Group assignment – Data Mining

Data Mining

Post-graduation in Business Analytics and Business Intelligence

Group -6

Contents

[1. Objective 2](#_Toc33980902)

[2. Assumptions 2](#_Toc33980903)

[3. Thera Bank - Loan Purchase Modelling 2](#_Toc33980904)

[3.1 EDA - Basic data summary, Univariate, Bivariate analysis, graphs 3](#_Toc33980906)

[3.2 Applying Supervised Machine Learning Techniques (Test & Train) 11](#_Toc33980907)

[3.3 Applying CART – Full tree 12](#_Toc33980908)

[3.4 Interpret the CART model output (pruning, remarks on pruning, plot the pruned tree) 13](#_Toc33980909)

[3.5 Applying Random Forests (plot the tree) 19](#_Toc33980910)

[3.6 Interpret the RF model output (with remarks, making it meaningful for everybody) 21](#_Toc33980911)

[4. Various Model Performance Measures (Test & Train): Confusion Matrix 26](#_Toc33980912)

[5. Remarks on Model validation exercise (Which model performed the best) 34](#_Toc33980913)

[6. RemarksBuild the Model using the algorithm selected on the last step and interpret the results) 34](#_Toc33980914)

# Objective

For the given problems we mustdo the EDA, find FA/PCA and come up with best fit models.

# Assumptions

The Given data is a sample of the population.

# Thera Bank - Loan Purchase Modelling

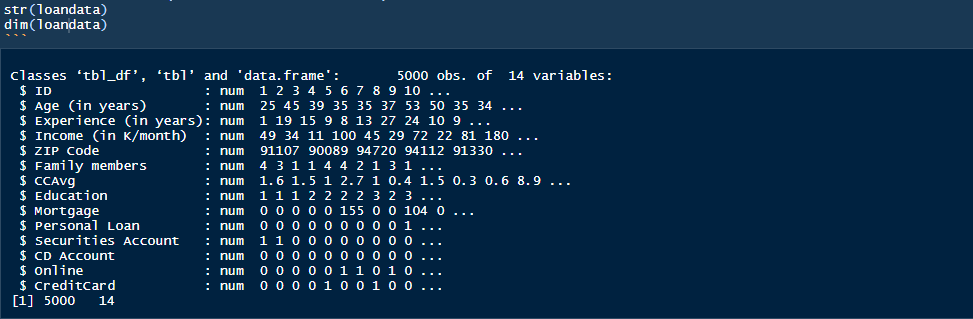
## 

## This case is about a bank (Thera Bank) which has a growing customer base. Majority of these customers are liability customers (depositors) with varying size of deposits. The number of customers who are also borrowers (asset customers) is quite small, and the bank is interested in expanding this base rapidly to bring in more loan business and in the process, earn more through the interest on loans. In particular, the management wants to explore ways of converting its liability customers to personal loan customers (while retaining them as depositors). A campaign that the bank ran last year for liability customers showed a healthy conversion rate of over 9% success. This has encouraged the retail marketing department to devise campaigns with better target marketing to increase the success ratio with a minimal budget. The department wants to build a model that will help them identify the potential customers who have a higher probability of purchasing the loan. This will increase the success ratio while at the same time reduce the cost of the campaign. The dataset has data on 5000 customers. The data include customer demographic information (age, income, etc.), the customer’s relationship with the bank (mortgage, securities account, etc.), and the customer response to the last personal loan campaign (Personal Loan). Among these 5000 customers, only 480 (= 9.6%) accepted the personal loan that was offered to them in the earlier campaign.

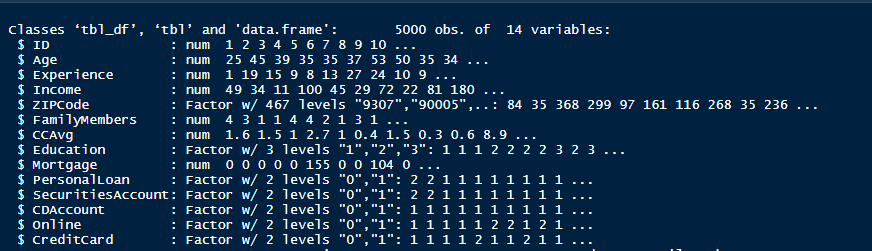


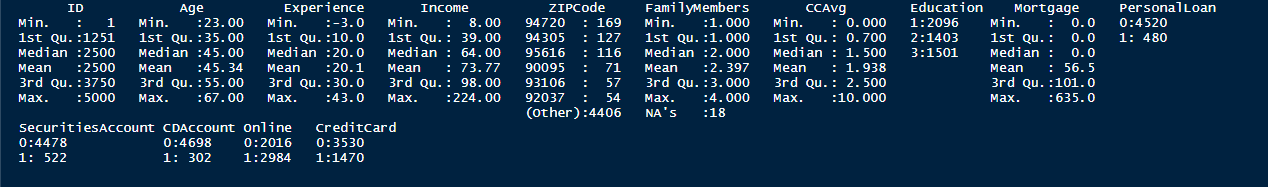
## 3.1 EDA - Basic data summary, Univariate, Bivariate analysis, graphs

