Submitted by Team 2

Aparna Krishna Bhat | Surbhi Zambad

Data Mining: Project 1

Classification

**Project Overview**

* The Banking dataset consisting of 41,000+ rows was downloaded and imported to R studio as a CSV file for analysis.
* The number of missing or unknown(?) values in the dataset was found to be **12718.**
* The tuples with missing/unknown values(?) were omitted using the **na.omit()** function and a dataset consisting of 30488 records and 21 attributes was obtained.
* Columns marital, default, housing, loan, contact were dropped using **select()** function and a clean dataset consisting of 30488 records and 16 attributes was obtained for the analysis.
* The **seed value was set to 20** as instructed and, about **10000 rows were sampled** and was split into training data and test data.
* The **split ratio used** for the analysis are **(80:20)** i.e. 80% of the sampled data for the training and 20% of the sampled data for testing the model **and (50:50)** i.e. 50% of the sampled data for the training the model and 50% of the sampled data for testing.
* The classification was done using Information gain, GINI index for decision tree and Naïve Bayes on the dataset.

**Libraries Used**

dplyr, rpart, rpart.plot, caret, e1071, naïvebayes, phych, FSelector, ggplot2, GGally, ROCR

**GINI Index**

Using the rpart library, decision tree was built for the training set for different complexity values (cp). Here we did not remove any attributes. Split ratio used here is **(80:20)**

**gini\_tree <- rpart(y ~ .,data = training\_data, parms = list(split = 'gini'),minsplit = 2,minbucket = 1)**

Here the cp value is considered as 0.01 which is the default cp value in rpart library.

After using **printcp(gini\_tree)** we obtained the below mentioned result(s)**:**

Variables used in tree construction:  
cons.price.idx, day\_of\_week, duration, nr.employed, poutcome

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **cp** | **nsplit** | **Rel error** | **xerror** | **xstd** |
| 0.088177 | 0 | 1.00000 | 1.00000 | 0.029324 |
| 0.018227 | 2 | 0.82365 | 0.83941 | 0.027179 |
| 0.013793 | 4 | 0.78719 | 0.84039 | 0.027193 |
| 0.010345 | 6 | 0.75961 | 0.79901 | 0.026593 |
| 0.010000 | 8 | 0.73892 | 0.79113 | 0.026476 |

**gini\_tree <- rpart(y ~ .,data = training\_data, parms = list(split = 'gini'),minsplit = 2,minbucket = 1,cp=0.05)**

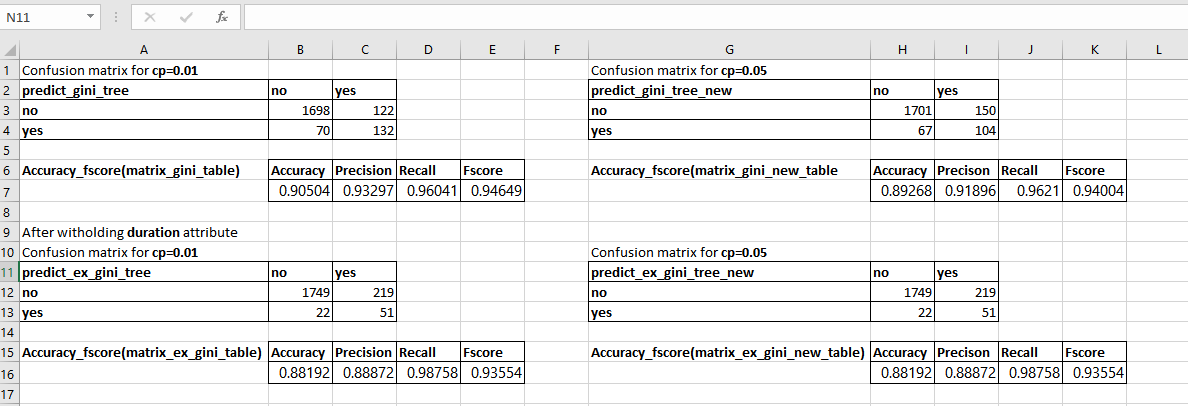
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **cp** | **nsplit** | **rel error** | **xerror** | **xstd** |
| 0.088177 | 0 | 1.00000 | 1.00000 | 0.029324 |
| 0.050000 | 2 | 0.82365 | 0.83941 | 0.027179 |

