دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

Kasra Khalafi: 9531306

Homework 3

Data Mining

Dr.Ehsan Nazerfard

Winter Term 2019

1-

Gradient Tree Boosting:

Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.

It is similar to AdaBoost as they both use ensemble of decision tree but despite AdaBoost it has depth larger than one.

Random Forest:

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (Classification) or mean prediction (Regression) of the individual trees.

Each classifier in the ensemble is generated using a random selection of attributes at each node to determine the split. During classification, each tree votes and the most popular class is returned.

Two Methods to construct Random Forest:

a) Forest-RI (random input selection)
b) Forest-RC (random linear combinations)

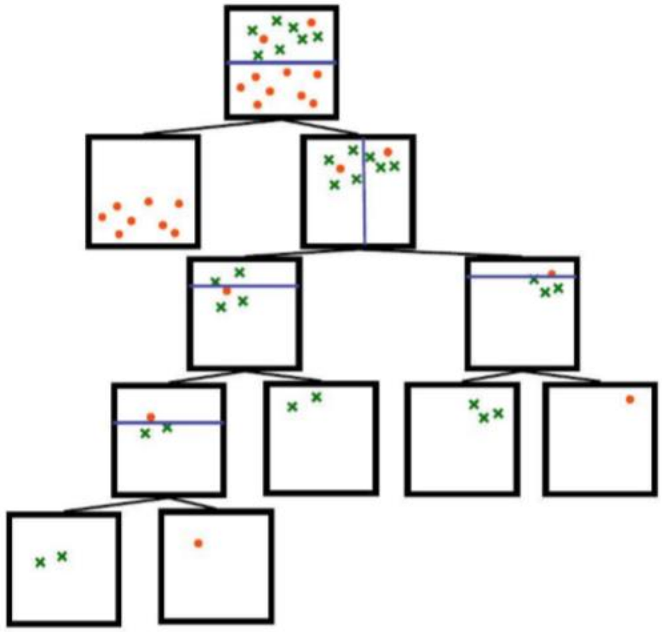
Comparison:

Random Forest and Gradient Tree Boosting are both comparable in accuracy to each other, but Random Forest is more robust to errors and outliers. Also Random Forest correct for decision tree's habit of overfitting to their training set.

Random Forest is insensitive to the number of attributes selected for consideration at each split, and is faster than bagging or boosting as also Gradient Tree Boosting.

In Gradient Tree Boosting, Training takes longer time because of the fact that trees are built sequentially.

2-



As we see for above picture, firs division operates well for the left part in first depth but for the right one it is needed to be divided again cause there are some different labeled data among same labeled. If those 2 red circule shown data among green multiplication sign are considered as noise, then our classification would be finished in the first depth and there would be no need to increment depth to separate data which is time consuming, and it is prone to overfitting.

For avoiding overfitting there are two main approaches: prepruning and post pruning:

- **Prepruning**: Halt tree construction early. We should not split a node if it would result in the goodness measure falling below a threshold. The backward of prepruning is that it is difficult to choose an appropriate threshold.
- **Postpruning**: Remove branchesfrom a fully grown tree. We should get a sequence of progressively pruned trees, then use a set of data different from the training data to decide which is the best pruned tree.

3- Accuracy, the proportion of correct classifications among all classifications, is very simple and very intuitive measure, yet it may be a poor measure for **unbalanced data**, and it is improper scoring rule. accuracy is discontinuous in the threshold: moving the threshold a tiny little bit may make one (or multiple) predictions change classes and change the entire accuracy by a discrete amount.

Below are some proper evaluation measures like precision and recall formula:

Metric	Formula
True positive rate, recall	$\frac{\text{TP}}{\text{TP+FN}}$
False positive rate	$\frac{\text{FP}}{\text{FP+TN}}$
Precision	$\frac{\text{TP}}{\text{TP+FP}}$
Accuracy	$\frac{\text{TP+TN}}{\text{TP+TN+FP+FN}}$
F-measure	$\frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$

Also in most cases F-Measure is used as evaluation measure cause it uses both precision and recall simultaneously.

4- As supposed in the question, there are 25 base independent classifiers and each has error rate = 35%. For calculating probability of wrong prediction for the ensemble classifier we will use bionomial random variables as equation bellow:

$$P(X=x)=\binom{n}{x}p^x(1-p)^{n-x}$$

Because voting is used to classify the records, if we consider that all classifiers have same impact on calculation of voting, then atleast we must have 13 classifiers out of 25 to choose the wrong ensemble classifier.

So we should compute a sum over 13 to 25 on bionomial random variables for calculation of probability of wrong prediction for the ensemble classifier:

$$\sum_{i=13}^{25}\binom{25}{i}e^i(1-e)^{25-i} = 0.06$$

5- Below is the table of data:

Habitat	Cap Color	Cap Shape	Odor	Edible?
Woods	Red	Flat	None	**Poisonous**
Woods	Red	Flat	Foul	**Poisonous**
Grasses	Red	Flat	None	**Edible**
Leaves	Green	Flat	None	**Edible**
Leaves	White	Convex	None	**Edible**
Leaves	White	Convex	Foul	**Poisonous**
Grasses	White	Convex	Foul	**Edible**
Woods	Green	Flat	None	**Poisonous**
Woods	White	Convex	None	**Edible**
Leaves	Green	Convex	None	**Edible**
Woods	Green	Convex	Foul	**Edible**
Grasses	Green	Flat	Foul	**Edible**
Grasses	Red	Convex	None	**Edible**
Leaves	Green	Flat	Foul	**Poisonous**

a) here is step by step calculation to show which feature would be selected:

Q5

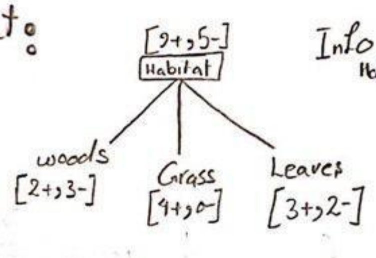
First Layer

Edible → +
Poisonous → −
shown as

$$Entropy(D) = Info(D) \rightsquigarrow D[9+, 5-]$$

$$\Rightarrow Entropy(D) = -\sum_{i=1}^{m} P_i \log_2(P_i) = -\frac{9}{14}\log_2\left(\frac{9}{4}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right) = 0.940$$

Habitat:



$$Info_{Habitat}(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D)$$

$$= \frac{5}{14}\left(-\frac{2}{5}\log_2\left(\frac{2}{5}\right) - \frac{3}{5}\log_2\left(\frac{3}{5}\right)\right)$$

$$+ \frac{4}{14}\left(-\frac{4}{4}\log_2\left(\frac{4}{4}\right) - \frac{0}{4}\log(0)\right)$$