**Basic summary of Data**

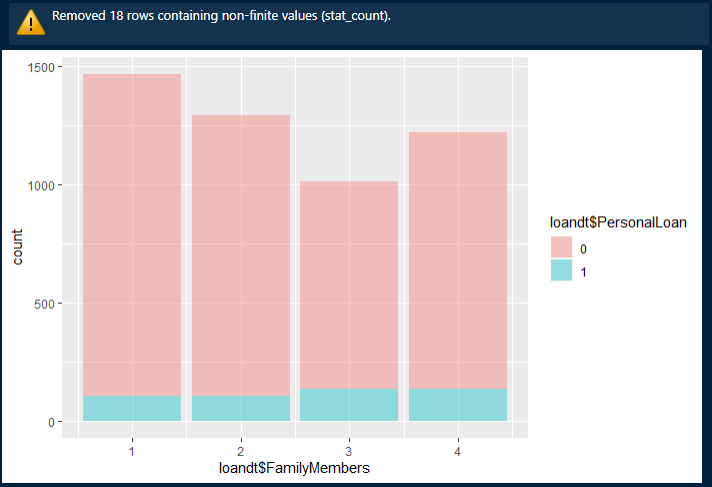


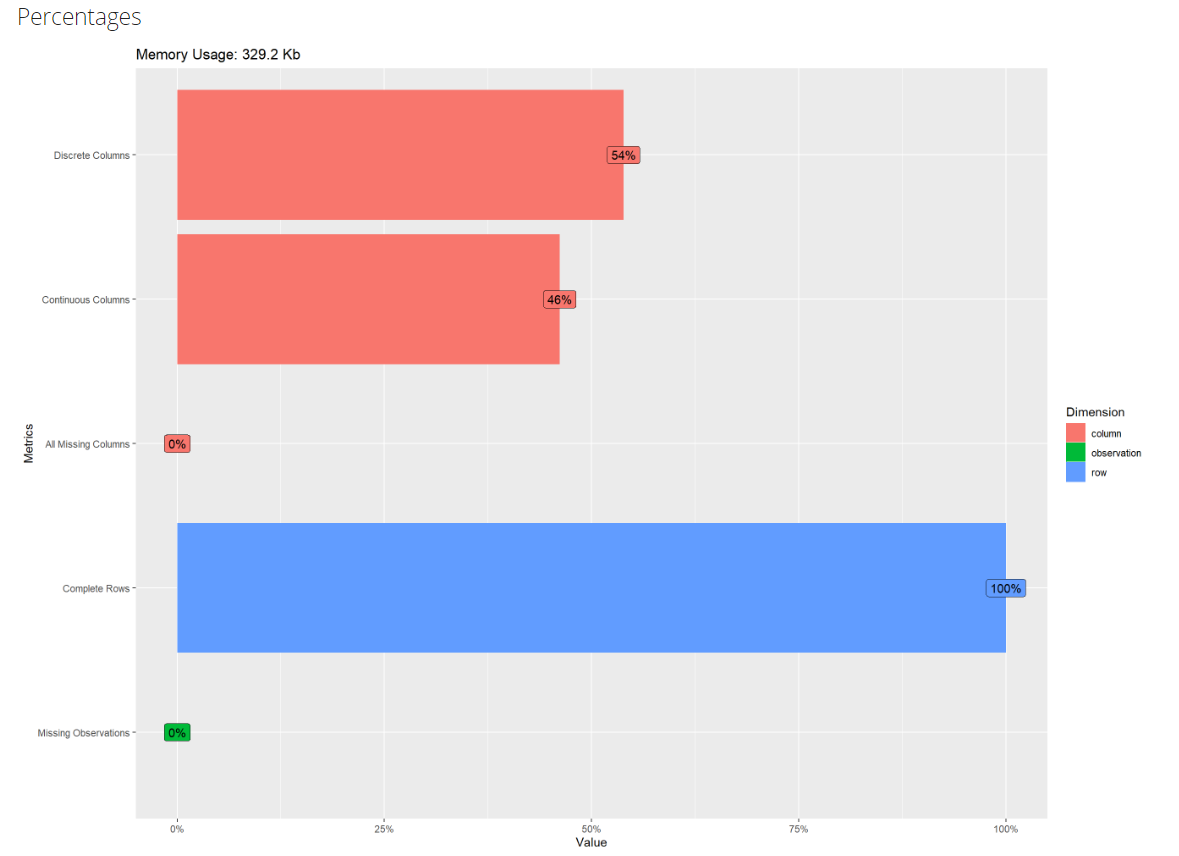
Structure and Summary after converting Zip Code , Education, Personal loan, Securities Account, CD Account, Online and CreditCard to factor

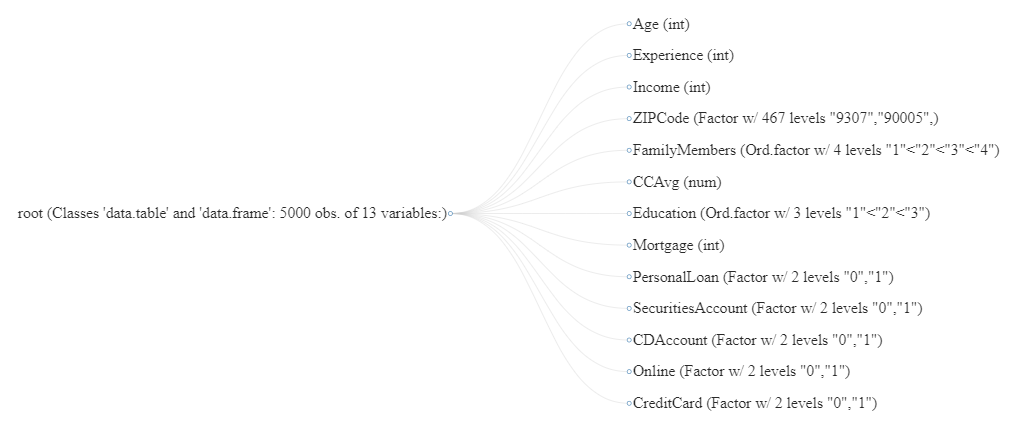




1. Our field of interest is PersonalLoan field and last year only 480 (= 9.6%) accepted the personal loan that was offered to them in the earlier campaign so there is data imbalance in our dataset.
2. 75% of people that were targeted were below Age 55 and have below 30 Years of professional experience.
3. It is a good mix of Undergrad, Graduate and Advanced/Professional. Converted Education into ordered factors.
4. We do not need column ID and Zip Code so will remove it from dataset, experience columns has negative values so we will take abs values assuming it was data entry error.
5. We have used Feature engineering for family member variable.
   1. We have replaced 18 null values to 1 for cart model analysis
   2. We have used linear regression model to predict the missing family data for 18 rows and used the same for Random forest model analysis.

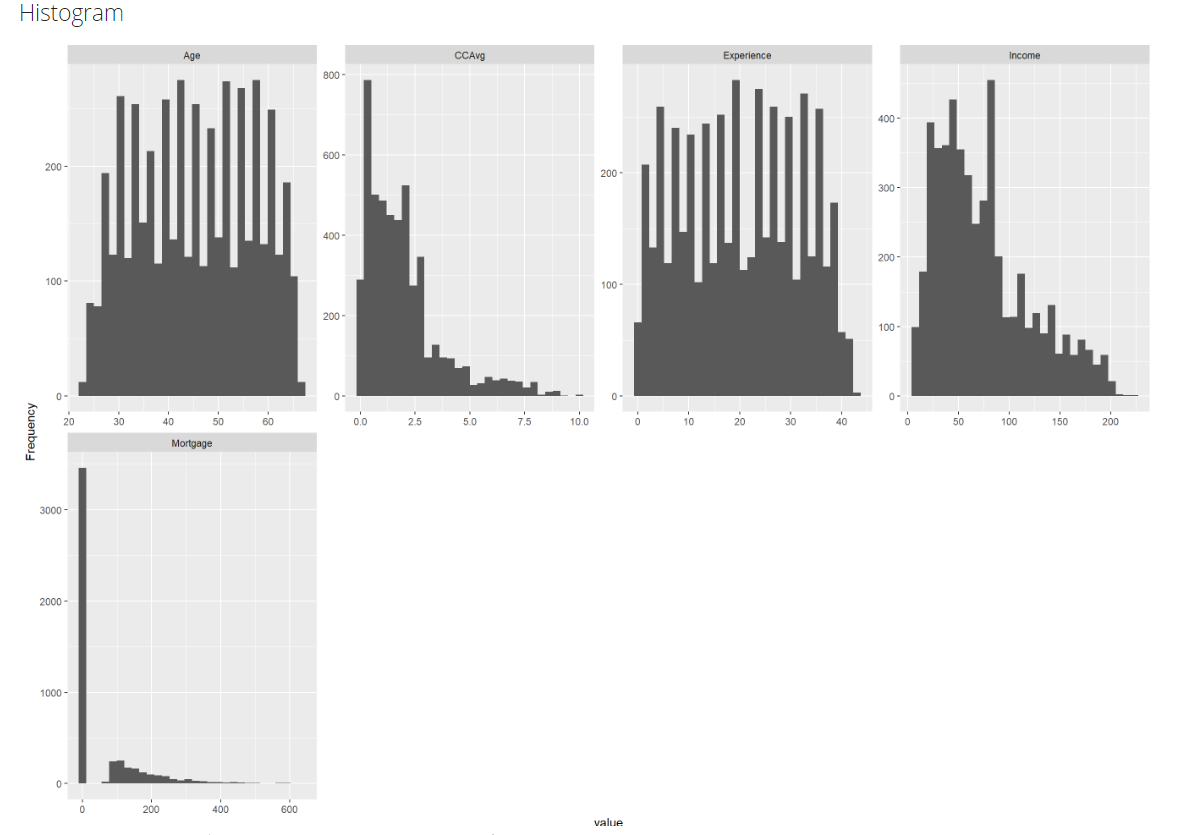


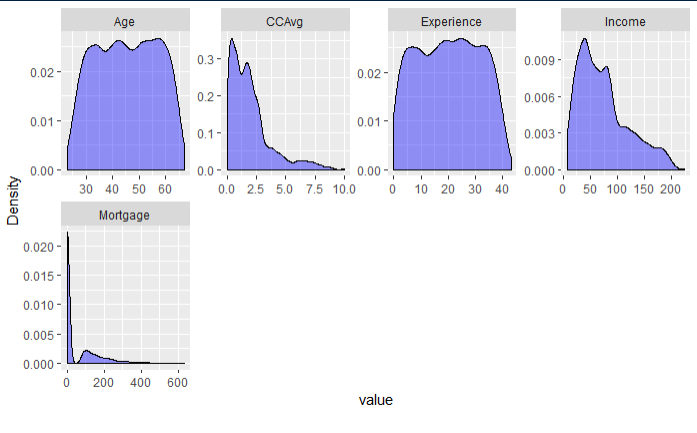
**Basic Data Profile after cleansing and imputing missing values**

****

**UnivariateAnalysis**

Plotting Histogram and Density for all numerical x variables.