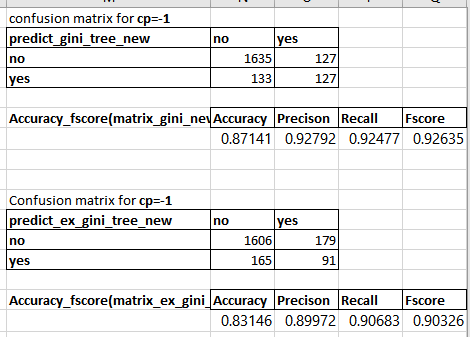
we must **minimize the root-mean-square xerror and ensure a greater confidence interval of xstd. Such a tree is considered the best pruned tree with least overfitting.**

After trying different cp values and running **gini\_tree\_new$variable.importance** we obtained the list of attributes with their corresponding importance value as mentioned below:

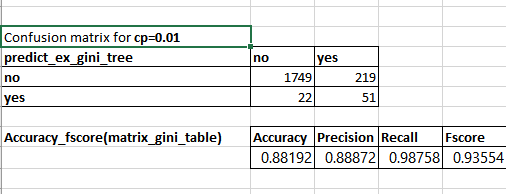
**nr.employed   euribor3m    emp.var.rate cons.conf.idx cons.price.idx**  
# 284.0835068    249.8593475    161.6087296    147.3562408    118.8512630  
 **duration           month       education     day\_of\_week   job**# 113.9821659     77.7847697      0.7385454      0.4923636      0.2461818

The confusion matrix obtained when **duration** attribute withheld with different cp values:





When cp = -1 there a huge difference in the accuracy, precision. recall and fscore.On the other hand, when cp = 0.05 before and after removing the duration attribute there is a minimal difference in accuracy, precision, recall and fscore. Hence, we can analyze that in the best pruned tree, nr.employed, pdays, cons.price.idx are the only attributes used to build the tree. Therefore, it does not make much of difference if we remove duration as an attribute.



Original Tree (80:20) Pruned Tree (80:20)