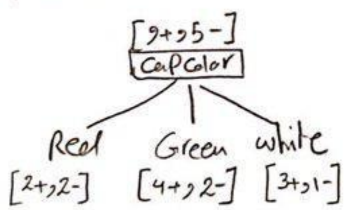
$$+ \frac{5}{14}\left(-\frac{3}{5}\log_2\left(\frac{3}{5}\right) - \frac{2}{5}\log_2\left(\frac{2}{5}\right)\right)$$

$$= 0.692$$

✓ this one is selected ✓

$$\Rightarrow \boxed{Gain(D, Habitat) = 0.248} \rightarrow 0.940 - 0.692$$

Cap Color:



$$Info_{Capcolor}(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D)$$

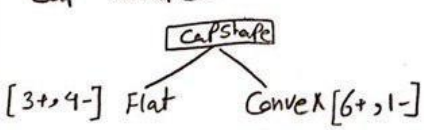
$$= \frac{4}{14}\left(-\frac{2}{4}\log_2\left(\frac{2}{4}\right) - \frac{2}{4}\log_2\left(\frac{2}{4}\right)\right)$$

$$+ \frac{6}{14}\left(-\frac{4}{6}\log_2\left(\frac{4}{6}\right) - \frac{2}{6}\log_2\left(\frac{2}{6}\right)\right)$$

$$+ \frac{4}{14}\left(-\frac{3}{4}\log_2\left(\frac{3}{4}\right) - \frac{1}{4}\log_2\left(\frac{1}{4}\right)\right) = 0.909$$

$$\rightarrow \boxed{Gain(D, Cap\ Color) = 0.031}$$

Cap shape:



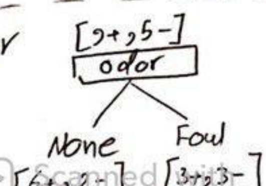
$$Info_{Shape}(D) = \frac{7}{14}\left(\frac{-3}{7}\log_2\left(\frac{3}{7}\right) + \frac{-4}{7}\log_2\left(\frac{4}{7}\right)\right)$$

$$+ \frac{7}{14}\left(\frac{-6}{7}\log_2\left(\frac{6}{7}\right) + \frac{-1}{7}\log_2\left(\frac{1}{7}\right)\right) = 0.787$$

$$\boxed{Gain(D, Cap\ Shape) = 0.153}$$

---

odor



$$Info_{Odor}(D) = \frac{8}{14}\left(\frac{-6}{8}\log_2\left(\frac{6}{8}\right) + \frac{-2}{8}\log_2\left(\frac{2}{8}\right)\right)$$

$$+ \frac{6}{14}\left(-\frac{3}{6}\log_2\left(\frac{1}{2}\right) - \frac{3}{6}\log_2\left(\frac{3}{6}\right)\right) = 0.891$$

$$\Rightarrow \boxed{Gain(D, odor) = 0.049}$$

*Second Layer

woods [2+,3-] | Grass [4+,0-] | Leaves [3+,2-]

$$Gain(D_1) = Info(D_1) = ?$$

$$Gain(D_1) = -\frac{2}{5}\left(\log_2 \frac{2}{5}\right) - \frac{3}{5}\log_2\left(\frac{3}{5}\right) = 0.970$$

cap Color:

woods

Color

Red [0+,2-] | Green [1+,1-] | white [1+,0-]

$$Info_{color}(D_1) = \frac{2}{5}\left(-\frac{2}{2}\log_2\left(\frac{2}{2}\right)\right) + \frac{2}{5}\left(-\frac{1}{2}\log_2\left(\frac{1}{2}\right) + -\frac{1}{2}\log_2\left(\frac{1}{2}\right)\right) + \frac{1}{5}\left(-\frac{1}{1}\log_2(1)\right) = 0.4$$

$$\boxed{Gain(D_1, Color) = 0.57}$$

cap shape:

Shape

Flat [0+,3-] | Convex [2+,0-]

$$Info_{shape}(D_1) = \frac{3}{5}\left(-\frac{3}{3}\log_2\left(\frac{3}{3}\right)\right) + \frac{2}{5}\left(-\frac{2}{2}\log_2\left(\frac{2}{2}\right)\right) = 0$$

$$\Rightarrow \boxed{Gain(D_1, shape) = 0.970}$$ ✓

this one is selected

odor

Odor

None [1+,2-] | Foul [1+,1-]

$$Info_{odor}(D_1) = \frac{3}{5}\left(-\frac{1}{3}\log_2\left(\frac{1}{3}\right) + -\frac{2}{3}\log_2\left(\frac{2}{3}\right)\right) + \frac{2}{5}\left(-\frac{1}{2}\log_2\left(\frac{1}{2}\right) + -\frac{1}{2}\log_2\left(\frac{1}{2}\right)\right) = 0.95$$

$$\Rightarrow \boxed{Gain(D_1, odor) = 0.02}$$

Seanel Layer again → but for Leaves

$$Gain(D_2) = Info(D_2) = \frac{-3}{5}\log_2\left(\frac{3}{5}\right) - \frac{2}{5}\log\left(\frac{2}{5}\right) = \boxed{0.970}$$

capcolor :

$$Info_{color}(D_2) = \frac{3}{5}\left(\frac{3}{3}\log_2\left(\frac{3}{3}\right)\right)$$
$$+ \frac{2}{5}\left(\frac{-1}{2}\log_2(0.5) - \frac{1}{2}\log_2\left(\frac{1}{2}\right)\right) = 0.4$$

$$\boxed{Gain(D_2, Color) = 0.570}$$

---

* cap shape:

$$Info_{shape}(D_2) = \frac{2}{5}\left(\frac{-1}{2}\log_2\left(\frac{1}{2}\right) + \frac{-1}{2}\log_2(0.5)\right)$$
$$+ \frac{3}{5}\left(\frac{-2}{9}\log_2\left(\frac{2}{3}\right) + \frac{-1}{3}\log_2\left(\frac{1}{3}\right)\right) = 0.950$$

$$\Rightarrow \boxed{Gain(D_2, Shape) = 0.02}$$

---

* odor:

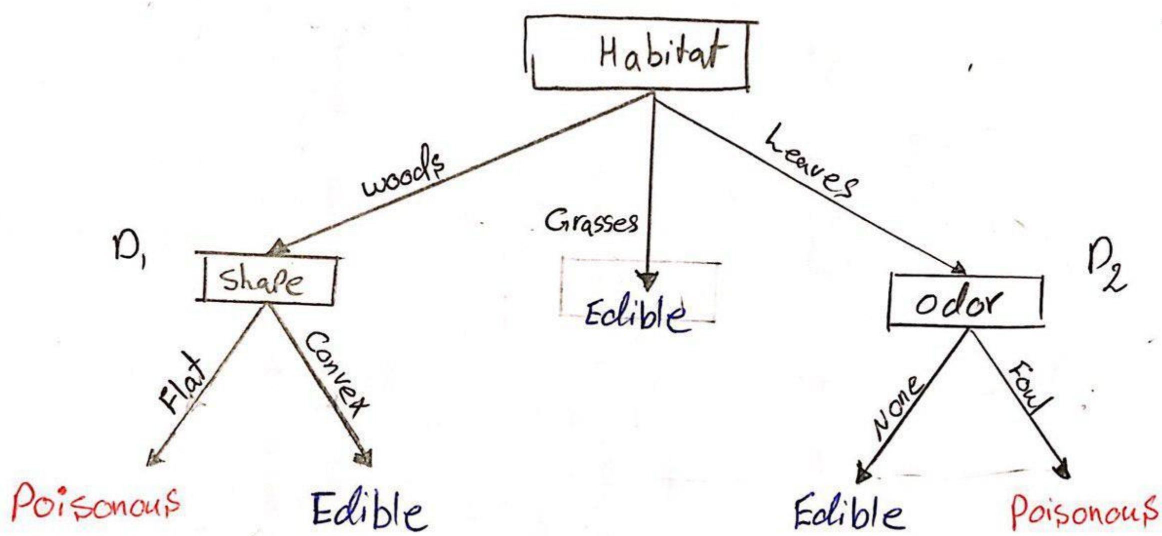
$$Info_{odor}(D_2) = \frac{3}{5}\left(\frac{3}{3}\log_2\left(\frac{3}{3}\right)\right)$$
$$+ \frac{2}{5}\left(\frac{2}{2}\log_2\left(\frac{2}{2}\right)\right) = 0$$