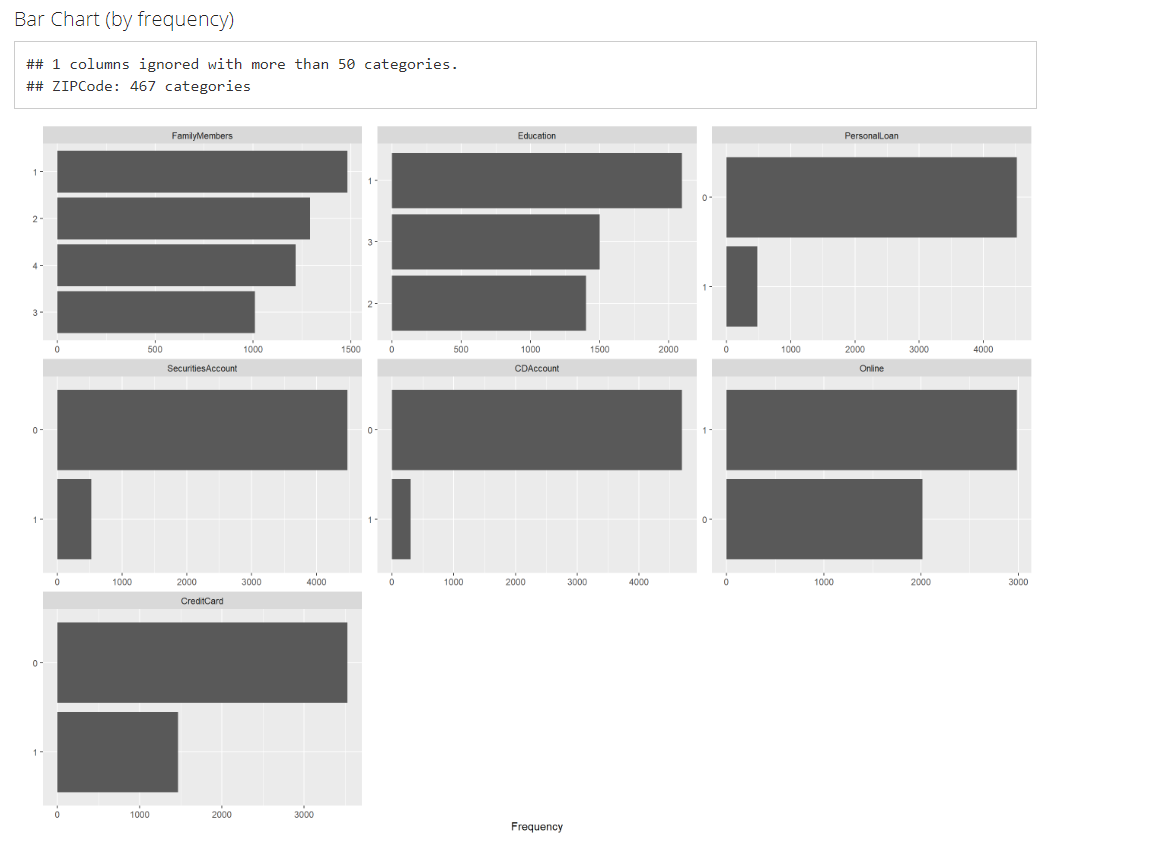
****

****

Observation

* Data is normally distributed by Age and Experience. Data is left skewed for Credit card average spend, Income and Mortgage.
* Majority of customers spend under 5K per month using credit card.
* Most of the customers age fall in the age range of 30 to 60 yrs and their experience fall in the range of 5 to 35 years and most earn an income between 10K to 100K.

Bar Chart (by frequency) for Categorical Variables



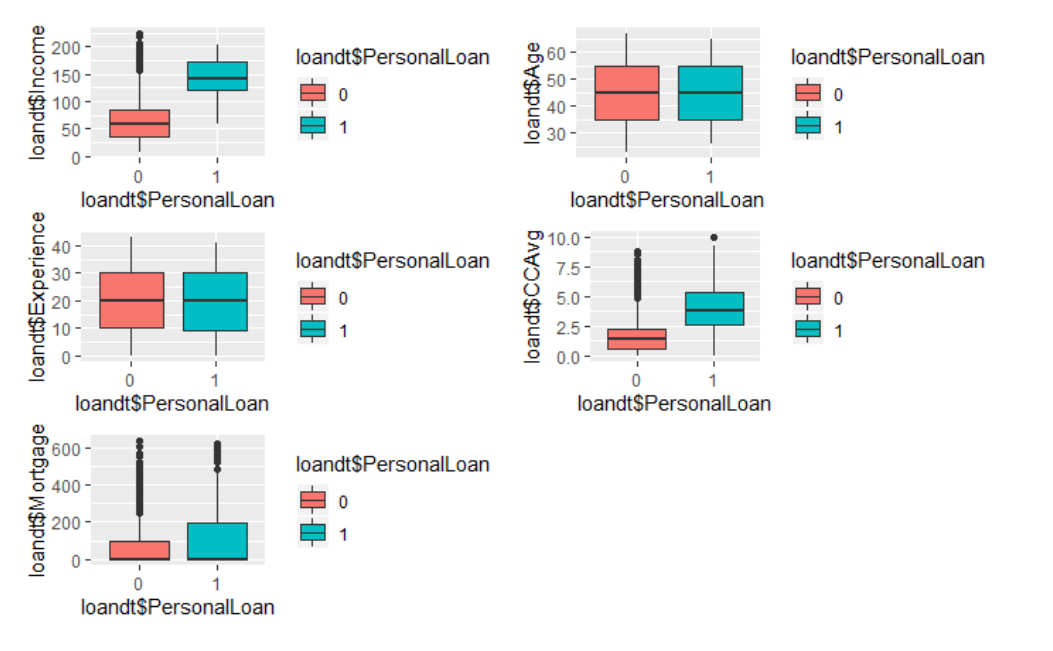
Observation

* Data is skewed when we look at the fieldsCD Account, Securities Account and credit card issued by the bank.
* Good distribution across Family members, Education and Online banking facilities.
* The binary categories have 5 variables as below
* Personal Loan:-Did the customer accept the personal loan offered in the last campaign?(Yes/No)This is our target variable
* Securities Account-Does the customer have a securities account with the bank?(Yes/No)
* CD Account- Does the customer have a certificate of deposit account with in the bank(Yes/No)
* Online-Does the customer using internet banking facilities?(Yes/No)
* Credit Card-Does the customer use a credit card issued by Thera Bank?

(Yes/No)

**Bivariate Analysis**

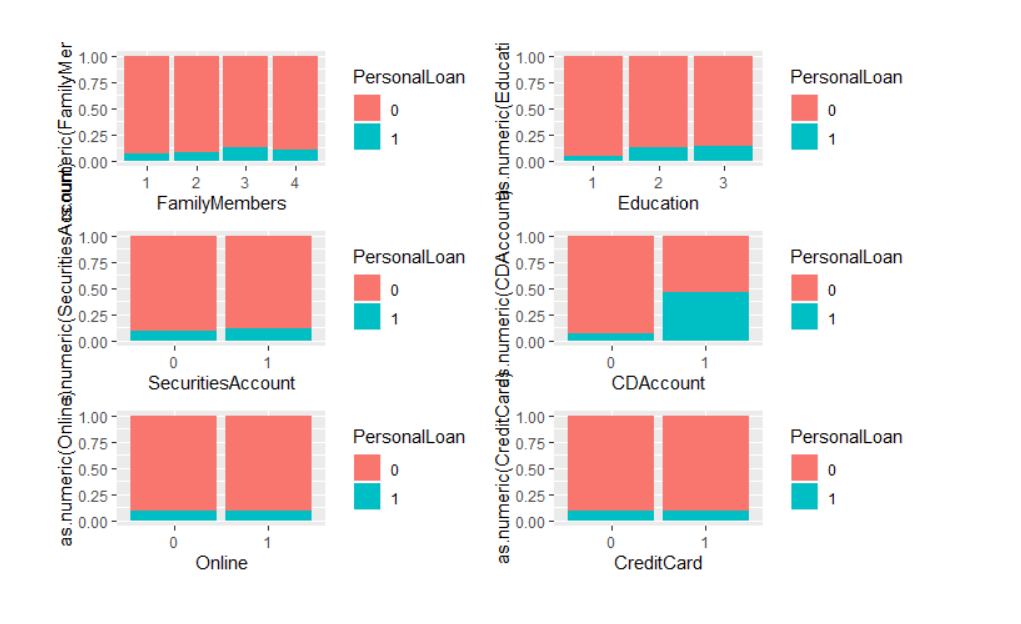
Box plotting Personal Loan against numerical variables.