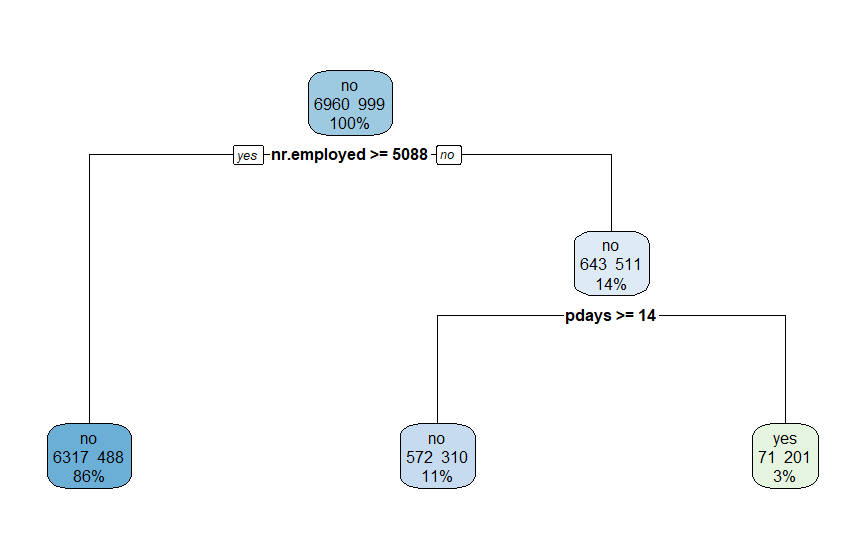
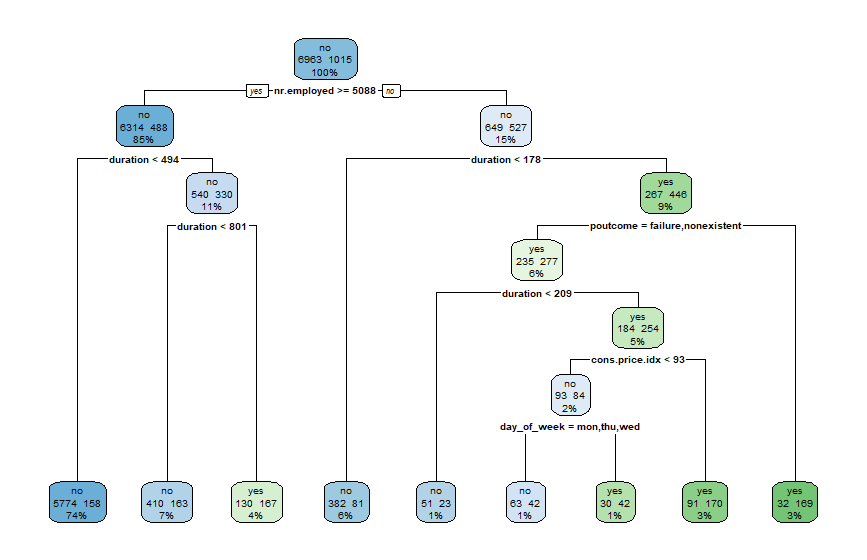
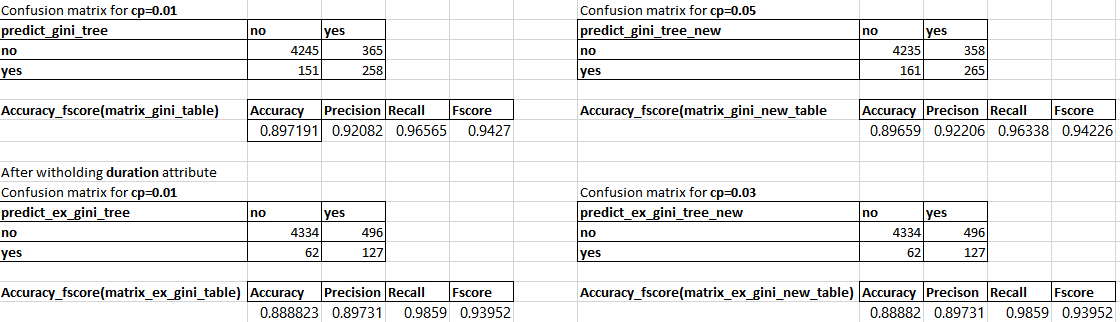
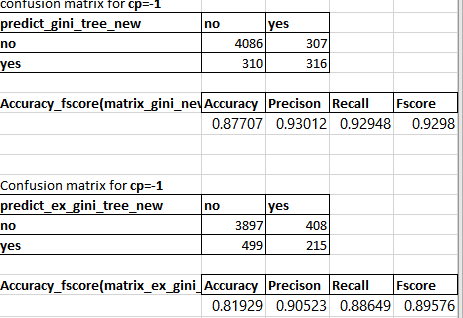
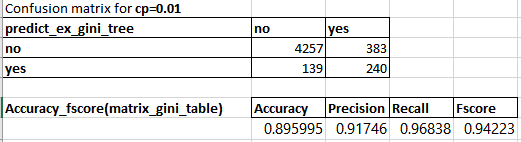


Figure 1 Figure 2

**For the split ratio (50:50)**

We used the same approach as (80:20) and we obtained the below mentioned results:



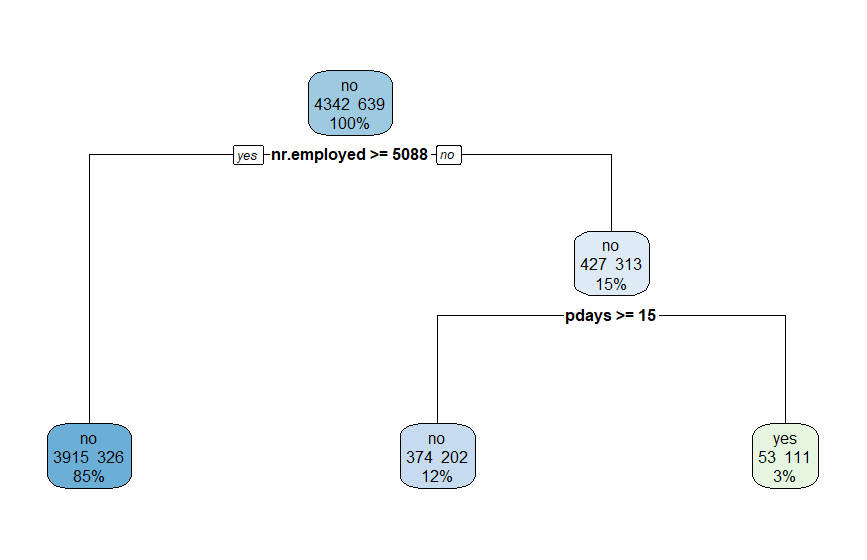
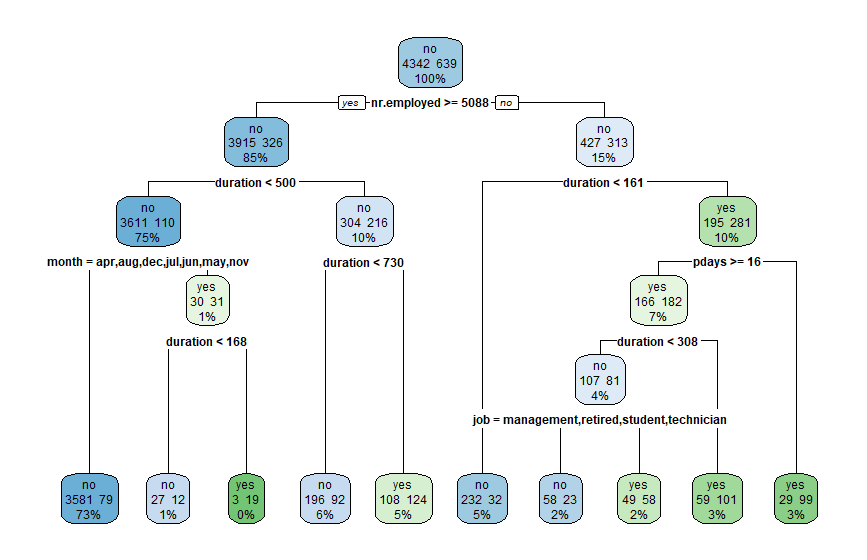
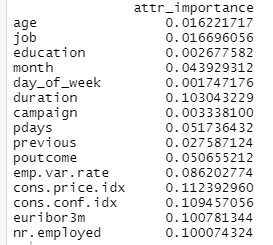
Original Tree (50:50) Pruned Tree (50:50) 

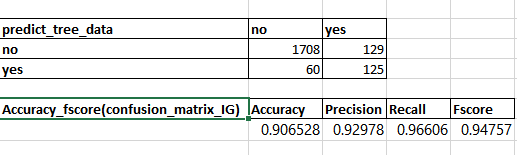
Figure 3 Figure 4

**Information Gain**

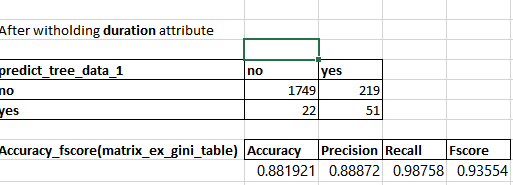
* Using the rpart library, decision tree was built for the training set. By using the **function information.gain**, the library built the decision tree using information.gain. Split ratio used is (80:20). The decision tree helped us in understanding the attributes which are used in split criteria.
* All the attributes got listed in the relative order of their Information gain as stated below using the **info\_gain <- information.gain(y ~ ., data=training\_data, unit="log2")**



* The attributes involved in construction of decision tree were found to be nr.employed,duration,poutcome,euribor3m,cons.price.idx,day\_of\_week
* The predict function was used to predict the test dataset and a confusion matrix was obtained with the predicted values of the test dataset and actual values of the test dataset.



