
gini = 0.0 samples = 13 value = [13, 0] class = No
gini = 0.0 samples = 132 value = [132, 0] class = No
chol <= 1.164 gini = 0.087 samples = 22 value = [21, 1] class = No
age <= -0.93 gini = 0.444 samples = 3 value = [2, 1] class = No
trestbps <= -1.322 gini = 0.015 samples = 136 value = [1, 135] class = Yes
oldpeak <= 1.413 gini = 0.056 samples = 35 value = [1, 34] class = Yes
sex <= -0.447 gini = 0.49 samples = 14 value = [6, 8] class = Yes
gini = 0.0 samples = 16 value = [16, 0] class = No
oldpeak <= -0.003 gini = 0.375 samples = 24 value = [6, 18] class = Yes
gini = 0.0 samples = 17 value = [0, 17] class = Yes
gini = 0.0 samples = 2 value = [2, 0] class = No
thalach <= -0.382 gini = 0.42 samples = 20 value = [6, 14] class = Yes
chol <= 0.95 gini = 0.287 samples = 23 value = [19, 4] class = No
gini = 0.0 samples = 2 value = [2, 0] class = No
age <= 0.422 gini = 0.234 samples = 59 value = [8, 51] class = Yes
age <= -1.436 gini = 0.5 samples = 4 value = [2, 2] class = No
gini = 0.0 samples = 9 value = [9, 0] class = No
gini = 0.0 samples = 2 value = [2, 0] class = No
gini = 0.0 samples = 8 value = [0, 8] class = Yes
gini = 0.0 samples = 1 value = [0, 1] class = Yes
gini = 0.0 samples = 21 value = [21, 0] class = No
gini = 0.0 samples = 1 value = [0, 1] class = Yes
gini = 0.0 samples = 2 value = [2, 0] class = No
chol <= -0.282 gini = 0.32 samples = 5 value = [1, 4] class = Yes
gini = 0.0 samples = 131 value = [0, 131] class = Yes
gini = 0.0 samples = 34 value = [0, 34] class = Yes
gini = 0.0 samples = 1 value = [1, 0] class = No
gini = 0.0 samples = 8 value = [0, 8] class = Yes
gini = 0.0 samples = 6 value = [6, 0] class = No
thal <= -1.311 gini = 0.105 samples = 18 value = [1, 17] class = Yes
slope <= 0.219 gini = 0.278 samples = 6 value = [5, 1] class = No
gini = 0.0 samples = 4 value = [4, 0] class = No
chol <= -0.241 gini = 0.219 samples = 16 value = [2, 14] class = Yes
gini = 0.0 samples = 17 value = [17, 0] class = No
thalach <= -0.705 gini = 0.444 samples = 6 value = [2, 4] class = Yes
gini = 0.0 samples = 36 value = [0, 36] class = Yes
age <= 0.985 gini = 0.454 samples = 23 value = [8, 15] class = Yes
gini = 0.0 samples = 2 value = [2, 0] class = No
gini = 0.0 samples = 2 value = [0, 2] class = Yes
gini = 0.0 samples = 3 value = [0, 3] class = Yes
chol <= 0.176 gini = 0.5 samples = 2 value = [1, 1] class = No
gini = 0.0 samples = 1 value = [1, 0] class = No
gini = 0.0 samples = 17 value = [0, 17] class = Yes
gini = 0.0 samples = 5 value = [5, 0] class = No
gini = 0.0 samples = 1 value = [0, 1] class = Yes
gini = 0.0 samples = 2 value = [2, 0] class = No
gini = 0.0 samples = 14 value = [0, 14] class = Yes
gini = 0.0 samples = 2 value = [2, 0] class = No
gini = 0.0 samples = 4 value = [0, 4] class = Yes
exang <= 0.342 gini = 0.473 samples = 13 value = [8, 5] class = No
gini = 0.0 samples = 10 value = [0, 10] class = Yes
gini = 0.0 samples = 1 value = [1, 0] class = No
gini = 0.0 samples = 1 value = [0, 1] class = Yes
thalach <= 0.761 gini = 0.469 samples = 8 value = [3, 5] class = Yes
gini = 0.0 samples = 5 value = [5, 0] class = No
gini = 0.0 samples = 4 value = [0, 4] class = Yes
chol <= 1.103 gini = 0.375 samples = 4 value = [3, 1] class = No
gini = 0.0 samples = 1 value = [0, 1] class = Yes
gini = 0.0 samples = 3 value = [3, 0] class = No