****

Observation

* Credit card monthly spend, Income and Mortgage seems to be directly affecting personal loan decision.
* Lots of non personal loan (Class 0) takers are present as outliers in Credit Cardspend, Mortgage and Income. These could be a very good Target Class.
* Age and Experience have no bearing on the personal loan acceptance.

Stacked Bar plot of Personal Loan for categorical variables.



Observation

* Only CD Account holders have some effect on personal loan acceptance.
* Data is mostly skewed towards non personal loan takers and other categorical variables are not impacting a lot.
* Education class Advanced/Professional has slightly better conversion rate in terms of acceptance.
* Family sizes of 3 are highest acceptors of loan offer.

Creating scatter plot for numerical variables impacting field of interest



Observation

* Customers with higher income have more mortgages and credit card spend.
* Acceptance is higher for customers with more income.
* People having income between 20 K to 90K have no Personal loans and moderate Credit Card spending 2.5K.
* Customers with high credit card spend and income has good potential of conversion and can be targeted.
* Customers with no mortgages have accepted personal loans more; they can be a good segment of customers for targeting.

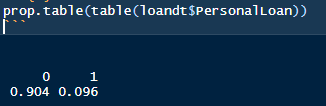
## 3.2 Applying Supervised Machine Learning Techniques (Test & Train)

Positive Class is Personal loan Accepted – “1”, which is ~10% of the dataset so there is significant class imbalance.

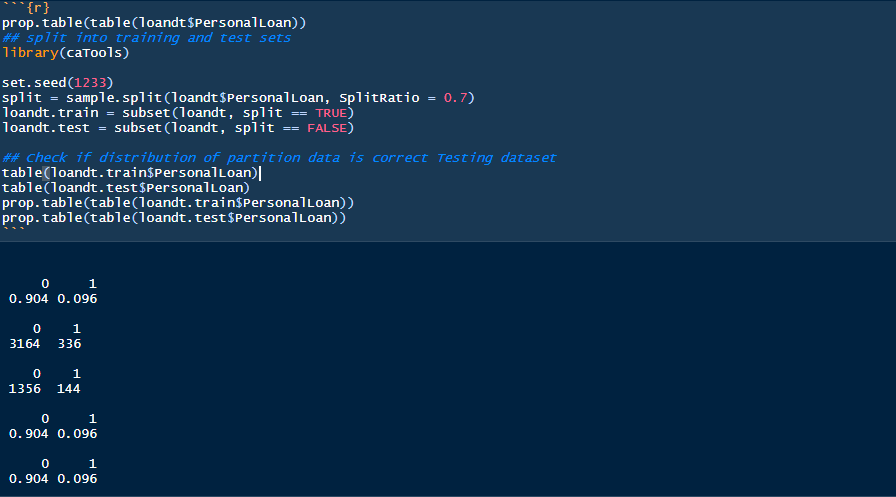
FP – Actually will not accept loan offer but predicted as they will accept

FN – Actually will be taking the offer but predicted as they will not accept the offer.

For our use case FN will be very costly because we will be losing out on potential customers but FP is something that we can live with as it will only lead to additional promotion cost.Ideally we should not even have a single False Negative scenario.

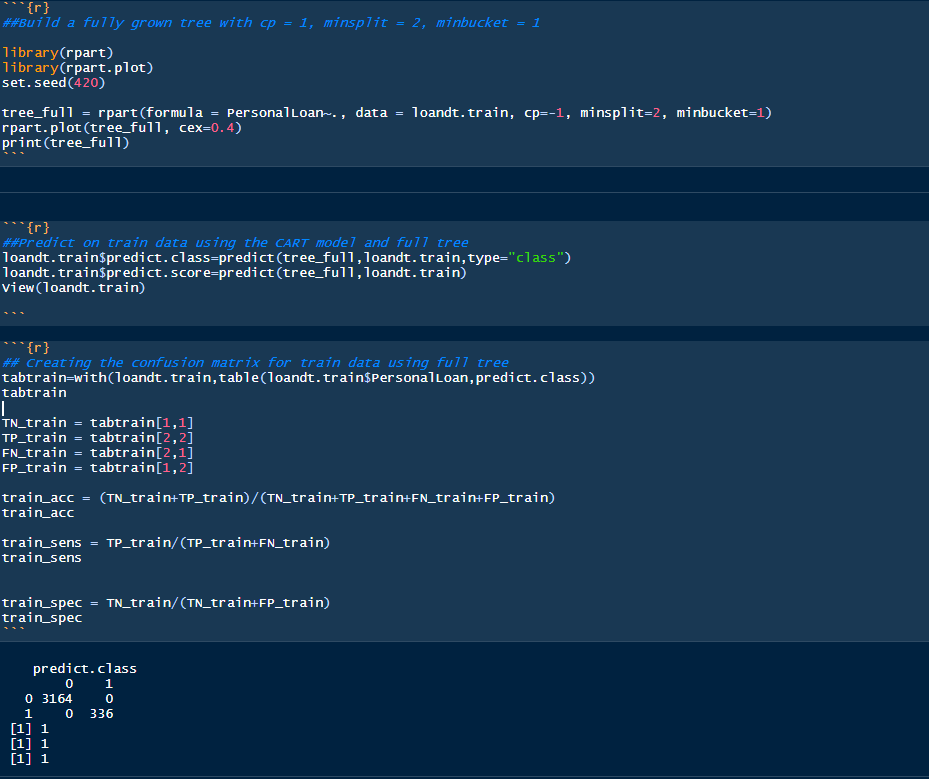


Splitting the data into train and test based on 70/30 ratio fixing the percentage of personal loan same as initial data.



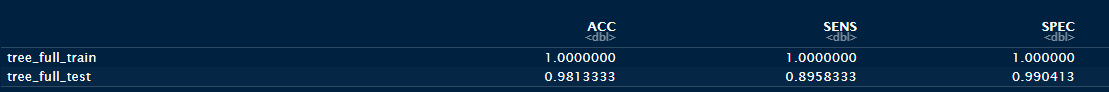
## 3.3 Applying CART – Full tree

Running a cart with full tree for prediction and creating the confusion matrix on train data. As we see below, the model is over fitting and there are no false positives or false negatives.



Once we run the model on test data and consolidate the results, we observe that model overfits especially in terms of Sensitivity which deals with false negative and is very costly in our use case.

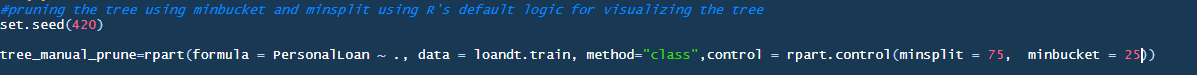




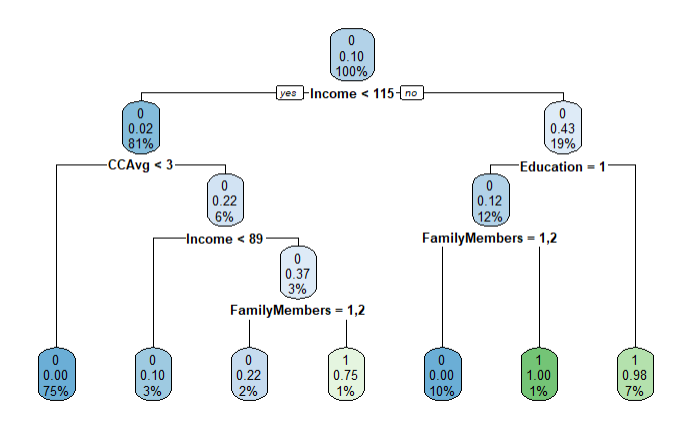
## 3.4 Interpret the CART model output (pruning, remarks on pruning, plot the pruned tree)

Classification trees use recursive partitioning algorithms to learn and grow on data, pruning will be done to limit the recursive based on criteria.