$$\Rightarrow \boxed{Gain(D_2, odor) = 0.970}$$ ✓

→ this one is selected

and also here calculation is finished :)

b) Final decision tree would be illustrated as :



6- There is a dataset with three Boolean features A, B, C and class attribute Y.

a) For writing an expression for P(y|a,b,c) in terms of P(y), P(b|y), and P(c|y):

we Knew that : $P(H|X) = \frac{P(X|H)\,P(H)}{P(X)}$

$\xrightarrow[\text{above}]{\text{from Eq.}} P(y|a,b,c) = \frac{P(a,b,c|y).P(y)}{P(a,b,c)}$ (*)

If attributes are Independent $\Rightarrow P(a,b|X) = P(a|X).P(b|X) =$ (**)

(**) & (*) $\longrightarrow$ $\boxed{P(y|a,b,c) = \frac{P(a|y).P(b|y).P(c|y).P(y)}{P(a,b,y)}}$

b) For a given observation: A = false, B = true, C = false we have:

* A = False & B = True & C = False *

$$\begin{cases} P(yes) = \frac{5}{8} = 0.625 \\ P(NO) = \frac{3}{8} = 0.375 \end{cases}$$

$$\begin{cases} P(A=False \mid Y=OK) = \frac{4}{5} = 0.8 \\ P(A=False \mid Y=bad) = \frac{1}{3} = 0.\bar{3} \\ P(B=True \mid Y=OK) = \frac{2}{5} = 0.4 \\ P(B=True \mid Y=bad) = \frac{2}{3} = 0.\bar{6} \\ P(C=False \mid Y=OK) = \frac{4}{5} = 0.8 \\ P(C=False \mid Y=bad) = \frac{1}{5} = 0.2 \end{cases}$$

$\longrightarrow P(X \mid Y=OK) = 0.8 \times 0.4 \times 0.8$

$\rightarrow P(x \mid Y=OK)P(OK) = 0.8 \times 0.4 \times 0.8 \times 0.625 = \boxed{0.16}$ ✓

$\longrightarrow P(X \mid Y=bad) = 0.\bar{3} \times 0.\bar{6} \times 0.2$

$\rightarrow P(X \mid Y=bad)P(bad) = 0.\bar{3} \times 0.\bar{6} \times 0.2 \times 0.375 \simeq \boxed{0.016}$

$\longrightarrow$ The answer is OK

IMPLEMETATION 1: Decision Tree

In this implementation, heart diseas should have been classified with decision tree using sklearn library.

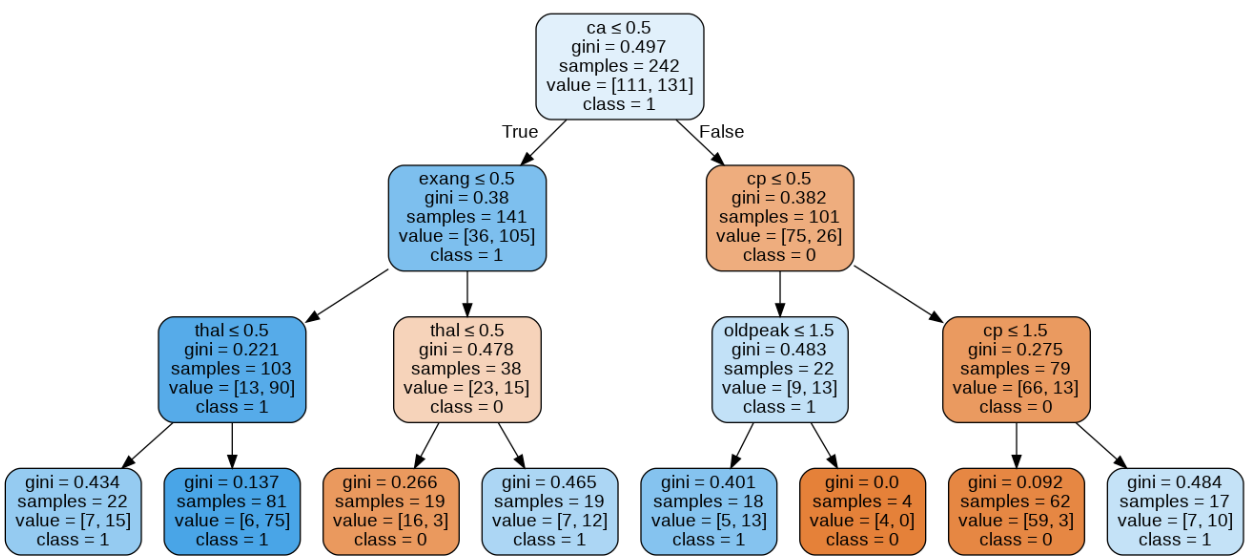
Implementation consists of 3 main parts:

- Preparing Data: Data given are not encoded. Data should be preprocessed with converting numerical features to categories each representing possible value ranges. So everything is converted to digits between 0 to 9.
- Spliting Data: in this section, Data should be splitted into two training and test parts with ratio of 8 to 2. Also Data should be shuffled and then split into to sections.
- Training and Classification: by sklearn library, decision tree classification is implemented with different criterion, max_depth and min_sample_split.