Figure 8: The original Decision Tree

3 Decision Tree Classifier

3.0.1 Precision, Recall, F1-score and Support:

Precision : be "how many are correctly classified among that class"

Recall : "how many of this class you find over the whole number of element of this class"

F1-score : harmonic mean of precision and recall values.

F1 score reaches its best value at 1 and worst value at 0.

F1 Score = 2 x ((precision x recall) / (precision + recall))

Support: # of samples of the true response that lie in that class.

4 Plot a 4-layer decision tree

4.1 Pruning

4.1.1 1. prepruning depth of decision tree

Prepruning sets constraint for growth of decision tree on an early stage. By limiting *max_depth* , *min_samples* etc., parameters, it is effective to grid search parameter space and choose the optimal parameters for selected datasets.

We set max_depth, meaning maximum depth of decision tree, for parameter tuning.

```
Train score 0.9983739837398374
Test score 0.9658536585365853
```

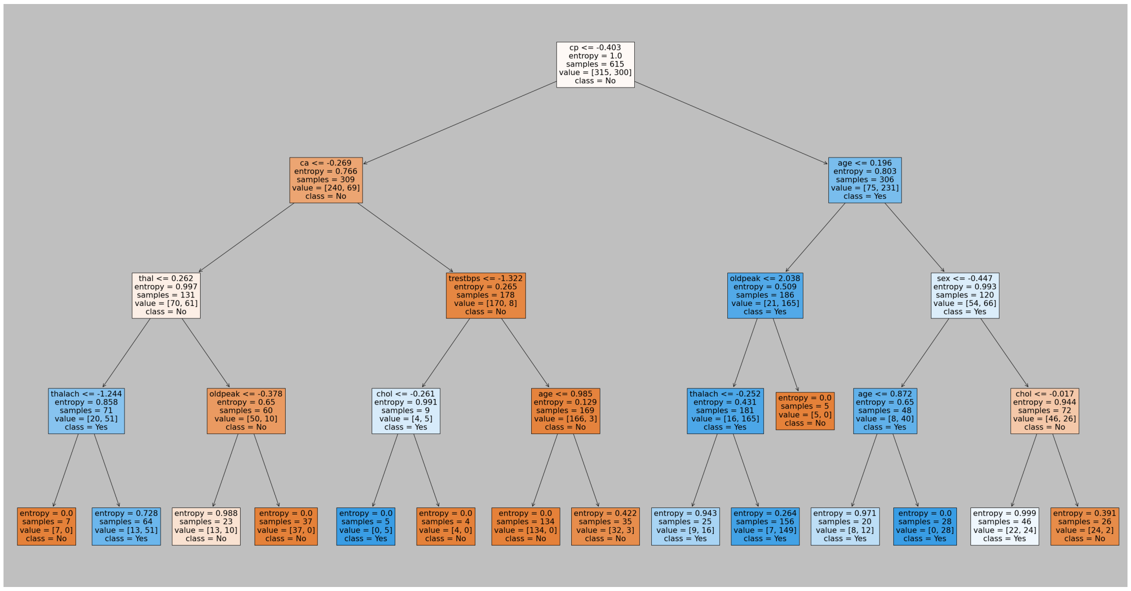