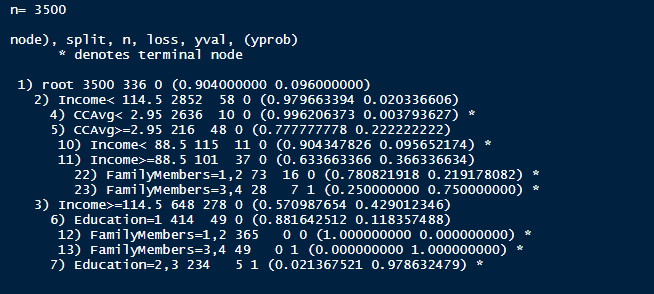
We will first do manual pruning using R’s default logic of minsplit as 1-3% of train data and minbucket as ‘minsplit/3’.



**Plotting the pruned tree**



**Printing the pruned tree**

****Observation

* Income, CC Avg, Family Members and Education are important predictors on which data is split by pruned tree algorithm.
* First split happens on whether Income is less than or greater than $ 114K.
* Second split happens on whether the monthly credit card is less than or greater than 2.95K.
* Also using the manual prune tree we observe thatSensitivity of the model has actually decreased to ~83%.

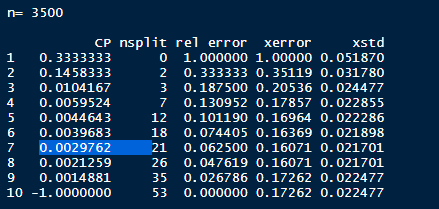
Confusion Matrix for Manual Prune Tree

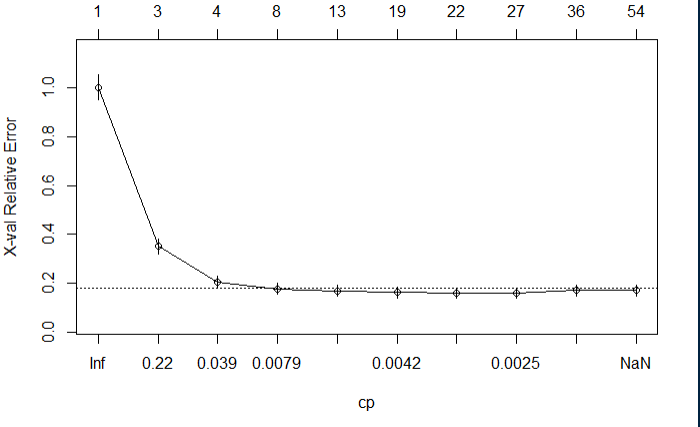




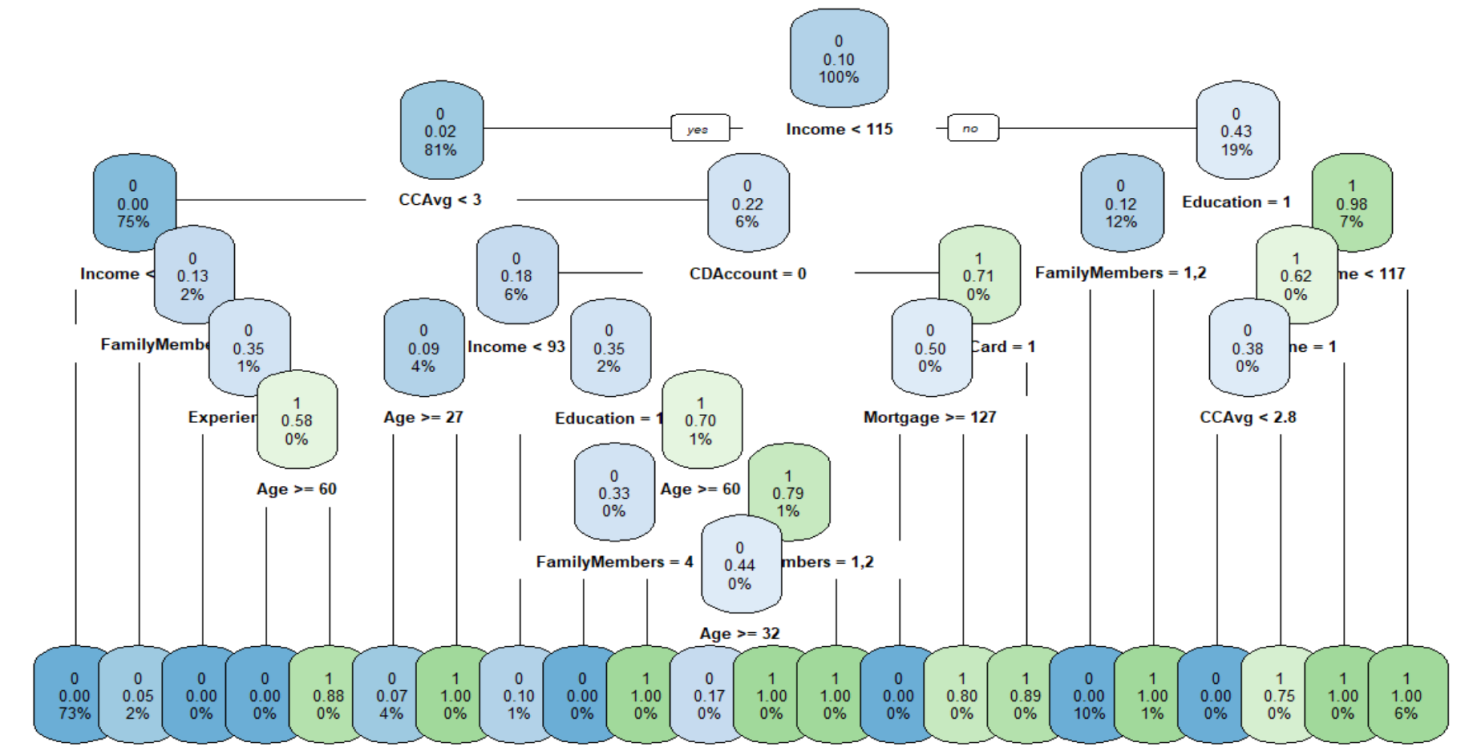
**Pruning the tree using Complexity Parameter**

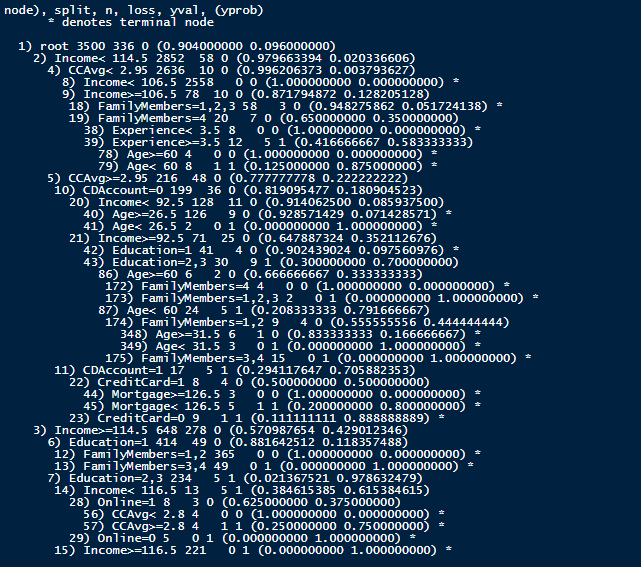
After printing the CP of full tree we observe that cross validation error is minimum at **CP 0.0029762** which is evident from below two plots.