Here are some results of decision tree classification with different parameters:

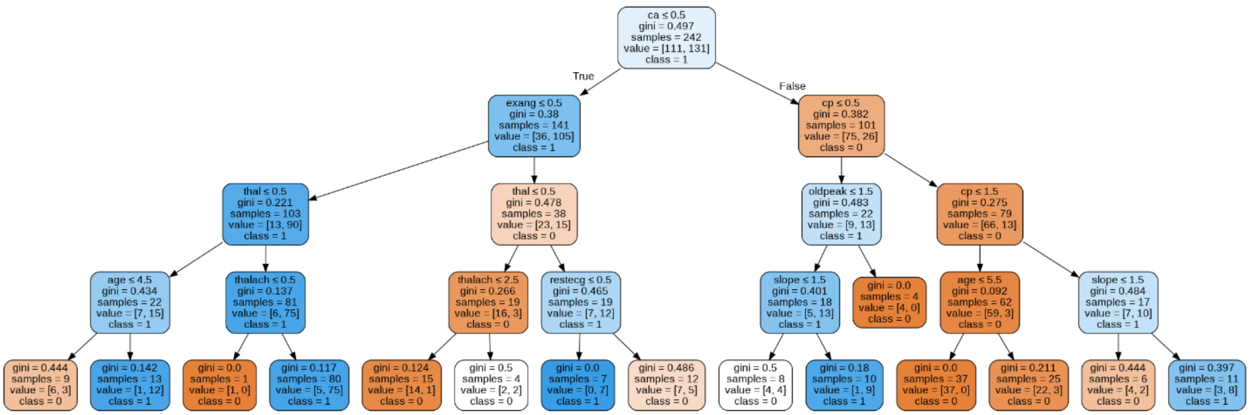
------------------------------ Criterion: gini, Max_Depth: 3, Min_Split: 2 -------------------------------------

```
              precision    recall  f1-score   support

           0       0.85      0.63      0.72        27
           1       0.76      0.91      0.83        34

    accuracy                           0.79        61
   macro avg       0.80      0.77      0.78        61
weighted avg       0.80      0.79      0.78        61
```



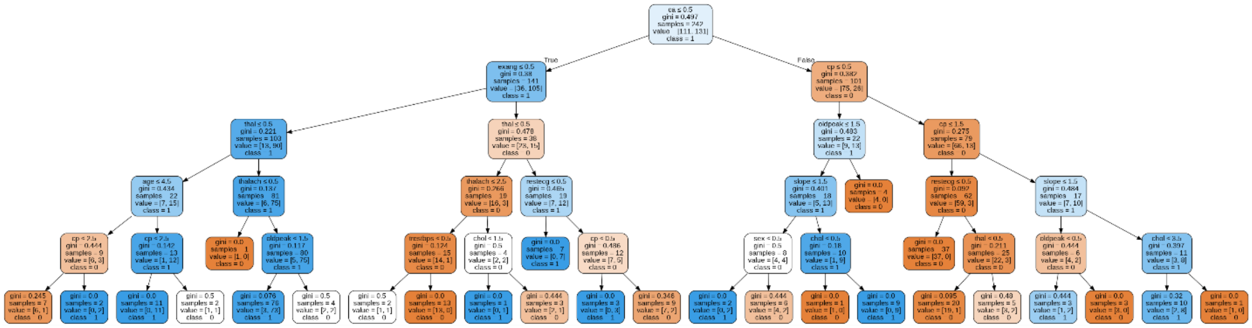
------------------------------ Criterion: gini, Max_Depth: 4, Min_Split: 2 -------------------------------------

	precision	recall	f1-score	support
0	0.76	0.81	0.79	27
1	0.84	0.79	0.82	34
accuracy			0.80	61
macro avg	0.80	0.80	0.80	61
weighted avg	0.81	0.80	0.80	61



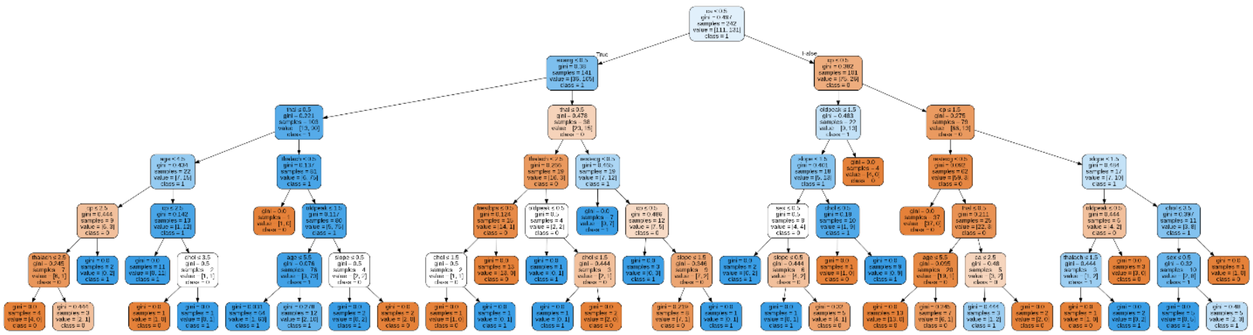
------------------------------ Criterion: gini, Max_Depth: 5, Min_Split: 2 -------------------------------------

	precision	recall	f1-score	support
0	0.59	0.70	0.64	27
1	0.72	0.62	0.67	34
accuracy			0.66	61
macro avg	0.66	0.66	0.66	61
weighted avg	0.67	0.66	0.66	61



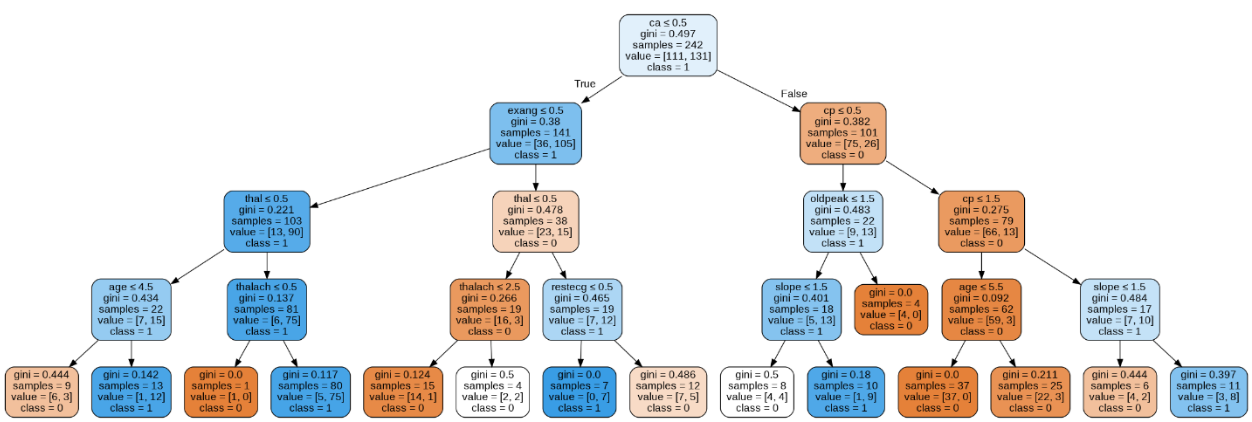
------------------------------ Criterion: gini, Max_Depth: 6, Min_Split: 2 ------------------------------------

	precision	recall	f1-score	support
0	0.64	0.67	0.65	27
1	0.73	0.71	0.72	34
accuracy			0.69	61
macro avg	0.69	0.69	0.69	61
weighted avg	0.69	0.69	0.69	61