Figure 9: The 4-layer pruned Decision Tree

```
Train Confusion matrix
```

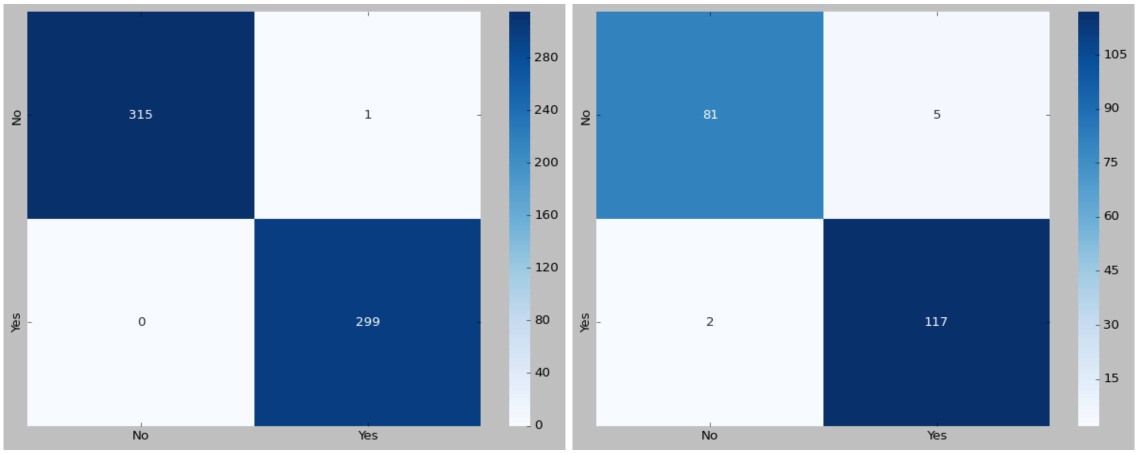


Figure 10: Confusion matrix of training and testing data

4.2 2. Post pruning

Cost complexity pruning is is an effect common postpruning technique to avoid the overfitting of decision trees. In the cose complexity pruning method, we will find the right parameter for **alpha**. We will get the alpha values and check the accuracy with the pruned trees.

We will remove the last element in clfs and ccp_alphas, because it is the trivial tree with only one node.

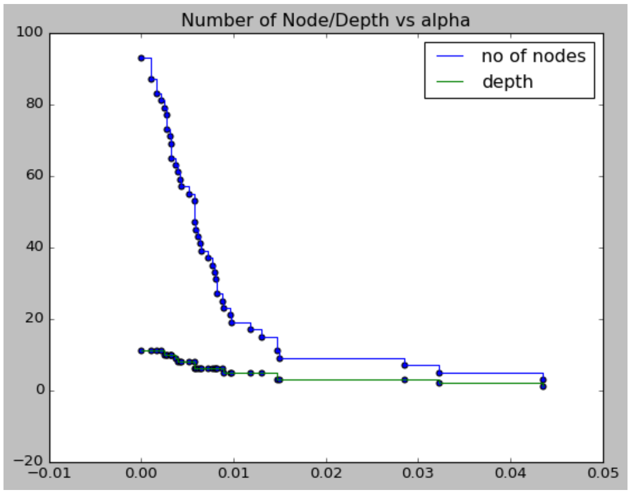


Figure 11: The variation of nodes and depths with alpha

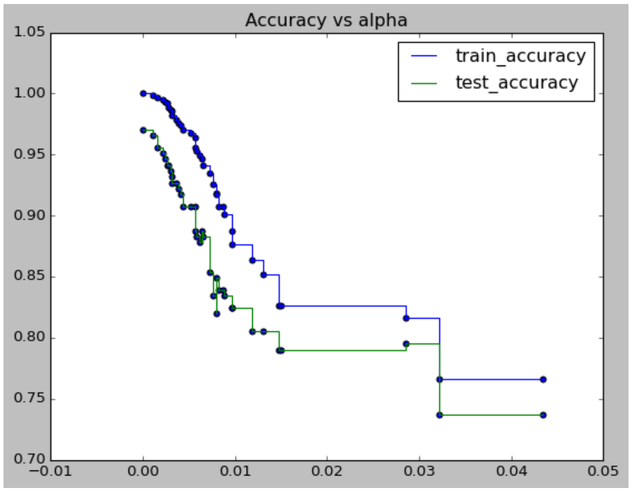


Figure 12: The training and testing accuracy at different alpha

we can see that As alpha increases no of nodes and depth decreases. The best performance is obtained at smaller alpha value = 0.005. We can choose alpha = 0.005.

```
Train score = 0.9707317073170731
Test score = 0.9073170731707317
```