****

****

**Building tree using the best CP and plotting the tree**

****

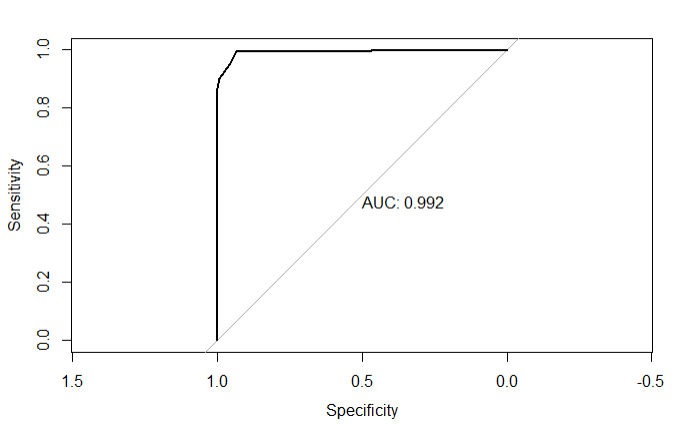
****

****

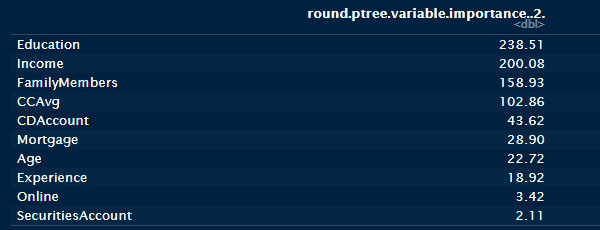
****

Observation

* Even after pruning the tree using best CP, the model performance on test data is not better than full tree.
* Pruning using CP gives slightly better results ~87% than manual pruning ~83%.
* First split is now happening at whether Income is less than or greater than $ 114.5K
* Second split happens on whether the monthly credit card is less than or greater than 2.95K.
* Even though full tree is giving best results in terms of sensitivity we will opt for pruned tree using best CP. Pruning reduces the complexity of the final classifier, and hence improves predictive accuracy by the reduction of over fitting. Also the pruned tree using CP has a very high AUC ~**99%**



**Identifying the variable importance for pruned tree using CP**

****

The variable importance coincides with our EDA findings. Factors affecting the model are

1. Education
2. Income
3. Family Members
4. CC Avg
5. CD Account
6. Mortgage

## 3.5 Applying Random Forests (plot the tree)

Random Forest is one of the most popular and most powerful machine learning algorithms.It is an ensemble method used by combining weak and strong learners to give a better accuracy or output. It’s a combination of multiple trees each chosen randomly to grow on dataset.

First split the data set to train and test model

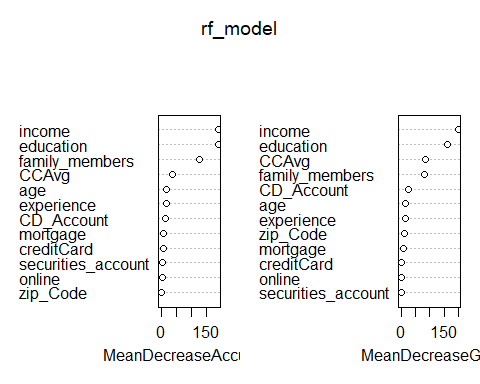
The data we use is usually split into training data and test data. The training set contains a known output and the model learns on this data in order to be generalized to other data later on. We have the test dataset (or subset) in order to test our model’s prediction on this subset. It’s generally around 80/20 or 70/30.

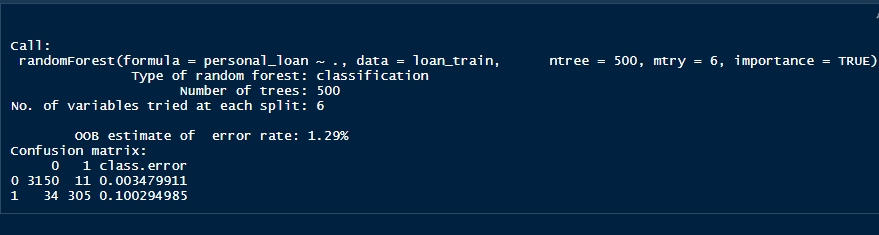
Create random forest model using random forest package

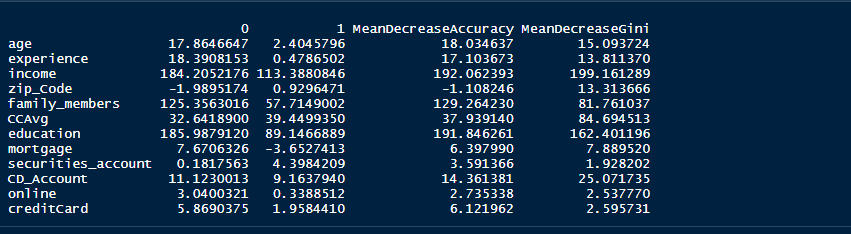
1. Run the Model against Train Data

rf\_model =randomForest(personal\_loan~., data =loan\_train, ntree=500, mtry =6,importance=TRUE)

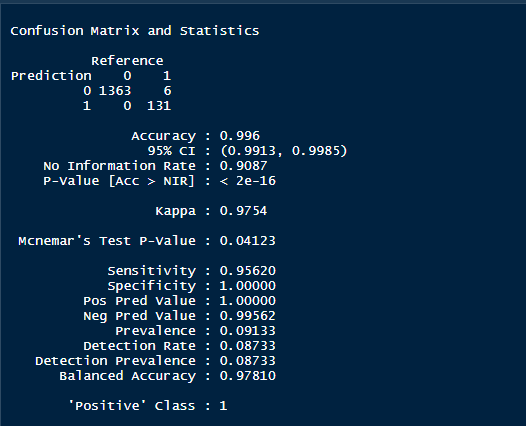
* mtry: Number of variables is randomly collected to be sampled at each split time.
* ntree: Number of branches will grow after each time split.





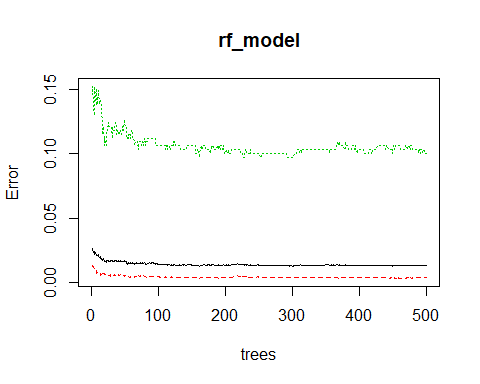
#Variable Importance  


Confusion Matrix for Test Data



## 3.6 Interpret the RF model output (with remarks, making it meaningful for everybody)

Below plot suggests the optimal trees we can use to tune Random forest model  
Somewhere between 90-100 trees should suffice as it saves time to train less trees and achieve same or even better results depending on cases



The error rate plot w.r.t number of trees reveals that anything more than, say 100, trees is really not that valuable.

