This document summarizes the results of Assignment 2 of IAI5101 (GNG5300) course.
The document first presents E. Model Comparison, Evaluation.
Later in the document, parts A(EDA), B(Feature Engineering), C(Model Development I) and D(Model Development II) are also presented.
Assignment2_IAI5101_Winter2022_ZainUrRehman.ipynb file is also uploaded on bright space.

E. Model Comparison, Evaluation

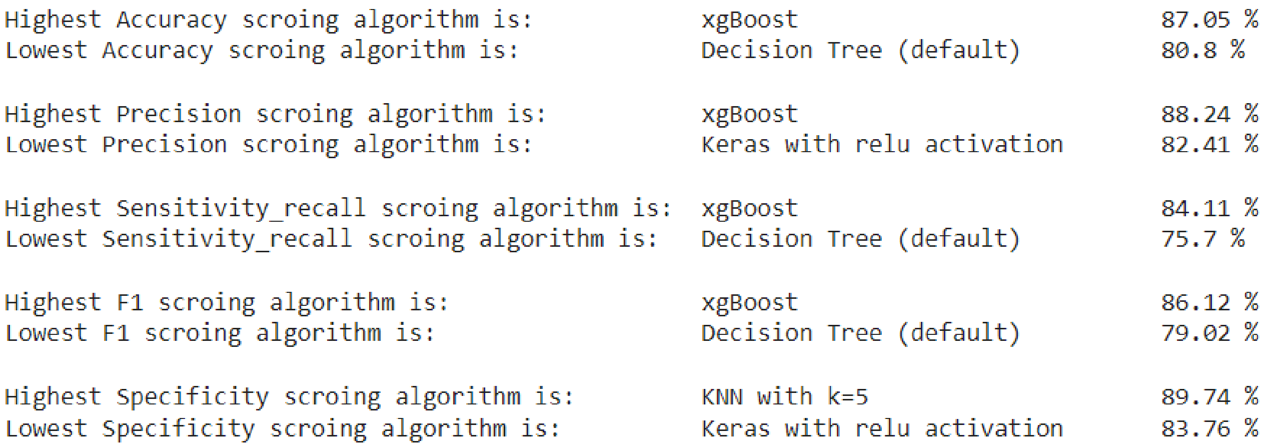
Summary:

- Performance of ensemble and deep neural classifiers were very close, with slightly better results for the neural network classifier. Among all the classifiers explored, xgBoost has the best performance and is the champion classifier.
- Soft and hard voting classifiers had similar results
- Decision Tree with optimal tree depth showed better results than Decision Tree default

Following classifiers were explored in Assignment 2:
1) KNN with k=5
2) SVM with rbf kernel
3) Decision Tree (default)
4) Decision Tree with Optimal Tree depth
5) XgBoost
6) gradBoost
7) Majority voting Soft (with classifiers 1,2,4,5)
8) Majority voting Hard (with classifiers 1,2,4,5)
9) Keras with tanh activation in first and second layer, sigmoid activation in the last output layer
10) Keras with relu activation in first and second layer, sigmoid activation in the last output layer
11) Keras with sigmoid activation in first and second layer, sigmoid activation in the last output layer

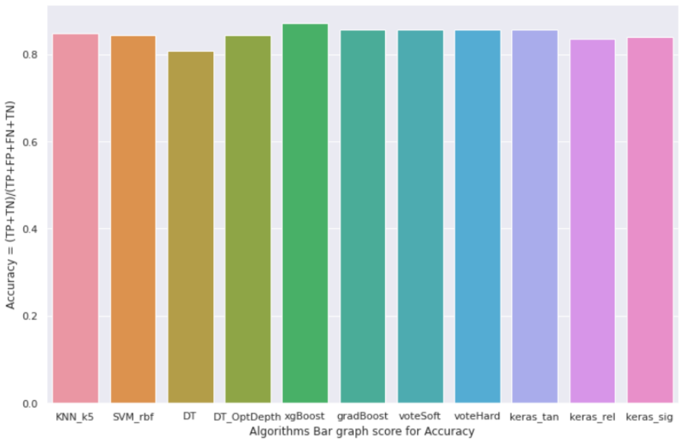
Please note that for Majority Voting Soft/Hard following classifiers (1,2,4,5 in above list) were used as required by the assignment:
1) KNN with k=5
2) SVM with rbf kernel
4) Decision Tree with Optimal Tree depth
5) XgBoost