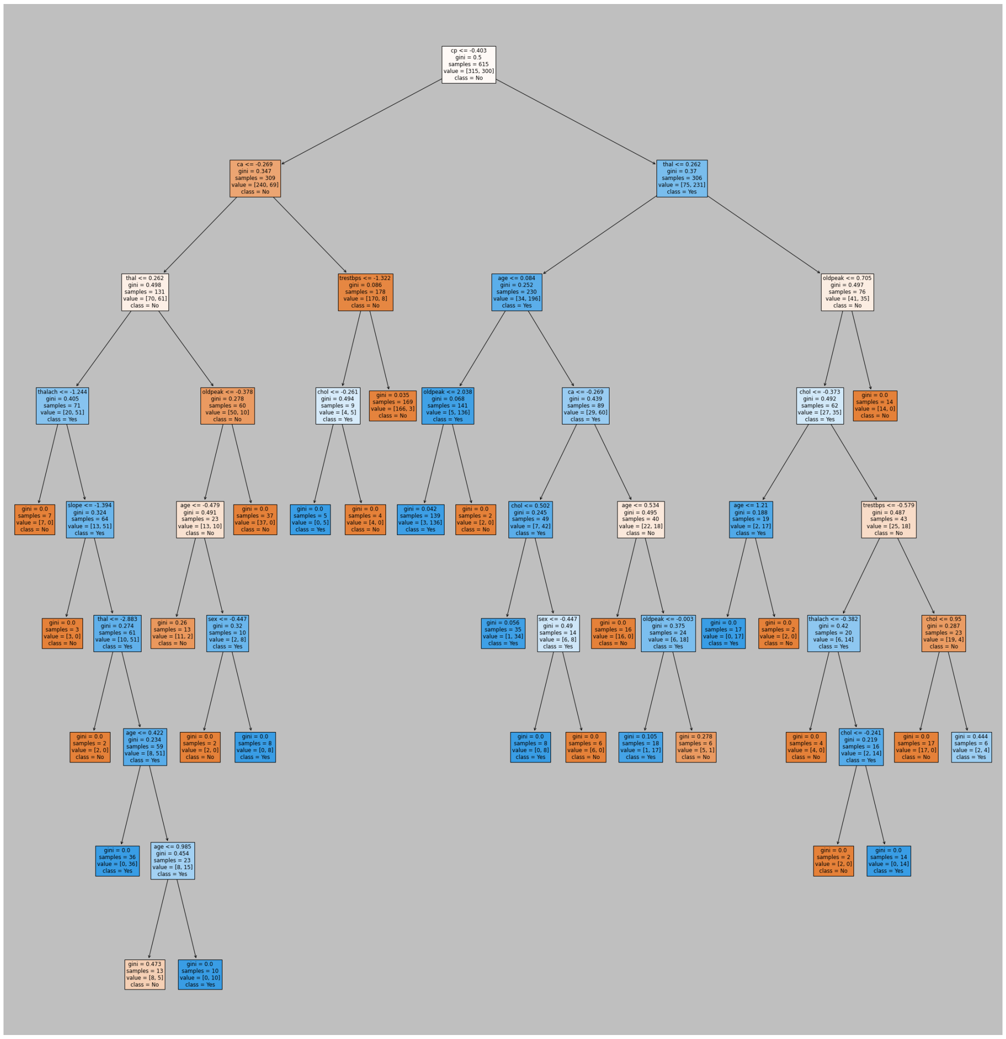

cp <= -0.403 gini = 0.5 samples = 615 value = [315, 300] class = No
ca <= -0.269 gini = 0.347 samples = 309 value = [240, 69] class = No
thal <= 0.262 gini = 0.37 samples = 306 value = [75, 231] class = Yes
thal <= 0.262 gini = 0.498 samples = 131 value = [70, 61] class = No
trestbps <= -1.322 gini = 0.086 samples = 178 value = [170, 8] class = No
age <= 0.084 gini = 0.252 samples = 230 value = [34, 196] class = Yes
oldpeak <= 0.705 gini = 0.497 samples = 76 value = [41, 35] class = No
thalach <= -1.244 gini = 0.405 samples = 71 value = [20, 51] class = Yes
oldpeak <= -0.378 gini = 0.278 samples = 60 value = [50, 10] class = No
chol <= -0.261 gini = 0.494 samples = 9 value = [4, 5] class = Yes
gini = 0.035 samples = 169 value = [166, 3] class = No
oldpeak <= 2.038 gini = 0.068 samples = 141 value = [5, 136] class = Yes
ca <= -0.269 gini = 0.439 samples = 89 value = [29, 60] class = Yes
chol <= -0.373 gini = 0.492 samples = 62 value = [27, 35] class = Yes
gini = 0.0 samples = 14 value = [14, 0] class = No
gini = 0.0 samples = 7 value = [7, 0] class = No
slope <= -1.394 gini = 0.324 samples = 64 value = [13, 51] class = Yes
age <= -0.479 gini = 0.491 samples = 23 value = [13, 10] class = No
gini = 0.0 samples = 37 value = [37, 0] class = No
gini = 0.0 samples = 5 value = [0, 5] class = Yes
gini = 0.0 samples = 4 value = [4, 0] class = No
gini = 0.042 samples = 139 value = [3, 136] class = Yes
gini = 0.0 samples = 2 value = [2, 0] class = No
chol <= 0.502 gini = 0.245 samples = 49 value = [7, 42] class = Yes
age <= 0.534 gini = 0.495 samples = 40 value = [22, 18] class = No
age <= 1.21 gini = 0.188 samples = 19 value = [2, 17] class = Yes
trestbps <= -0.579 gini = 0.487 samples = 43 value = [25, 18] class = No
gini = 0.0 samples = 3 value = [3, 0] class = No