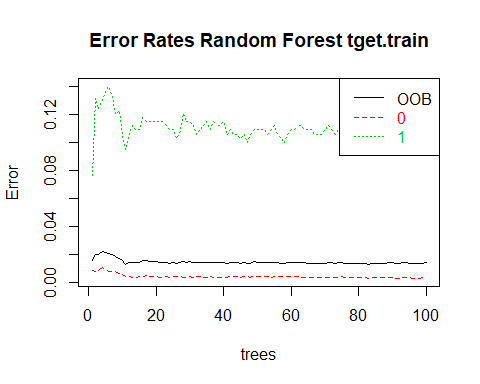
So we ran the model again with ntree=100

rf\_model2 =randomForest(personal\_loan~., data =loan\_train, ntree=100, mtry =6,importance=TRUE)

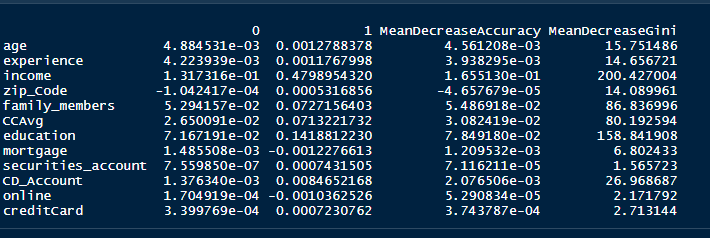
rf\_model2$err.rate

## OOB 0 1  
## [1,] 0.01558846 0.008673027 0.07692308  
## [2,] 0.02030217 0.007874016 0.13145540  
## [3,] 0.02028320 0.008913413 0.12451362  
## [4,] 0.02253329 0.010964083 0.13028169  
## [5,] 0.02166295 0.009184034 0.13636364  
## [6,] 0.02073171 0.007772896 0.14018692  
## [7,] 0.02051130 0.008245383 0.13253012  
## [8,] 0.01876833 0.007797271 0.12048193  
## [9,] 0.01771196 0.006428801 0.12312312  
## [10,] 0.01558442 0.006072228 0.10416667  
## [11,] 0.01321839 0.004454343 0.09495549  
## [12,] 0.01434720 0.004448681 0.10650888  
## [13,] 0.01432254 0.003805899 0.11242604  
## [14,] 0.01431025 0.004119138 0.10946746  
## [15,] 0.01459645 0.004435995 0.10946746  
## [16,] 0.01572777 0.004749842 0.11799410  
## [17,] 0.01572327 0.005064894 0.11504425  
## [18,] 0.01543298 0.004746835 0.11504425  
## [19,] 0.01543298 0.004746835 0.11504425  
## [20,] 0.01542857 0.004745334 0.11504425  
## [21,] 0.01485714 0.004112623 0.11504425  
## [22,] 0.01485714 0.004112623 0.11504425  
## [23,] 0.01485714 0.004428978 0.11209440  
## [24,] 0.01400000 0.003796267 0.10914454  
## [25,] 0.01457143 0.004428978 0.10914454  
## [26,] 0.01400000 0.004428978 0.10324484  
## [27,] 0.01457143 0.004745334 0.10619469  
## [28,] 0.01542857 0.004112623 0.12094395  
## [29,] 0.01485714 0.004112623 0.11504425  
## [30,] 0.01514286 0.004428978 0.11504425  
## [31,] 0.01457143 0.004112623 0.11209440  
## [32,] 0.01428571 0.004428978 0.10619469  
## [33,] 0.01457143 0.004428978 0.10914454  
## [34,] 0.01457143 0.004112623 0.11209440  
## [35,] 0.01485714 0.004112623 0.11504425  
## [36,] 0.01457143 0.004428978 0.10914454  
## [37,] 0.01485714 0.004112623 0.11504425  
## [38,] 0.01428571 0.003796267 0.11209440  
## [39,] 0.01457143 0.004112623 0.11209440  
## [40,] 0.01485714 0.004112623 0.11504425  
## [41,] 0.01400000 0.004112623 0.10619469  
## [42,] 0.01457143 0.004428978 0.10914454  
## [43,] 0.01428571 0.004428978 0.10619469  
## [44,] 0.01428571 0.004428978 0.10619469  
## [45,] 0.01371429 0.004112623 0.10324484  
## [46,] 0.01428571 0.004428978 0.10619469  
## [47,] 0.01371429 0.004428978 0.10029499  
## [48,] 0.01400000 0.004112623 0.10619469  
## [49,] 0.01514286 0.005061689 0.10914454  
## [50,] 0.01457143 0.004428978 0.10914454  
## [51,] 0.01485714 0.004745334 0.10914454  
## [52,] 0.01485714 0.004745334 0.10914454  
## [53,] 0.01428571 0.004428978 0.10619469  
## [54,] 0.01428571 0.004112623 0.10914454  
## [55,] 0.01485714 0.004428978 0.11209440  
## [56,] 0.01428571 0.004428978 0.10619469  
## [57,] 0.01400000 0.004428978 0.10324484  
## [58,] 0.01400000 0.004745334 0.10029499  
## [59,] 0.01457143 0.004745334 0.10619469  
## [60,] 0.01457143 0.004428978 0.10914454  
## [61,] 0.01457143 0.004428978 0.10914454  
## [62,] 0.01485714 0.004428978 0.11209440  
## [63,] 0.01428571 0.003796267 0.11209440  
## [64,] 0.01428571 0.004112623 0.10914454  
## [65,] 0.01371429 0.003479911 0.10914454  
## [66,] 0.01371429 0.003479911 0.10914454  
## [67,] 0.01400000 0.004112623 0.10619469  
## [68,] 0.01371429 0.003796267 0.10619469  
## [69,] 0.01400000 0.004112623 0.10619469  
## [70,] 0.01400000 0.003796267 0.10914454  
## [71,] 0.01428571 0.003796267 0.11209440  
## [72,] 0.01428571 0.004112623 0.10914454  
## [73,] 0.01400000 0.004112623 0.10619469  
## [74,] 0.01428571 0.004112623 0.10914454  
## [75,] 0.01485714 0.004428978 0.11209440  
## [76,] 0.01400000 0.003796267 0.10914454  
## [77,] 0.01371429 0.003796267 0.10619469  
## [78,] 0.01371429 0.003479911 0.10914454  
## [79,] 0.01371429 0.003479911 0.10914454  
## [80,] 0.01400000 0.003479911 0.11209440  
## [81,] 0.01400000 0.003796267 0.10914454  
## [82,] 0.01400000 0.003796267 0.10914454  
## [83,] 0.01342857 0.003163556 0.10914454  
## [84,] 0.01371429 0.003479911 0.10914454  
## [85,] 0.01371429 0.003479911 0.10914454  
## [86,] 0.01400000 0.003796267 0.10914454  
## [87,] 0.01400000 0.003796267 0.10914454  
## [88,] 0.01371429 0.003796267 0.10619469  
## [89,] 0.01428571 0.004112623 0.10914454  
## [90,] 0.01428571 0.003796267 0.11209440  
## [91,] 0.01371429 0.003163556 0.11209440  
## [92,] 0.01400000 0.003163556 0.11504425  
## [93,] 0.01457143 0.003479911 0.11799410  
## [94,] 0.01428571 0.003796267 0.11209440  
## [95,] 0.01400000 0.003479911 0.11209440  
## [96,] 0.01400000 0.003163556 0.11504425  
## [97,] 0.01400000 0.003163556 0.11504425  
## [98,] 0.01400000 0.003163556 0.11504425  
## [99,] 0.01371429 0.003479911 0.10914454  
## [100,] 0.01428571 0.003479911 0.11504425

plot(rf\_model2, main="")  
legend("topright", c("OOB", "0", "1"), text.col=1:6, lty=1:3, col=1:3)  
title(main="Error Rates Random Forest tget.train")



List the importance of the variables. Larger the MeanDecrease values, the more important the variable. Look at the help files to get a better sense of how these are computed.