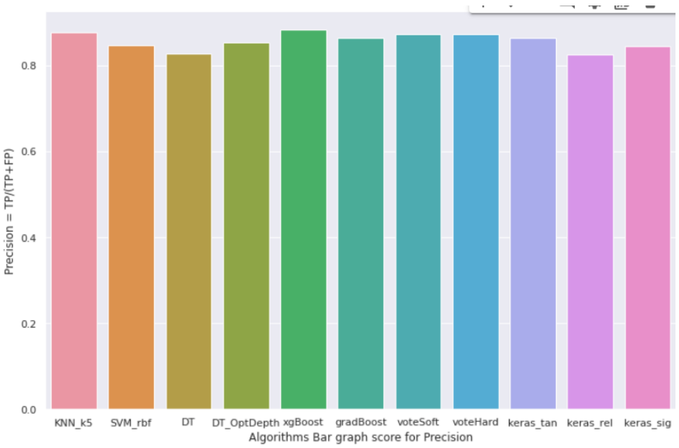
Classifier scores (min/max) for Accuracy, Precision, Sensitivity, F1 and Specificity:

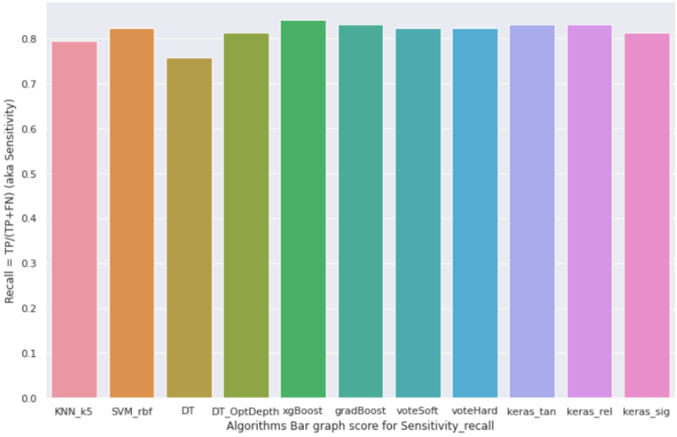


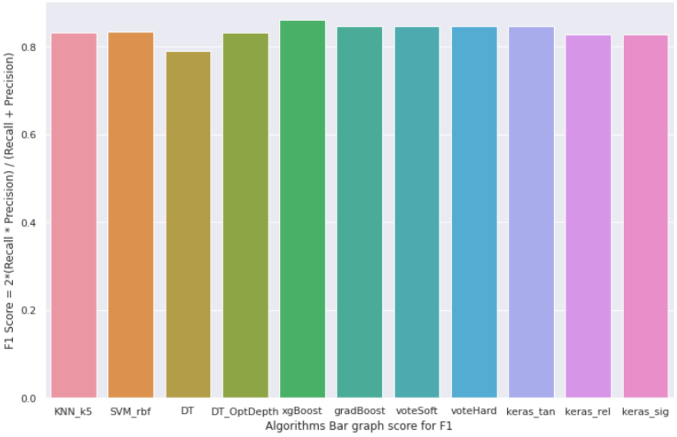
```
Highest Accuracy scroing algorithm is:           xgBoost                         87.05 %
Lowest Accuracy scroing algorithm is:            Decision Tree (default)         80.8 %

Highest Precision scroing algorithm is:          xgBoost                         88.24 %
Lowest Precision scroing algorithm is:           Keras with relu activation      82.41 %

Highest Sensitivity_recall scroing algorithm is: xgBoost                         84.11 %
Lowest Sensitivity_recall scroing algorithm is:  Decision Tree (default)         75.7 %

Highest F1 scroing algorithm is:                 xgBoost                         86.12 %
Lowest F1 scroing algorithm is:                  Decision Tree (default)         79.02 %

Highest Specificity scroing algorithm is:        KNN with k=5                    89.74 %
Lowest Specificity scroing algorithm is:         Keras with relu activation      83.76 %
```

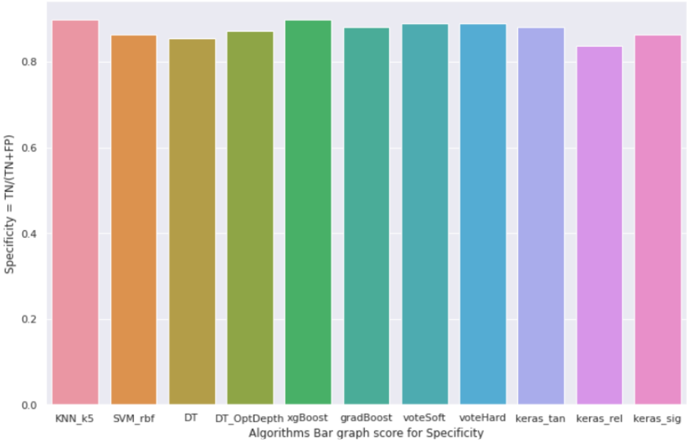
Bar graphs for Accuracy, Precision, Sensitivity, F1 and Specificity for all classifiers:



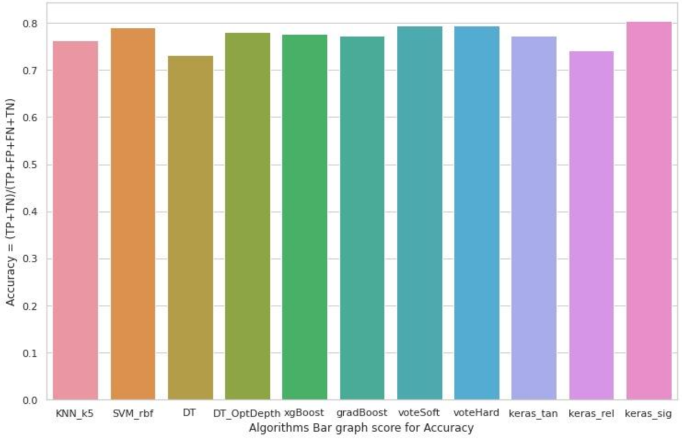


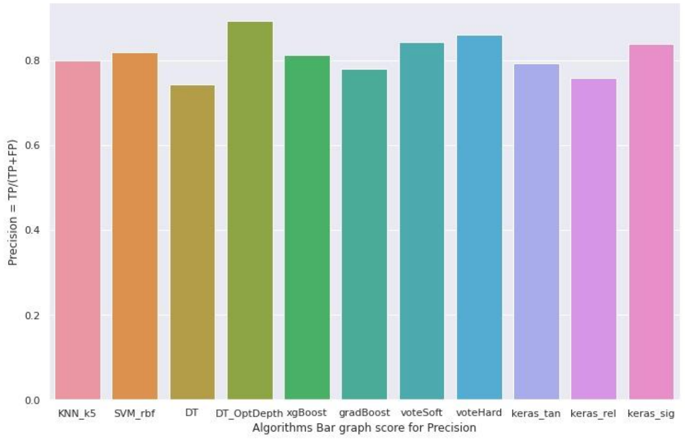


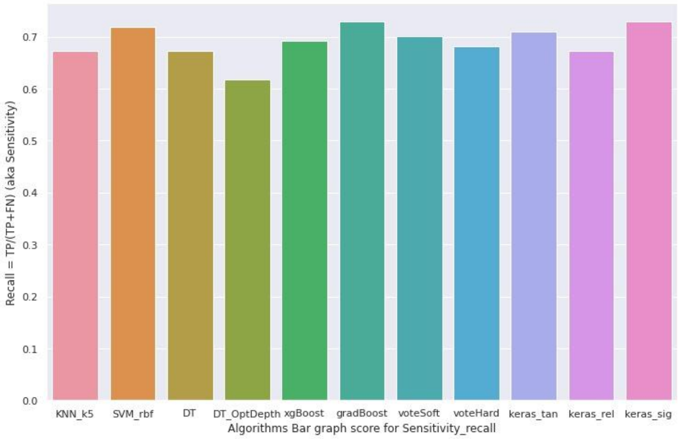


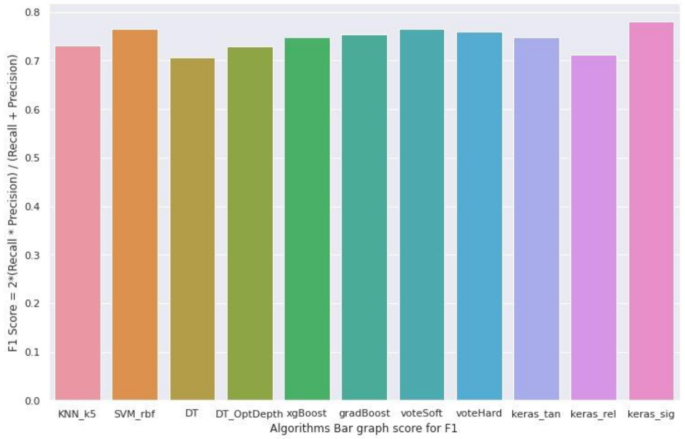


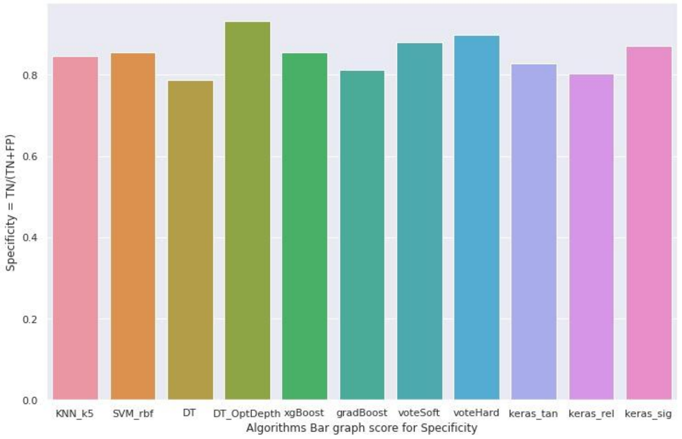
Accuracy, Precision, Sensitivity, F1 and Specificity scores get lower in general for all classifiers when only few features (Age,Sex, RestingBP,FastingBS,ExerciseAngina,Oldpeak) with higher correlation to HeartDisease are in X**.**