thal <= -2.883 gini = 0.274 samples = 61 value = [10, 51] class = Yes
gini = 0.26 samples = 13 value = [11, 2] class = No
sex <= -0.447 gini = 0.32 samples = 10 value = [2, 8] class = Yes
gini = 0.056 samples = 35 value = [1, 34] class = Yes
sex <= -0.447 gini = 0.49 samples = 14 value = [6, 8] class = Yes
gini = 0.0 samples = 16 value = [16, 0] class = No
oldpeak <= -0.003 gini = 0.375 samples = 24 value = [6, 18] class = Yes
gini = 0.0 samples = 17 value = [0, 17] class = Yes
gini = 0.0 samples = 2 value = [2, 0] class = No
thalach <= -0.382 gini = 0.42 samples = 20 value = [6, 14] class = Yes
chol <= 0.95 gini = 0.287 samples = 23 value = [19, 4] class = No
gini = 0.0 samples = 2 value = [2, 0] class = No
age <= 0.422 gini = 0.234 samples = 59 value = [8, 51] class = Yes
gini = 0.0 samples = 2 value = [2, 0] class = No
gini = 0.0 samples = 8 value = [0, 8] class = Yes
gini = 0.0 samples = 8 value = [0, 8] class = Yes
gini = 0.0 samples = 6 value = [6, 0] class = No
gini = 0.105 samples = 18 value = [1, 17] class = Yes
gini = 0.278 samples = 6 value = [5, 1] class = No
gini = 0.0 samples = 4 value = [4, 0] class = No
chol <= -0.241 gini = 0.219 samples = 16 value = [2, 14] class = Yes
gini = 0.0 samples = 17 value = [17, 0] class = No
gini = 0.444 samples = 6 value = [2, 4] class = Yes
gini = 0.0 samples = 36 value = [0, 36] class = Yes
age <= 0.985 gini = 0.454 samples = 23 value = [8, 15] class = Yes
gini = 0.0 samples = 2 value = [2, 0] class = No
gini = 0.0 samples = 14 value = [0, 14] class = Yes
gini = 0.473 samples = 13 value = [8, 5] class = No
gini = 0.0 samples = 10 value = [0, 10] class = Yes

Figure 13: Decision tree using postpruning technique

We can see that the size of decision tree significantly got reduced than the original decision tree while maintaining the same accuracy. And the decision tree is simplied than the original tree.