**After dropping the duration attribute**, we obtained the below mentioned results



age, education, job, month, campaign, previous are attributes that are never used in the creation of the decision tree. So, these attributes can also be dropped safely.

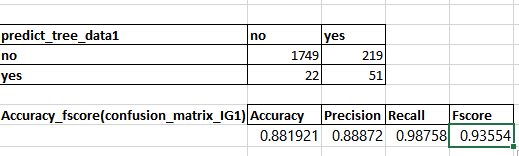
The other attribute that have higher information gain other than the important attribute was found to be pdays.

After using **info\_gain1 <- information.gain(y ~ ., data=train, unit="log2")** we obtained below mention attribute importance

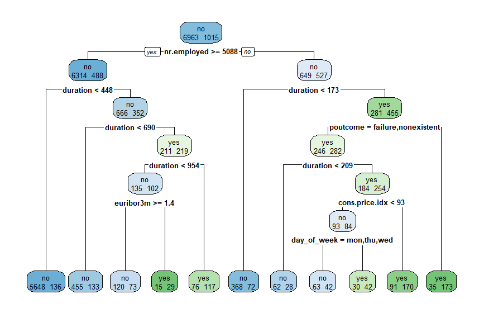
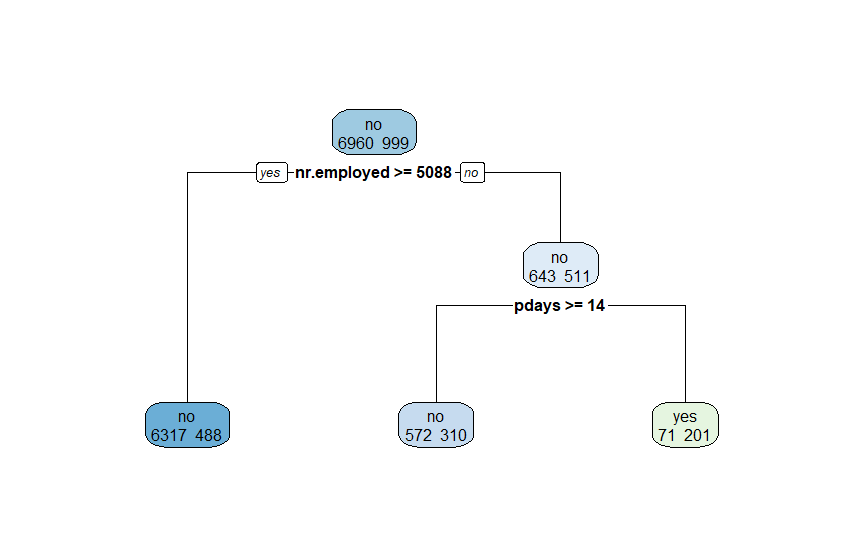
attr\_importance  
**pdays**                0.05318502, **cons.price.idx**      0.10570617, **nr.employed**        0.09630574

Based on this we analyzed the most important attributes for the prediction are nr.employed, pdays and cons.price.idx and concluded dropping the other attributes i.e euribor3m,cons.conf.idx,poutcome will not make any difference to the accuracy, prediction, recall and fscore.

After withholding the most important attributes i.e pdays, cons.price.idx and nr.employed and dropping all the other attributes we obtained the below mentioned results for the pruned tree.