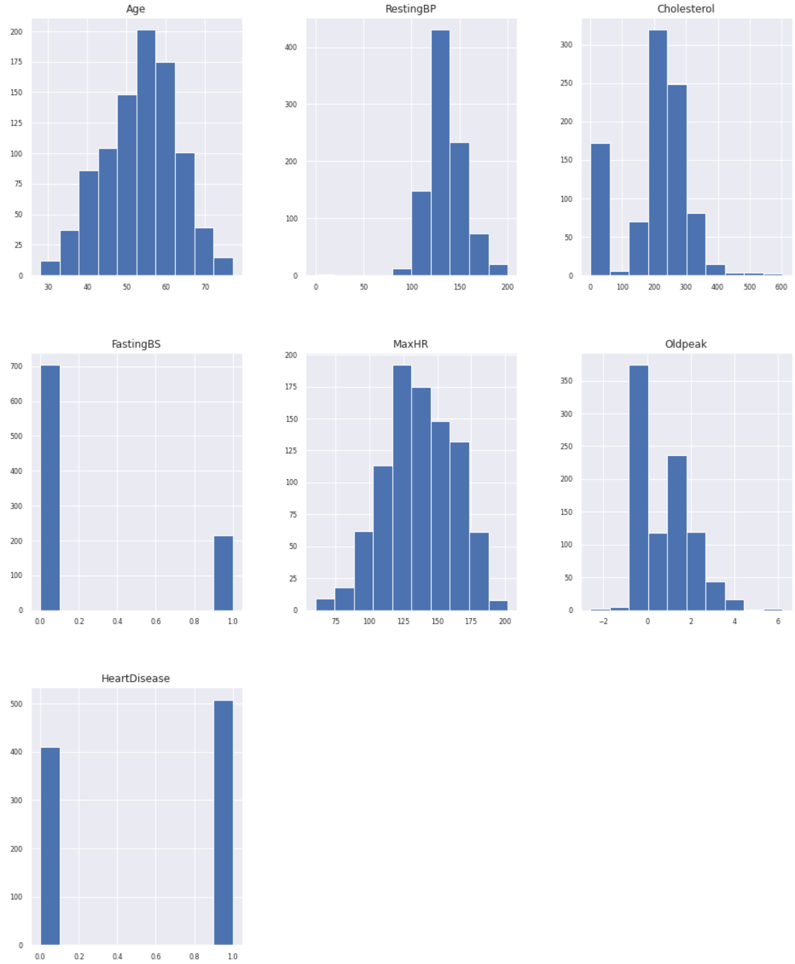




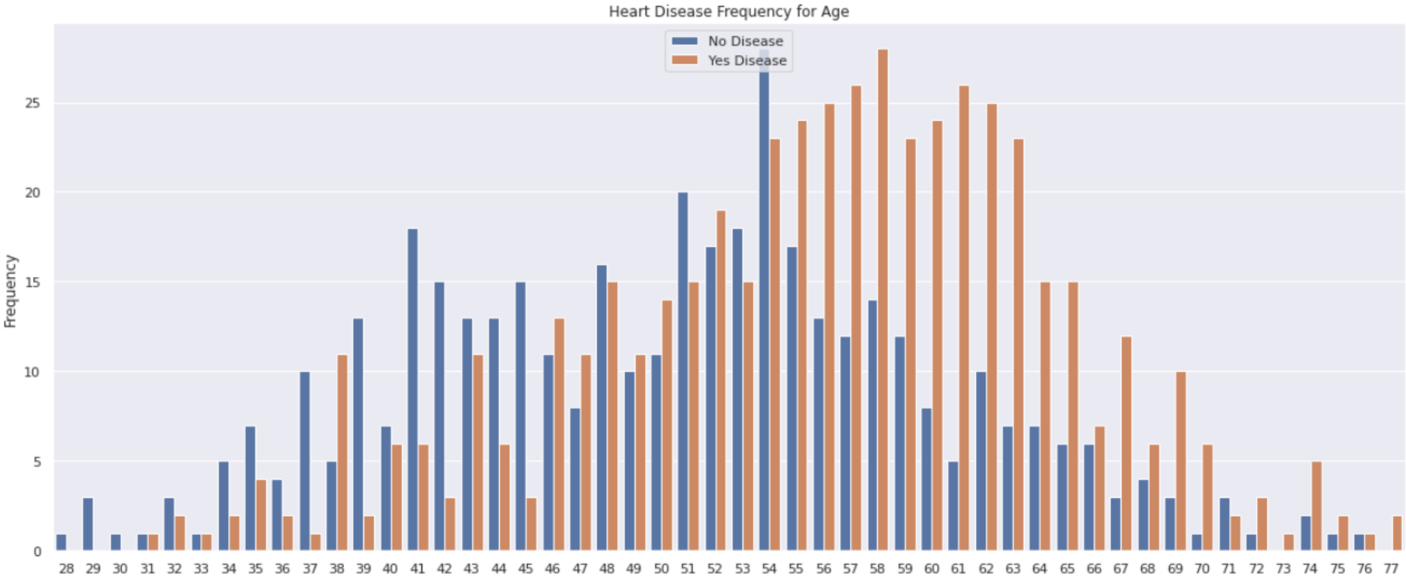


A,B. EDA and Feature Engineering

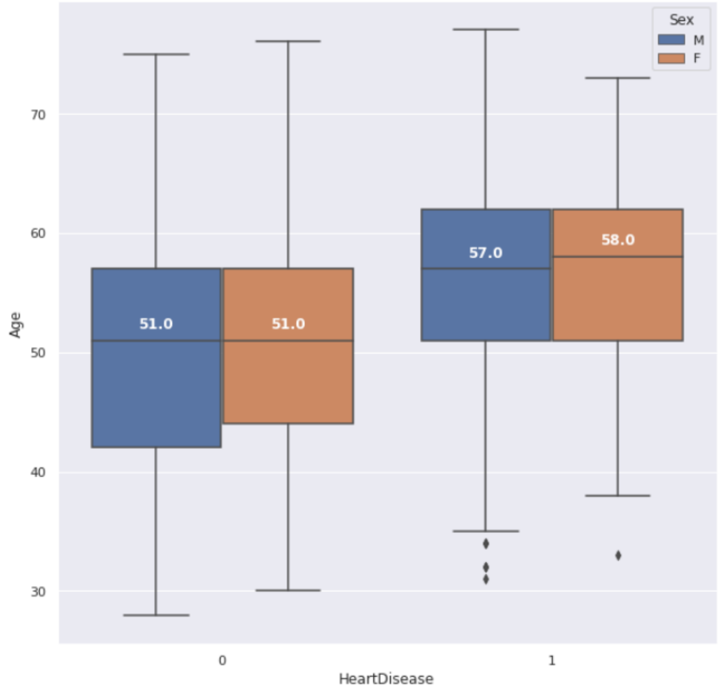
Univariate analysis



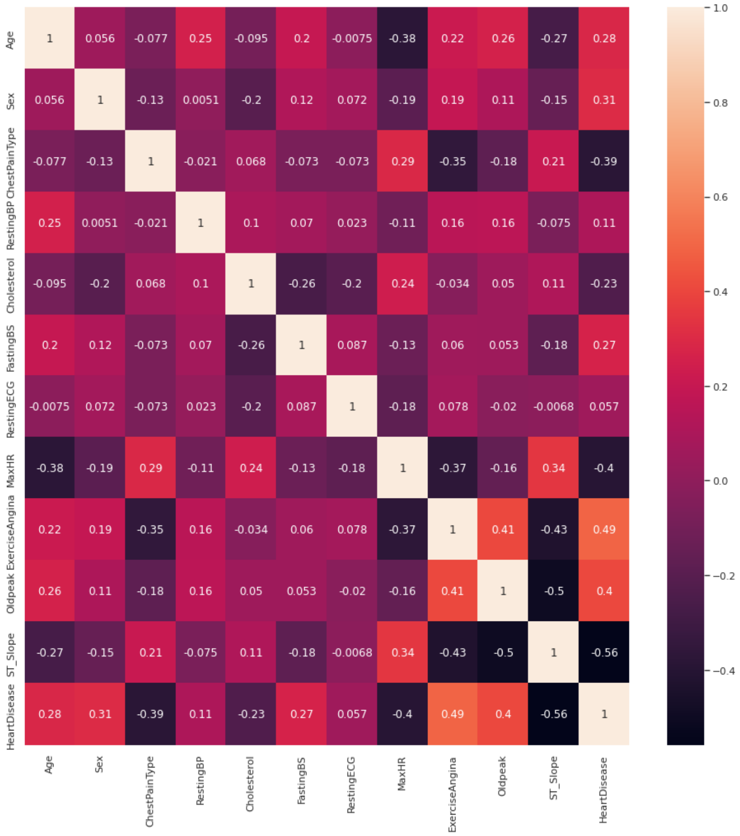
Bivariate analysis: Plot a histogram showing the age against the target variable (positive vs. negative cases)



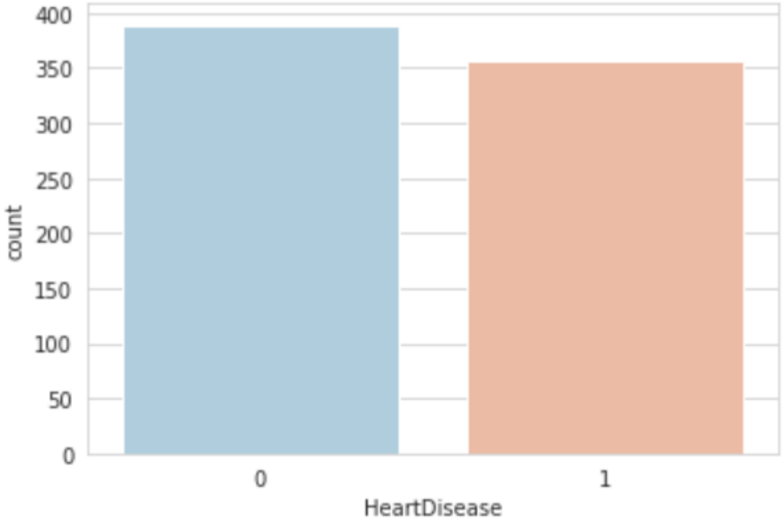
Compare the median age for male and female using a boxplot
median age positive female cases: 58
median age positive male cases: 57



Multivariate Analysis: Use a heatmap to check for correlation between predictor variables
These features have positive correlation with HeartDisease:
Age, Sex, RestingBP, FastingBS, ExerciseAngina, Oldpeak



Check for class imbalance.
There seems no class imbalance.
For no heart disease, the count is around 380.
For yes heard disease, the count is around 350.
Not much difference so this looks like a balanced dataset.



Following Feature Engineering points also checked and implemented:

label encoder to convert non-numeric data into numeric data
Check for duplicates & missing values
Handled the outliers (0 cholesterol, negative oldpeak)
Scale the data using a standard scale as there are features that have variation and measured in different units

C. Model Development I

Following models part of Ensemble method:

KNN_k5_classifier with k=5
SVM_classifier with kernel='rbf'
DecisionTree_classifier_grid with Optimal depth of the decision tree: max_depth: 1
xgBoost_classifier

Both Ensemble approaches votesoft_classifier and votehard_classifier are explored. Results for soft and hard voting are similar.