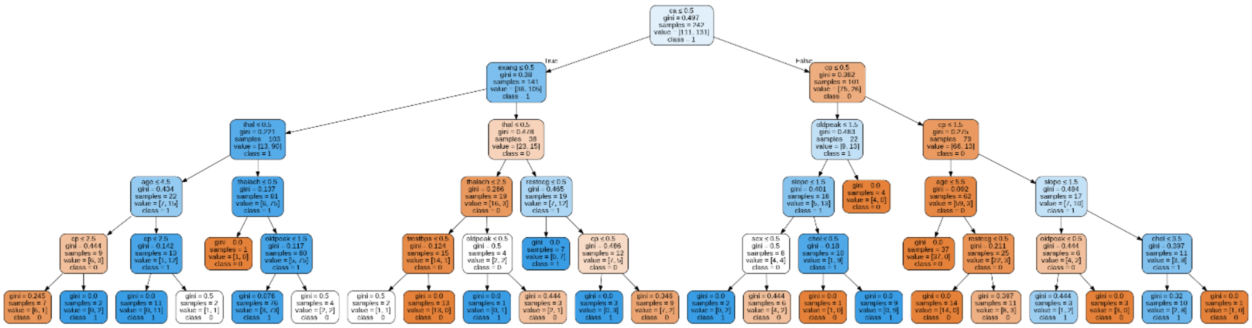
------------------------------ Criterion: gini, Max_Depth: 4, Min_Split: 4 ------------------------------------

	precision	recall	f1-score	support
0	0.76	0.81	0.79	27
1	0.84	0.79	0.82	34
accuracy			0.80	61
macro avg	0.80	0.80	0.80	61
weighted avg	0.81	0.80	0.80	61



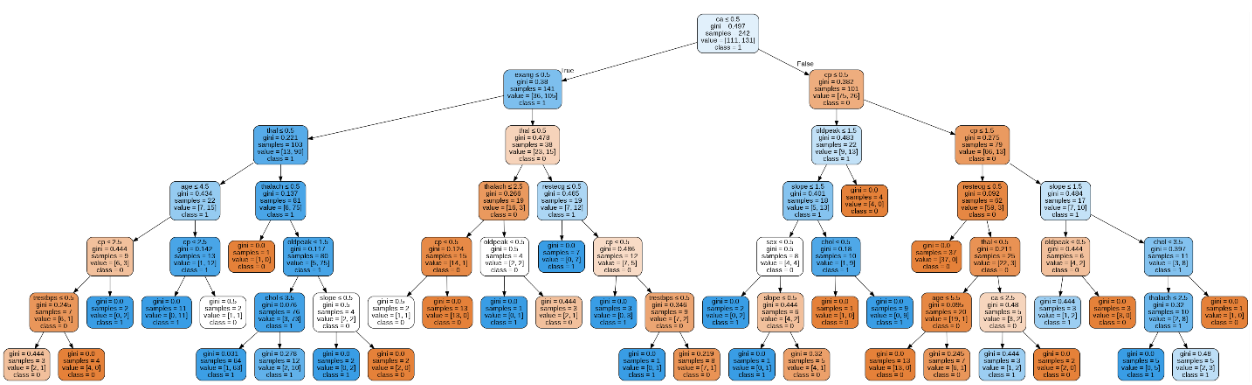
------------------------------ Criterion: gini, Max_Depth: 5, Min_Split: 4 ------------------------------------

	precision	recall	f1-score	support
0	0.59	0.70	0.64	27
1	0.72	0.62	0.67	34
accuracy			0.66	61
macro avg	0.66	0.66	0.66	61
weighted avg	0.67	0.66	0.66	61



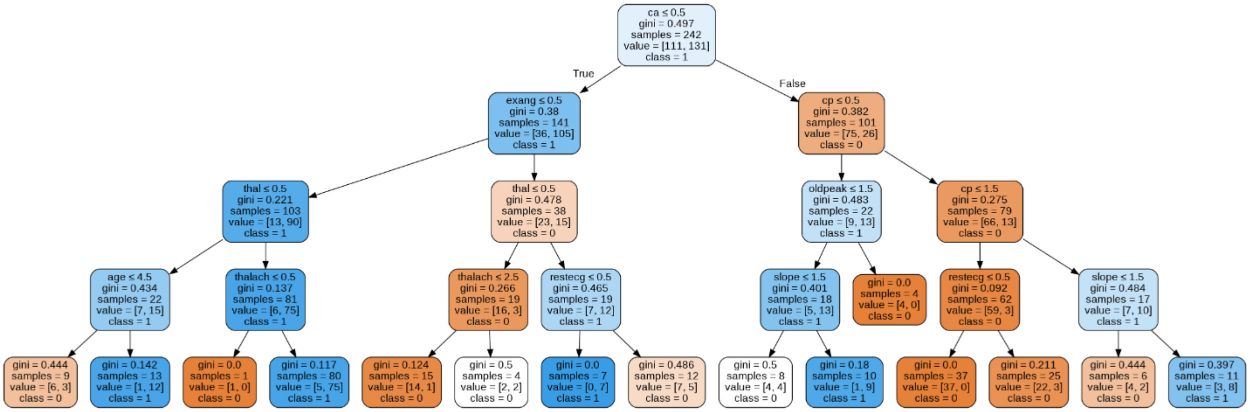
------------------------------ Criterion: gini, Max_Depth: 6, Min_Split: 4 ------------------------------------

	precision	recall	f1-score	support
0	0.60	0.67	0.63	27
1	0.71	0.65	0.68	34
accuracy			0.66	61
macro avg	0.65	0.66	0.65	61
weighted avg	0.66	0.66	0.66	61



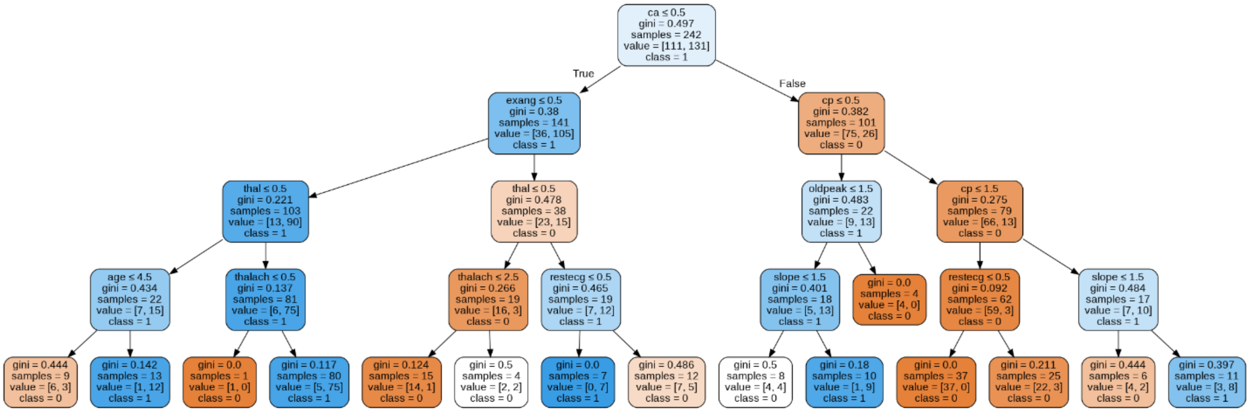
------------------------------ Criterion: gini, Max_Depth: 4, Min_Split: 5 ------------------------------------

	precision	recall	f1-score	support
0	0.76	0.81	0.79	27
1	0.84	0.79	0.82	34
accuracy			0.80	61
macro avg	0.80	0.80	0.80	61
weighted avg	0.81	0.80	0.80	61