4.2.1 Next we explore the relationship between auc and depth

Discussion: For this dataset, there is no clear optimal depth. As the tree is deeper than 6, the AUC is close to 1. But since the tree is easily overfitting, it would need additional pruning.

5 Gradient Boosting Classifier

In addition to the decision tree with pruning, we also investigate the gradient boosting method to search for the optimal hyperparameters,including depth, learning rate.

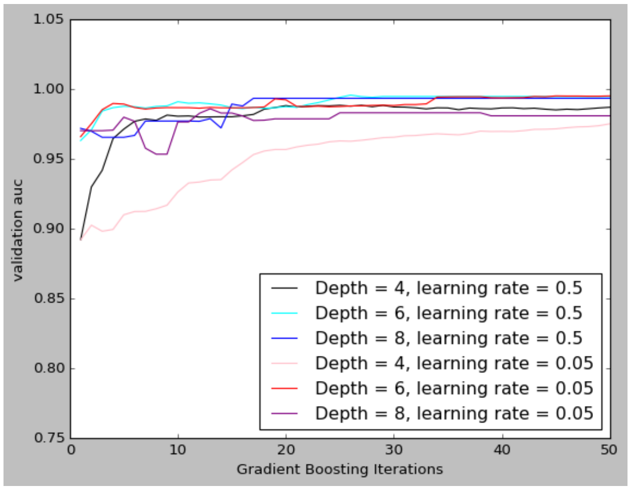


Figure 14: AUC at different depth and learning rate

```
CV Accuracy: mean = 0.997 (std deviation = 0.006)
```

It is clear that with a lower depth = 6, the gradient boosting method achieves a higher accuracy

6 k-Nearest Neighbors

As opposed to decision tree models, KNN is a an instance based learning, instead of fitting a learned function, to classify the train and testing instances.

For this reason, kNN training is fast by cleaning and storing the data, However, testing using kNN is comparatively slower since, it will search the feature space in the traing data to identify the 'nearest' neighbors, thus much more time-consuming. For our heart disease data, we have training data of hundreds of rows, thus k should be chosen accordingly.

kNN is an non-parametric method, only consists of the k closest training examples in the feature space. We shall use the scikit learning library again to make predictions based on KNN. Below we test to measure the training and test errors when the value of k increases.

A very important consideration in using distance metrics is scaling our input. In a tree-based model, the scale of the variable is not important, but for KNN, when measuring the nearest neighbore, the variable should be scaled equally, unless we have a specific reason to weight the variable higher, otherwise it has a large variance The variable will dwarf the variable with the lower variance.

```
Accuracy of Support Vector Classifier: 84.8780487804878
```

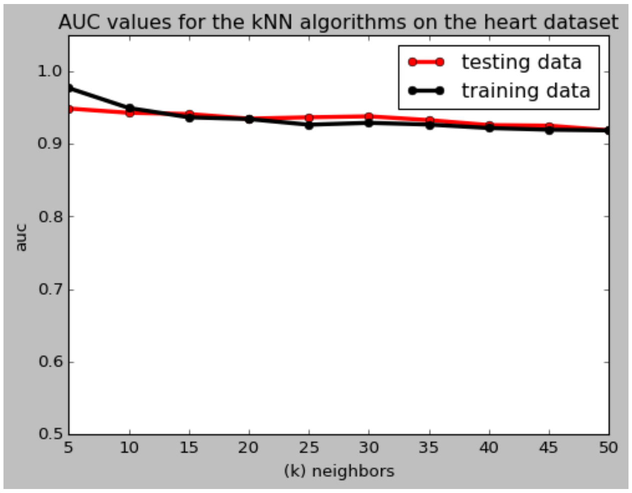


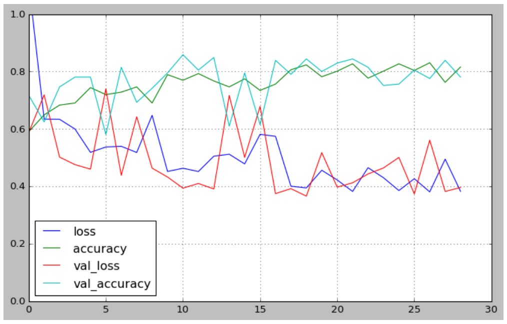
Figure 15: AUC at different depth and learning rate

```
              precision    recall  f1-score   support

           0       0.85      0.76      0.80        83
           1       0.85      0.91      0.88       122

    accuracy                           0.85       205
   macro avg       0.85      0.83      0.84       205
weighted avg       0.85      0.85      0.85       205
```