Other classifiers explored that are not part of Ensemble method:

DecisionTree_classifier (default without any depth)
gradBoost_classifier

```
<>-----The 5 fold KNN_k5_classifier_Score cross validation: ----------<>
[0.82857143 0.875      0.84615385 0.875      0.78846154]

<>-----KNN_k5_classifier_Score Mean and Standard Deviation: ----------<>
0.8426373626373627 0.03237257072501307

<>-----KNN_k5_classifier confusion_matrix: ---------------------------<>
[[105  12]
[ 22  85]]

<>-----KNN_k5_classifier accuracy score: -----------------------------------<>
0.8482142857142857

<>-----KNN_k5_classifier Classification report: ----------------------<>
precision    recall  f1-score   support

0       0.83      0.90      0.86       117
1       0.88      0.79      0.83       107

accuracy                           0.85       224
macro avg       0.85      0.85      0.85       224
weighted avg       0.85      0.85      0.85       224

<>-------------- The 5 fold SVM_classifier_Score cross validation: -------<>
[0.88571429 0.86538462 0.89423077 0.91346154 0.79807692]

<>----------- SVM_classifier_Score Mean and Standard Deviation: ----------<>
0.8713736263736264 0.03976875859422301

<>----------- SVM_classifier confusion_matrix: ---------------------------<>
[[101  16]
 [ 19  88]]

<>----------- SVM_classifier accuracy: -----------------------------------<>
0.84375

<>----------- SVM_classifier Classification report: ----------------------<>
              precision    recall  f1-score   support

           0       0.84      0.86      0.85       117
           1       0.85      0.82      0.83       107

    accuracy                           0.84       224
   macro avg       0.84      0.84      0.84       224
weighted avg       0.84      0.84      0.84       224
```

```
<>----- The 5 fold DecisionTree_classifier_Score cross validation: ----------<>
[0.78095238 0.76923077 0.81730769 0.79807692 0.72115385]

<>----- DecisionTree_classifier_Score Mean and Standard Deviation: ----------<>
0.7773443223443224 0.03244419552870659

<>----- DecisionTree_classifier confusion_matrix: ---------------------------<>
[[101  16]
 [ 28  79]]

<>----- DecisionTree_classifier accuracy: -----------------------------------<>
0.8035714285714286

<>----- DecisionTree_classifier Classification report: ----------------------<>
              precision    recall  f1-score   support

           0       0.78      0.86      0.82       117
           1       0.83      0.74      0.78       107

    accuracy                           0.80       224
   macro avg       0.81      0.80      0.80       224
weighted avg       0.81      0.80      0.80       224


Optimal depth of the decision tree:  {'max_depth': 1}
<>----- The 5 fold DecisionTree_classifier_grid_Score cross validation: ----------<>
[0.84761905 0.80769231 0.80769231 0.79807692 0.81730769]

<>----- DecisionTree_classifier_grid_Score Mean and Standard Deviation: ----------<>
0.8156776556776558 0.017089335221392138

<>----- DecisionTree_classifier_grid confusion_matrix: ---------------------------<>
[[96 21]
 [16 91]]

<>----- DecisionTree_classifier_grid accuracy: -----------------------------------<>
0.8348214285714286

<>----- DecisionTree_classifier_grid Classification report: ----------------------<>
              precision    recall  f1-score   support

           0       0.86      0.82      0.84       117
           1       0.81      0.85      0.83       107

    accuracy                           0.83       224
   macro avg       0.83      0.84      0.83       224
weighted avg       0.84      0.83      0.83       224



<>----- The 5 fold xgBoost_classifier_Score cross validation: ----------<>
[0.84761905 0.86538462 0.875      0.91346154 0.81730769]

<>----- xgBoost_classifier_Score Mean and Standard Deviation: ----------<>
0.8637545787545788 0.03167827958506335

<>----- xgBoost_classifier confusion_matrix: ---------------------------<>
[[105  12]
 [ 17  90]]

<>----- xgBoost_classifier accuracy: -----------------------------------<>
0.8705357142857143

<>----- xgBoost_classifier Classification report: ----------------------<>
              precision    recall  f1-score   support

           0       0.86      0.90      0.88       117
           1       0.88      0.84      0.86       107

    accuracy                           0.87       224
   macro avg       0.87      0.87      0.87       224
weighted avg       0.87      0.87      0.87       224
```

```
<>----- The 5 fold gradBoost_classifier_Score cross validation: ----------<>
[0.82857143 0.83653846 0.875      0.86538462 0.84615385]

<>----- gradBoost_classifier_Score Mean and Standard Deviation: ----------<>
0.8503296703296703 0.017414548909060888

<>----- gradBoost_classifier confusion_matrix: ---------------------------<>
[[103  14]
 [ 18  89]]

<>----- gradBoost_classifier accuracy: -----------------------------------<>
0.8571428571428571

<>----- gradBoost_classifier Classification report: ----------------------<>
              precision    recall  f1-score   support

           0       0.85      0.88      0.87       117
           1       0.86      0.83      0.85       107

    accuracy                           0.86       224
   macro avg       0.86      0.86      0.86       224
weighted avg       0.86      0.86      0.86       224
```

```
<>----- The 5 fold votesoft_classifier_Score cross validation: ----------<>
[0.87619048 0.86538462 0.875      0.89423077 0.83653846]

<>----- votesoft_classifier_Score Mean and Standard Deviation: ----------<>
0.8694688644688645 0.01892098295321054

<>----- votesoft_classifier confusion_matrix: ---------------------------<>
[[104  13]
 [ 19  88]]

<>----- votesoft_classifier accuracy: -----------------------------------<>
0.8571428571428571

<>----- votesoft_classifier Classification report: ----------------------<>
              precision    recall  f1-score   support

           0       0.85      0.89      0.87       117
           1       0.87      0.82      0.85       107

    accuracy                           0.86       224
   macro avg       0.86      0.86      0.86       224
weighted avg       0.86      0.86      0.86       224
```

```
<>----- The 5 fold votehard_classifier_Score cross validation: ----------<>
[0.87619048 0.82692308 0.89423077 0.89423077 0.82692308]

<>----- votehard_classifier_Score Mean and Standard Deviation: ----------<>
0.8636996336996337 0.030741996930748625

<>----- votehard_classifier confusion_matrix: ---------------------------<>
[[104  13]
 [ 19  88]]

<>----- votehard_classifier accuracy: -----------------------------------<>
0.8571428571428571

<>----- votehard_classifier Classification report: ----------------------<>
              precision    recall  f1-score   support

           0       0.85      0.89      0.87       117
           1       0.87      0.82      0.85       107

    accuracy                           0.86       224
   macro avg       0.86      0.86      0.86       224
weighted avg       0.86      0.86      0.86       224
```