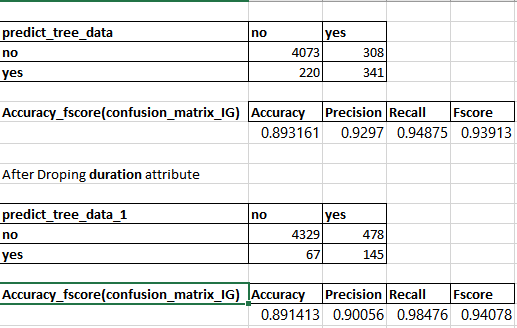


Original (80:20) Pruned (80:20)

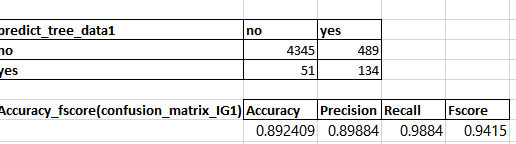
 

**Split ratio (50:50)**

We used the same approach as used for the split ratio of (80:20).Below mention are the results obtained for the original tree and after dropping the duration attribute.

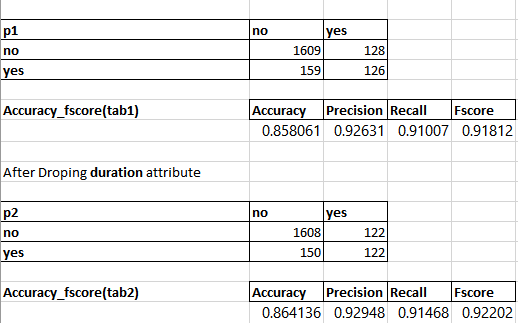


The result(s) of the pruned tree is obtained as below:

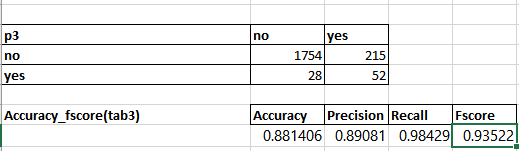


**Naïve Bayes**

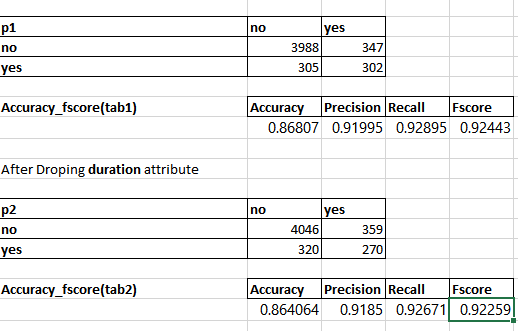
For the split ratio of (80:20) we have obtained the following results for the orignal tree and after dropping the duration attribute.



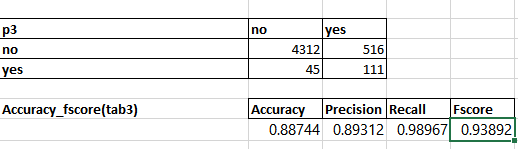
We have obtained the below mentioned results for the pruned tree after witholding the most inportant attributes.



For the split ratio of (50:50) we have obtained the below mentioned results for the original tree and after dropping the duration attribute



We have obtained the below mentioned results for the pruned tree after withholding the most important attributes which are nr.employed, pday and cons.price,idex



**Comparison**

1. **Gini index Vs Information Gain**

From the analysis performed, we can infer that the confusion matrix obtained for both Gini index and Information Gain as given below has not much difference.

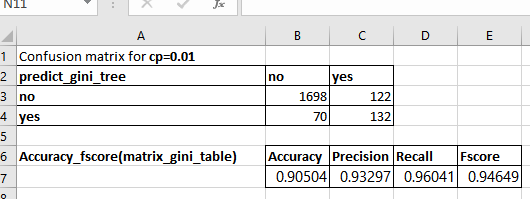
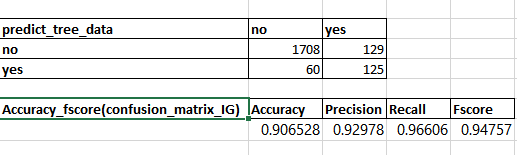
 

Fig:- Gini index Fig:- Information gain

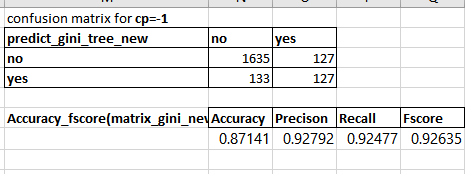
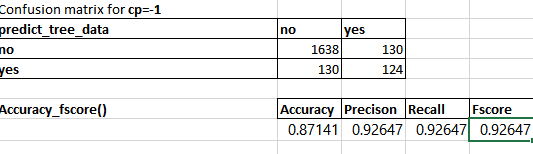
 

Fig:- Gini index cp=-1 Fig:- Information gain cp=-1

From the above results we have analysed that we are not getting any big difference between gini index and information gain considering the cp value as -1.