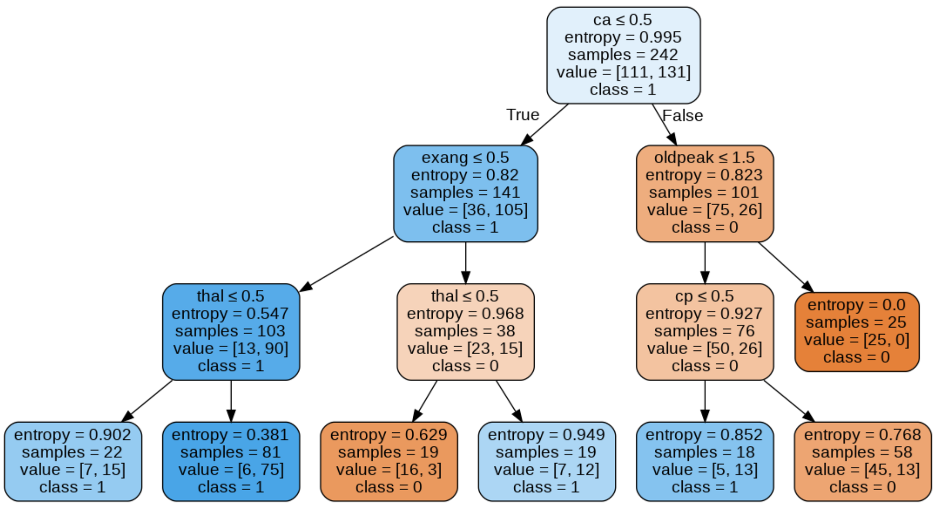
------------------------------ Criterion: gini, Max_Depth: 4, Min_Split: 6 ------------------------------------

	precision	recall	f1-score	support
0	0.76	0.81	0.79	27
1	0.84	0.79	0.82	34
accuracy			0.80	61
macro avg	0.80	0.80	0.80	61
weighted avg	0.81	0.80	0.80	61

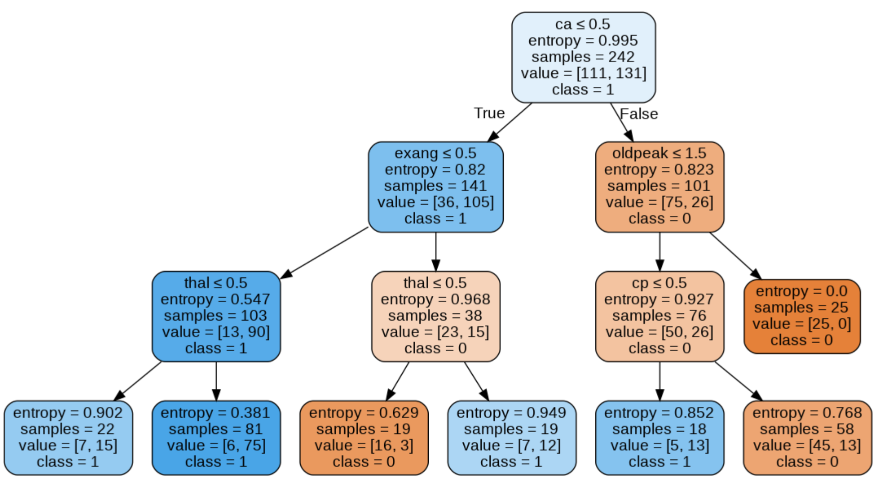


---------------------------- Criterion: entropy, Max_Depth: 3, Min_Split: 4 ----------------------------------

	precision	recall	f1-score	support
0	0.83	0.70	0.76	27
1	0.79	0.88	0.83	34
accuracy			0.80	61
macro avg	0.81	0.79	0.80	61
weighted avg	0.81	0.80	0.80	61



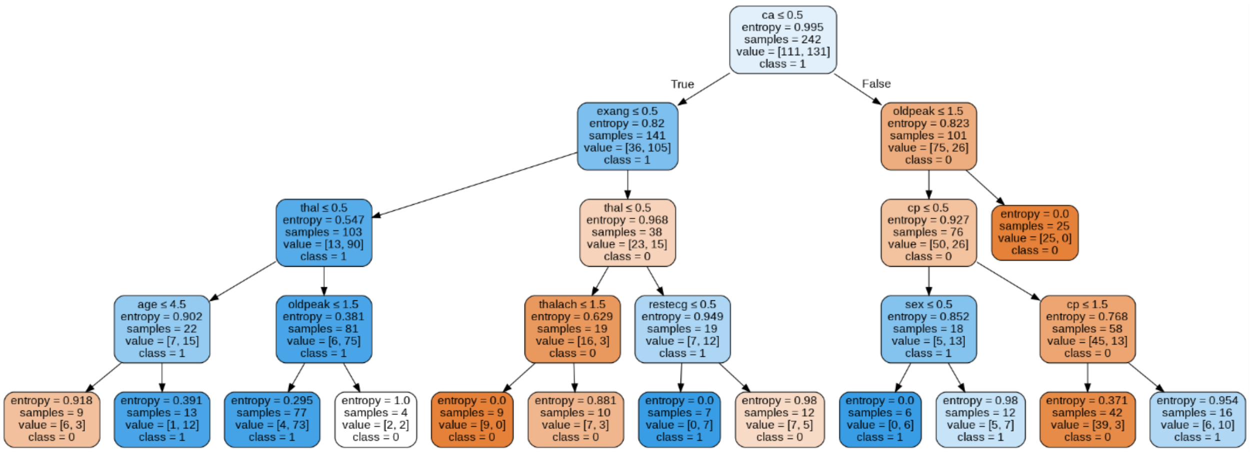
---------------------------- Criterion: entropy, Max_Depth: 3, Min_Split: 5 ----------------------------------



	precision	recall	f1-score	support
0	0.83	0.70	0.76	27
1	0.79	0.88	0.83	34
accuracy			0.80	61
macro avg	0.81	0.79	0.80	61
weighted avg	0.81	0.80	0.80	61

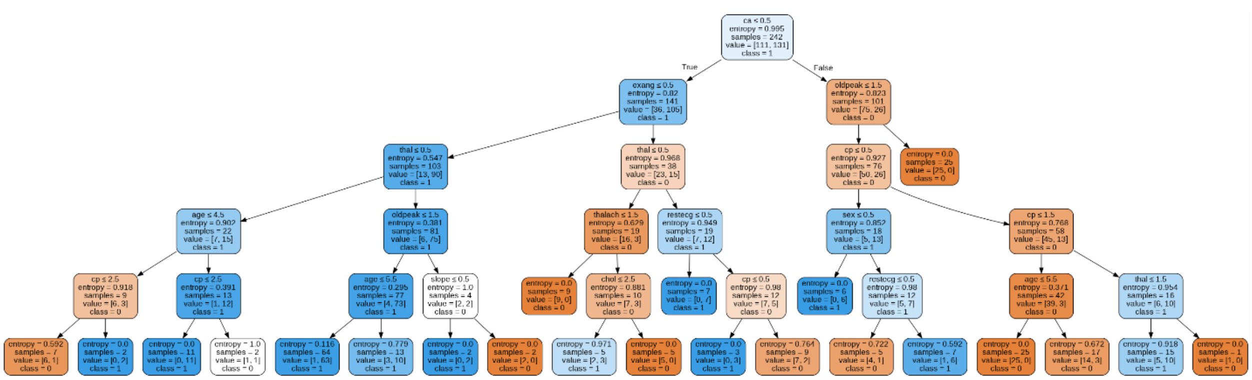
---------------------------- Criterion: entropy, Max_Depth: 4, Min_Split: 4 ----------------------------------

	precision	recall	f1-score	support
0	0.68	0.70	0.69	27
1	0.76	0.74	0.75	34
accuracy			0.72	61
macro avg	0.72	0.72	0.72	61
weighted avg	0.72	0.72	0.72	61