D. Model Development II

3 models are explored with different activation functions in the first/second hidden layers of the model:

Keras_model1_clf with activation='tanh'
Keras_model2_clf with activation='relu'
Keras_model3_clf with activation='sigmoid'

In the first line, we set the model as sequential.
Then, we add the three fully connected dense layers: two hidden and one output.
These are defined using the dense class. The first level has a dimension of 11 which corresponds to 11 column attributes in X.
The first and second layers has 30, 20 nodes/neurons. More nodes and layers mean more capacity for the network to learn. The output layer has a single neuron (output) and the sigmoid activation function suited for binary classification problems

3 activation functions are explored:

Keras_model1_clf with first and second layer activation='tanh', output layer activation=sigmoid
Keras_model2_clf with first and second layer activation='relu', output layer activation=sigmoid
Keras_model3_clf with first and second layer activation='sigmoid', output layer activation=sigmoid

```
<>-----Keras_model1_clf score: ---------------------------<>
Keras_model1_clf score =  0.8348214030265808

<>-----Keras_model1_clf confusion_matrix: ---------------------------<>
[[104  13]
 [ 24  83]]

<>-----Keras_model1_clf accuracy: ----------------------------------<>
0.8348214285714286

<>-----Keras_model1_clf Classification report: ----------------------<>
              precision    recall  f1-score   support

           0       0.81      0.89      0.85       117
           1       0.86      0.78      0.82       107

    accuracy                           0.83       224
   macro avg       0.84      0.83      0.83       224
weighted avg       0.84      0.83      0.83       224

<>-----Keras_model2_clf score: ---------------------------<>
Keras_model2_clf score =  0.8526785969734192

<>-----Keras_model2_clf confusion_matrix: ---------------------------<>
[[105  12]
 [ 21  86]]

<>-----Keras_model2_clf accuracy: ----------------------------------<>
0.8526785714285714

<>-----Keras_model2_clf Classification report: ----------------------<>
              precision    recall  f1-score   support

           0       0.83      0.90      0.86       117
           1       0.88      0.80      0.84       107

    accuracy                           0.85       224
   macro avg       0.86      0.85      0.85       224
weighted avg       0.85      0.85      0.85       224
```

```
<>-----Keras_model3_clf score: ---------------------------<>
Keras_model3_clf score =  0.8392857313156128

<>-----Keras_model3_clf confusion_matrix: ---------------------------<>
[[102  15]
 [ 21  86]]

<>-----Keras_model3_clf accuracy: -----------------------------------<>
0.8392857142857143

<>-----Keras_model3_clf Classification report: ----------------------<>
              precision    recall  f1-score   support

           0       0.83      0.87      0.85       117
           1       0.85      0.80      0.83       107

    accuracy                           0.84       224
   macro avg       0.84      0.84      0.84       224
weighted avg       0.84      0.84      0.84       224
```