1. **Gini Index Vs Naïve bayes**

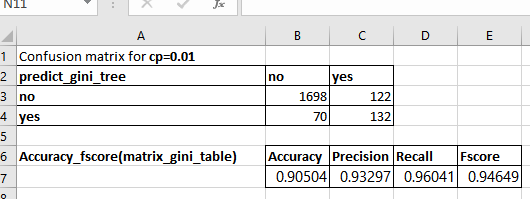
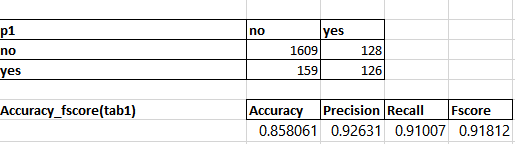
 

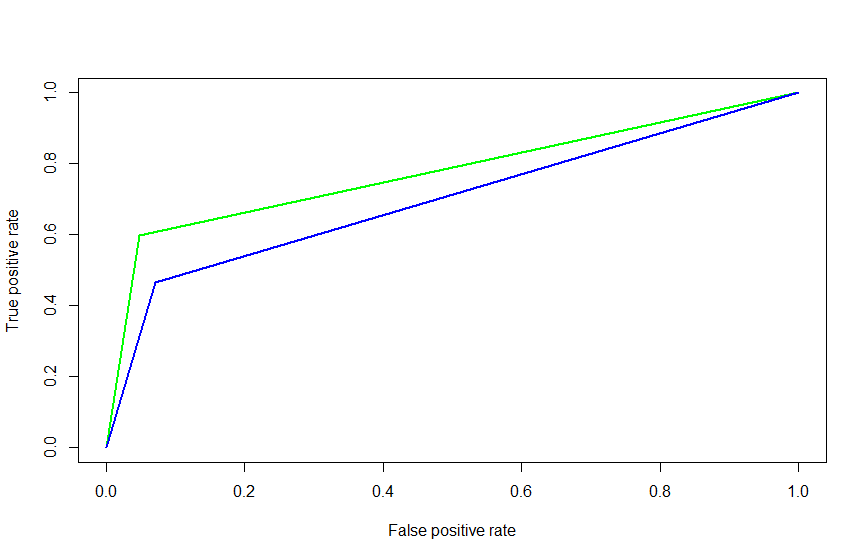
Fig:-Gini Index Fig:- Naïve Bayes

**ROC for Gini index Vs Naïve bayes**

Legends in the graph suggesting

Green:- Gini Index

Blue:- Naïve Bayes



**We can see that the area under ROC curve is more for Gini Index than Naïve Bayes. Hence, Gini Index is a better classification model than Naïve .Essentially decision tree performs better than Naïve Bayes for a larger dataset.**