7 Artificial Neural Networks

We developed a heart disease classification system in which they integrated neural networks with an artificial neural network. The ANN model is based on tensorflow and keras.

```
    Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_3 (Dense)              (None, 52)                728
_________________________________________________________________
dense_4 (Dense)              (None, 26)                1378
_________________________________________________________________
dense_5 (Dense)              (None, 1)                 27
=================================================================
Total params: 2,133
```



Conclusion

1) Gradient Boost Classifier gives the best accuracy compared to other models.

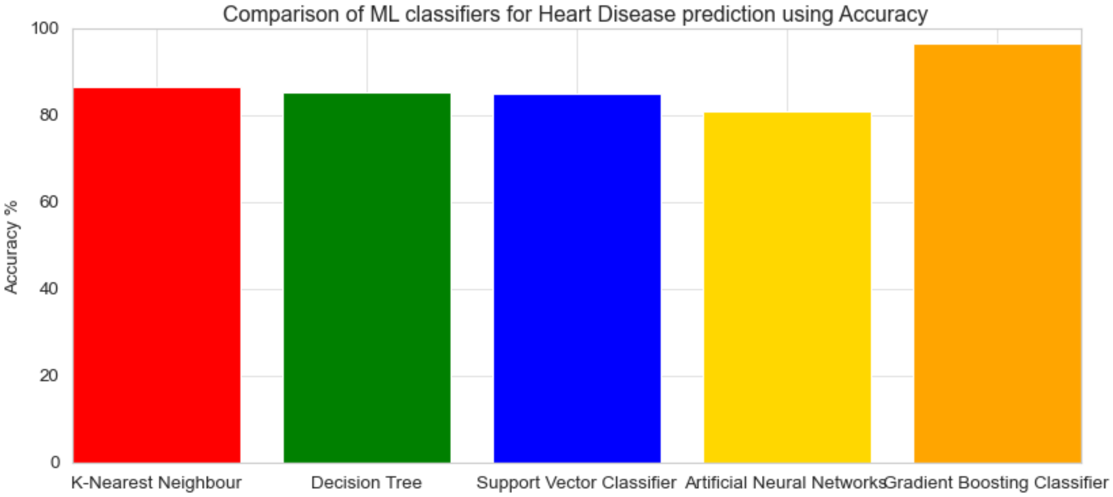
2) Exercise induced angina,Chest pain is major symptoms of heart attack.

3) Ensembling technique increase the accuracy of the model.

1. Out of the 13 features we examined, the top 4 significant features that helped us classify between a positive & negative Diagnosis were chest pain type (cp), maximum heart rate achieved (thalach), number of major vessels (ca), and ST depression induced by exercise relative to rest (oldpeak).

2. Our machine learning algorithm can now classify patients with Heart Disease. Now we can properly diagnose patients, & get them the help they needs to recover. By diagnosing detecting these features early, we may prevent worse symptoms from arising later.

3. Our Random Forest algorithm yields the highest accuracy, 80%. Any accuracy above 70% is considered good, but be careful because if your accuracy is extremely high, it may be too good to be true (an example of Over fitting). Thus, 80% is the ideal accuracy!



8 Prediction

Using the predicted model, we will be able to predict whether a patient has heart disease by providing selected features. For example, if a patient develops some symptoms as listed features, & you could input his vitals into the trained ML model to quickly obtain his risk of heart disease. Those vitals, including chese pain, resting blood pressure, fasting blood sugar, etc, are usually easily measured in smaller clinic. Based on this information, you can classify this patient with Heart Disease. early diagnosis of heart disease patient reduces the severe aftermath of heart disease by taking actions.