---------------------------- Criterion: entropy, Max_Depth: 4, Min_Split: 4 ----------------------------------

	precision	recall	f1-score	support
0	0.64	0.67	0.65	27
1	0.73	0.71	0.72	34
accuracy			0.69	61
macro avg	0.69	0.69	0.69	61
weighted avg	0.69	0.69	0.69	61



As we can see above, best F1-score obtained is 80 :

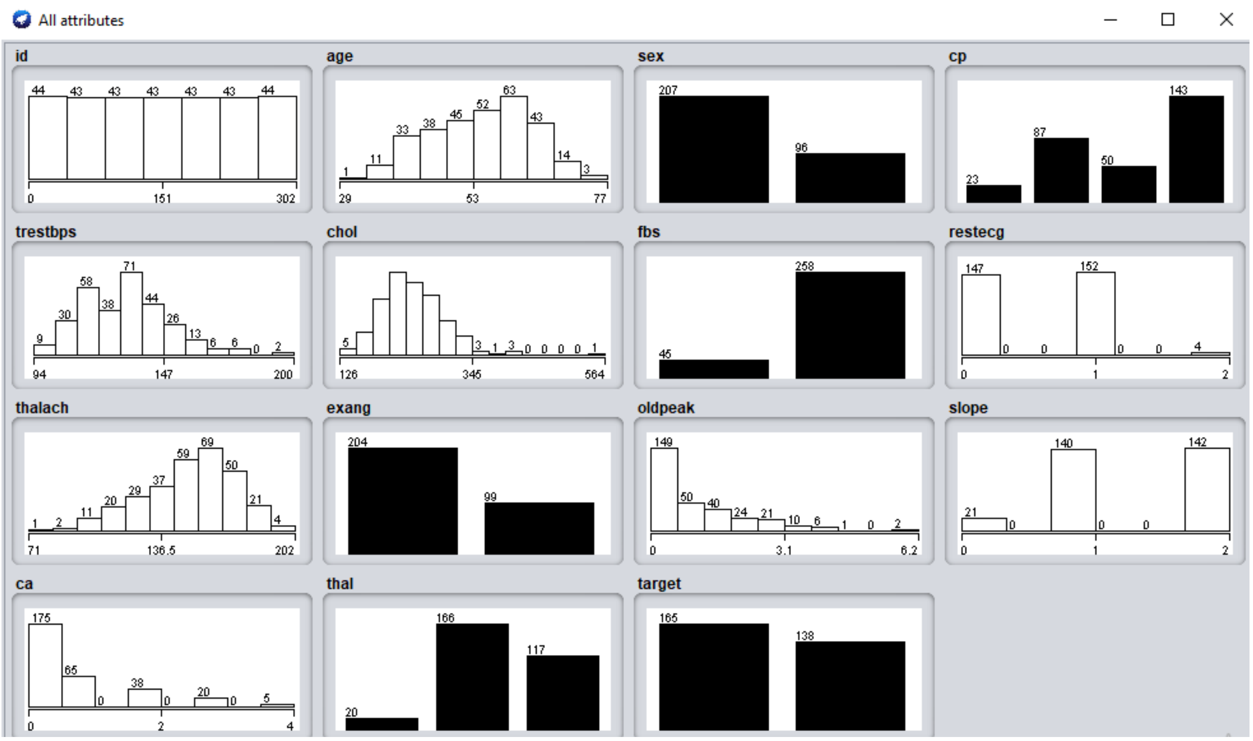
- If we are using gini criterion, then max_depth should be 4 and Min_Split could be 4, 5 or 6. For this parameters F1-score, precision and recall all are about 80.

- If we are using entropy criterion, then max_depth should be 3 and Min_Split could be 4 or 5. For this parameters F1-score, precision and recall all are about 80.

IMPLEMETATION 2: Decision Tree with Weka

In this part, Weka Is used for decision tree classification instead of sklearn library in python.

Firt of all a name was added to the first column of csv file as ‘id’ or also it could be deleted. Then it was converted to arff format with the help of Weka. After changing format of csv to Weka. Distribute of data for each column of csv is as follows:



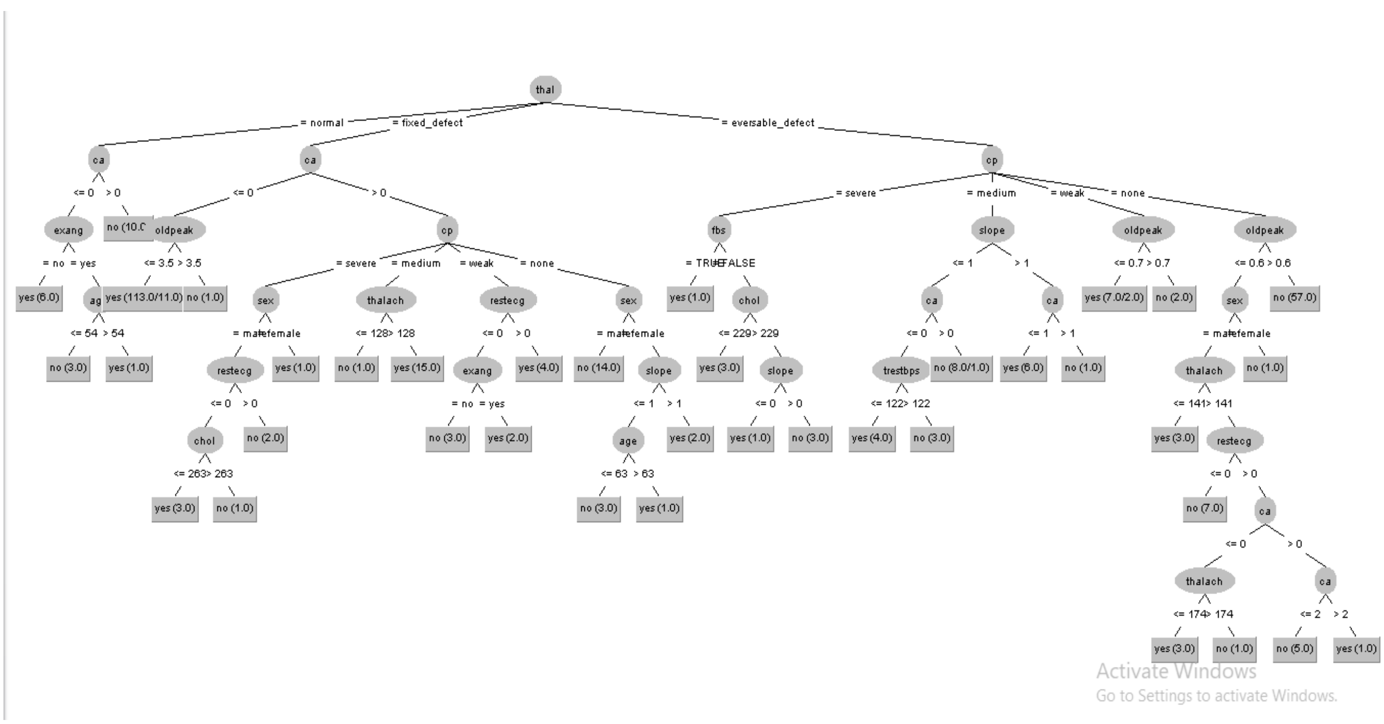
After training the result was:

```
Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         165               54.4554 %
Incorrectly Classified Instances       138               45.5446 %
Kappa statistic                          0
Mean absolute error                      0.4961
Root mean squared error                  0.498
Relative absolute error                100      %
Root relative squared error            100      %
Total Number of Instances              303

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    1.000    0.545      1.000    0.705      ?        0.487     0.537     yes
                 0.000    0.000    ?          0.000    ?          ?        0.487     0.448     no
Weighted Avg.    0.545    0.545    ?          0.545    ?          ?        0.487     0.497

=== Confusion Matrix ===

   a   b   <-- classified as
 165   0 |   a = yes
 138   0 |   b = no
```