1. **Information Gain Vs Naïve Bayes**

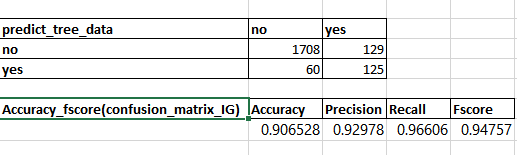
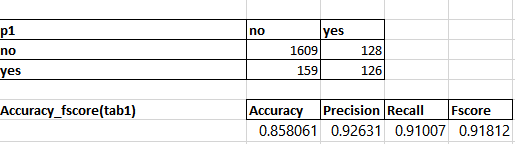
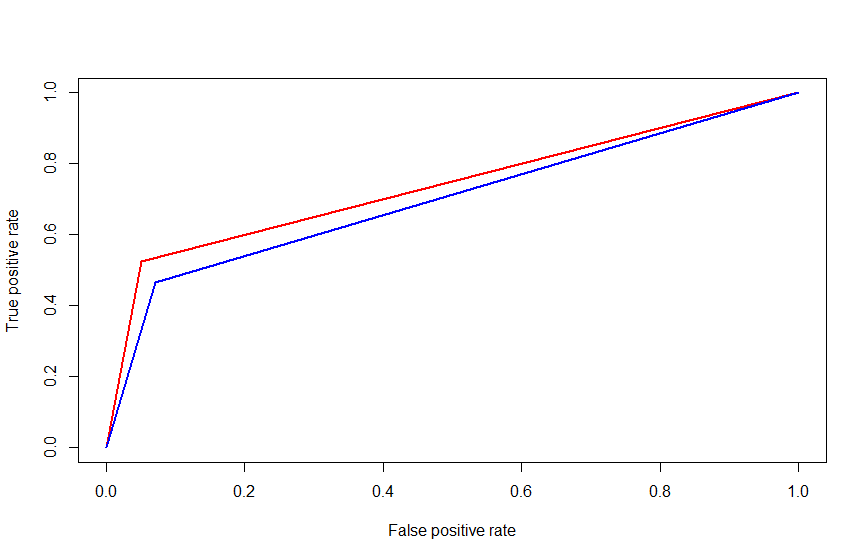
 

Fig:- Information Gain cp = 0.01 Fig:- Naïve Bayes cp = 0.01

**ROC for information Gain Vs Naïve Bayes**



Legends

Red:- Information Gain

Blue:- Naïve Bayes

**We can see that the area under ROC curve is more for Information Gain than Naïve Bayes.Hence, information gain is a better classification model than Naïve Bayes. Also, we can see that, Accuracy (Info\_gain)>accuracy (Naïve bayes) (precision, fscore,recall)[info\_gain] > (precision, fscore,recall)[Naïve\_Bayes]**

1. After the entire analysis we have come to know that initially we identified pdays attribute as non-contributing in our milestone analysis which was based on the visual assumption of the histogram analysis and which was wrong a analysis. As well the reason(s) we identified for the attributes to be noncontributing were also flawed as information gain/ Gini index, cp values were not taken into consideration.

One more thing that we noticed was that majority of the main attributes used for the pruned tree belong to social and economic context attributes and all the client bank data as non-contributing.

1. **Training/ test data for 80:20 split and 50:50 split was found as follows:**

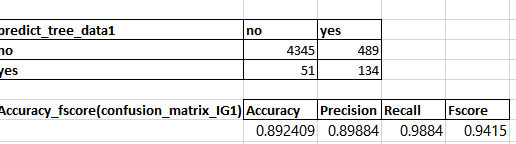
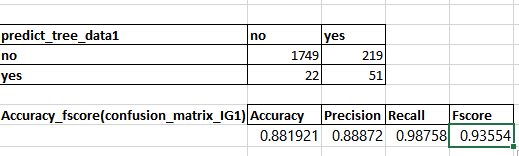
 

Fig: 50:50 split Fig:80:20 split

From above splitting of 80:20 and 50:50 we can see that 50:50 split is giving more accuracy than the 80:20 split.

**Challenges**

* To compute decision trees for different cp values and to understand which attribute to remove, to draw inference and to arrive at the conclusion on which cp value to choose
* To derive insights from confusion matrix and understand comparisons between various classification models i.e Gini index, Information gain and Naïve Bayes
* Understanding and learning about various libraries and packages used as well knowing and working with R and R studio. Understanding where to use which libraries and how to derive insights out of them. Analyzing each attribute to build better prediction model.

**Division of Labor:**

1. Surbhi: Information gain, Naïve Bayes, GINI index for split ratio 80:20 and wrote report for that accordingly.
2. Aparna: Information gain, Naïve Bayes, GINI index for split ratio 50:50 and wrote report for that accordingly.