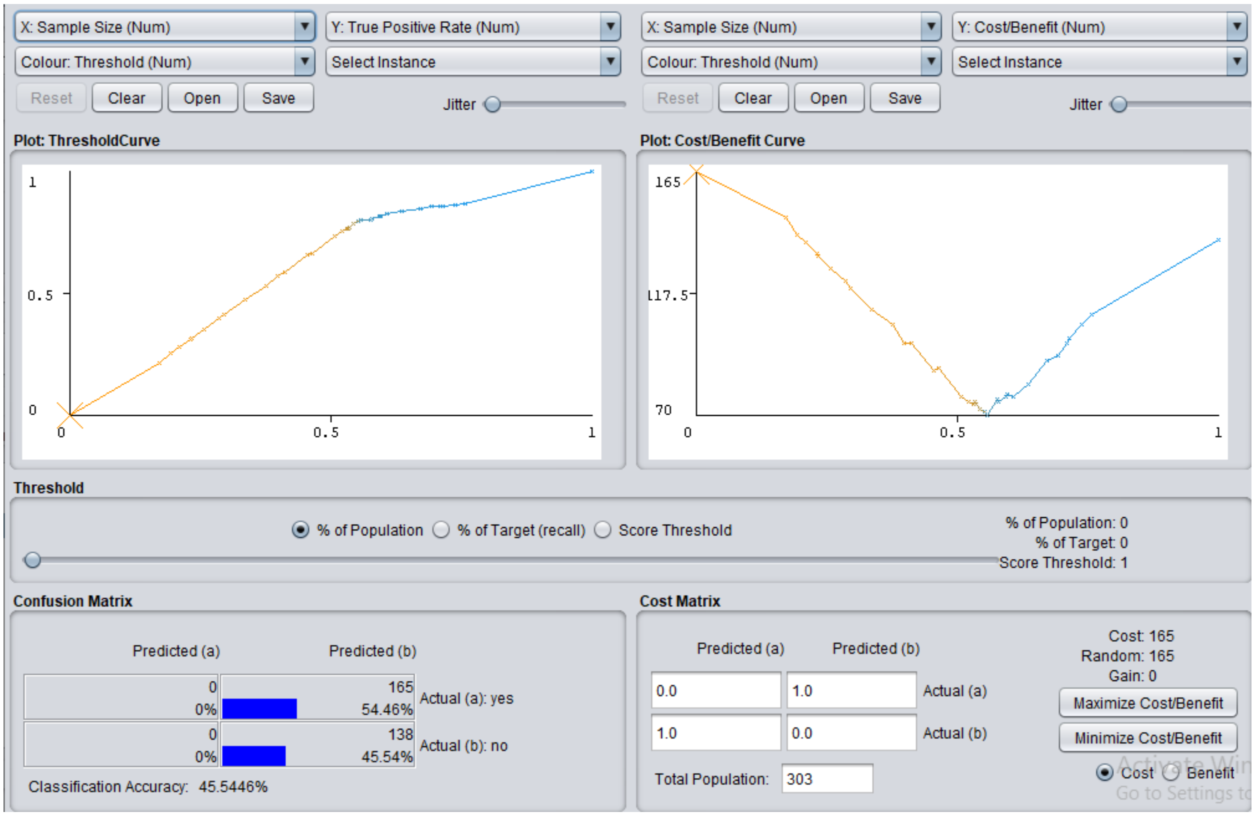
Because the data above was in format ID3 so it could not be visualized. So we changed the format and used C4.5 which is successor of ID3:

```
=== Summary ===

Correctly Classified Instances         231               76.2376 %
Incorrectly Classified Instances        72               23.7624 %
Kappa statistic                          0.5209
Mean absolute error                      0.2705
Root mean squared error                  0.4691
Relative absolute error                 54.522  %
Root relative squared error             94.1805 %
Total Number of Instances              303

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.782    0.261    0.782      0.782    0.782      0.521    0.740     0.710     yes
                 0.739    0.218    0.739      0.739    0.739      0.521    0.740     0.670     no
Weighted Avg.    0.762    0.241    0.762      0.762    0.762      0.521    0.740     0.692

=== Confusion Matrix ===

   a   b   <-- classified as
 129  36 |   a = yes
  36 102 |   b = no
```

And the visualized classification for decision tree is :



Also Cost/Benefit analysis for Data are as visualized as below:



IMPLEMETATION 3: Naïve Bayes Text Classification

In this implementation, we had to classify Amazon Users comments and find out that a comment has a positive or negative Idea.

At first, train data should be preprocessed by changing all words to lower case and eliminating the stop words given as a separate file.

After preprocessing, probability of word to its class was calculated and stored in a dictionary. In test set, after preprocessing as well, Naïve Bayes Classification should have been computed.

Without Laplace smoothing, if a word was not in a class, then the probability would have calculated as zero! This resulted in all zero probability of positive and negative and accuracy was:

```
Accuracy without Laplace Smoothing is: 0.0%
```

Then Laplace Smoothing was use. Formula of Laplace smoothing is as follows:

generalized Laplace estimate:

$$P(A_i = v_j | c_k) = \frac{n_{ijk} + 1}{n_k + s_i}$$

- n_{ijk}: number of examples in c_k where $A_i = v_j$
- n_k: number of examples in c_k
- s_i: number of possible values for A_i

With help of Laplace smoothing and with alpha 1, accuracy from zero raised to:

```
Accuracy with Laplace Smoothing is: 50.0%
```