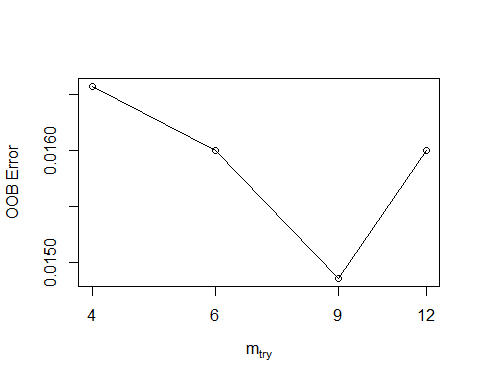


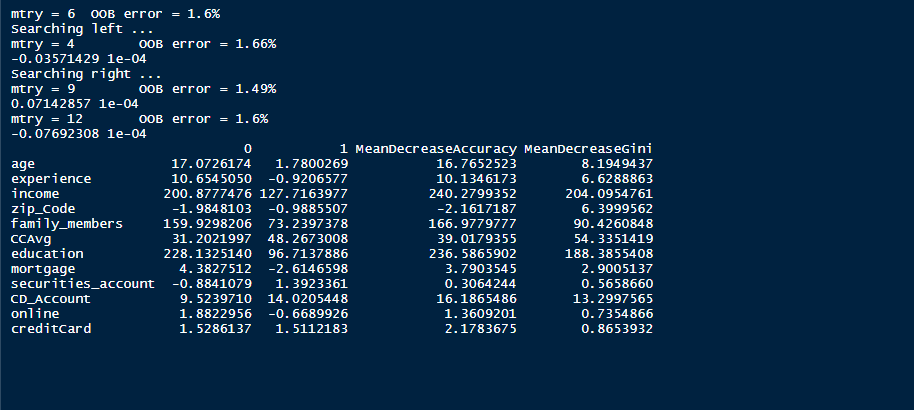
**Tuning Random Forest**

Now we will “**tune**” the Random Forest by trying different m values. We will stick with 101 trees (odd number of trees are preferable).

In randomeForest() have tuneRF() for searching best optimal mtry values given for your data. We will depend on OOBError to define the most accurate mtry for our model which have the least OOBEError.

The returned forest, “rf\_model3” is the one corresponding to the best m





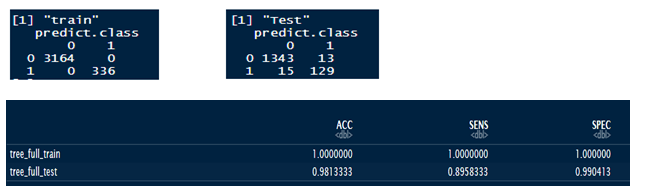
According to this results, mtry = 9 is the best parameter for our model.

Lets make predictions on the training data and measure the prediction error rate.

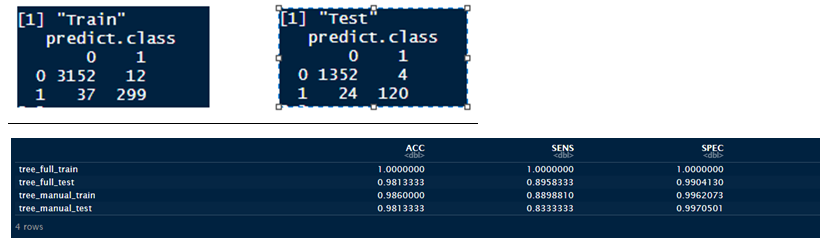
  Above plot depecits the Out of Bag error for Class 0 and Class 1 and Overall OOB error. Various Model Performance Measures (Test & Train): Confusion Matrix

**CART Model’s Confusion Matrix and Performance Measures**

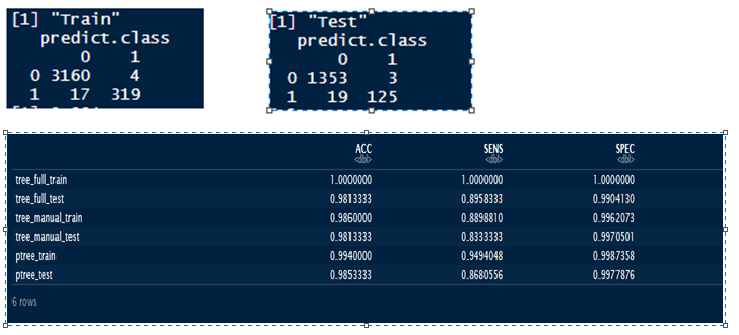
1. Full grown Tree



1. Manual Pruned Tree



1. Pruned Tree Using Complexity Paremeter



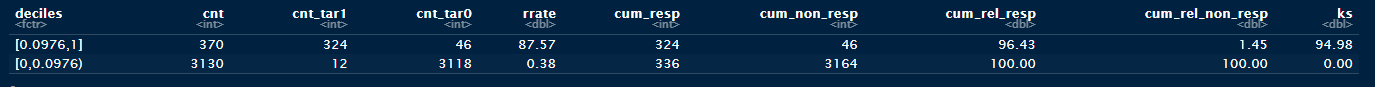
Performance Measures of CART model with CP pruned tree on both train and test data

Train Data

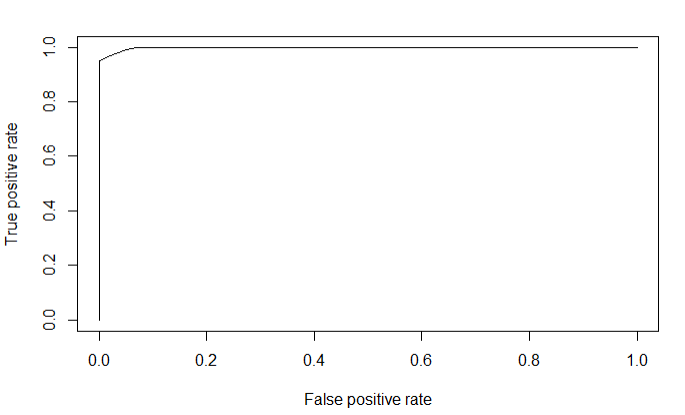
1. Deciles – based on train data, 2 deciles come up which encompasses all the data points.



1. Based on below table we could see that ks is **94.98** and majority of the data point is in first decile **0 to 0.0976** and targeting this decile has lot of potential because it has majority of non responses.



1. Plot True positive rates Vs False Positive Rates

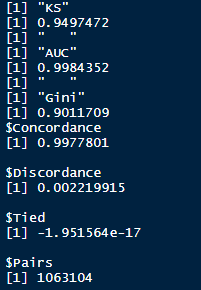


1. KS using the ROCR package also returns 94.98





1. Using Packages ROCR, ineq and InformationValue we find below performance measures

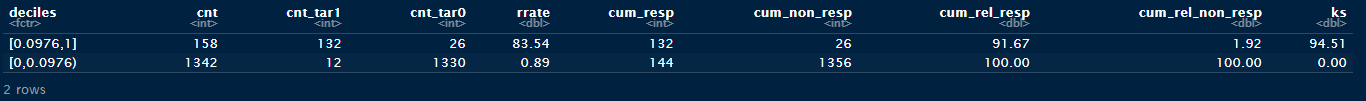


Test Data

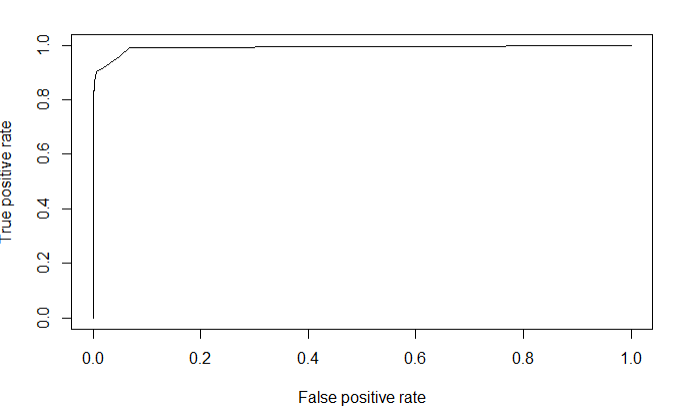
1. Deciles – based on test data, same 2 deciles come up which encompasses all the data points.



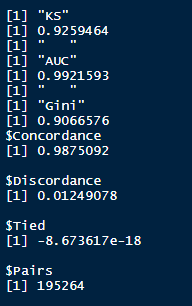
1. Based on below table we could see that ks is **94.51** and majority of the data point is in first decile **0 to 0.0976** and targeting this decile has lot of potential because it has majority of non responses.



1. Plot True positive rates Vs False Positive Rates



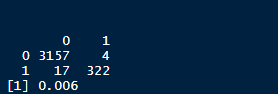
1. KS using the ROCR package also returns 92.59
2. Using Packages ROCR, ineq and InformationValue we find below performance measures



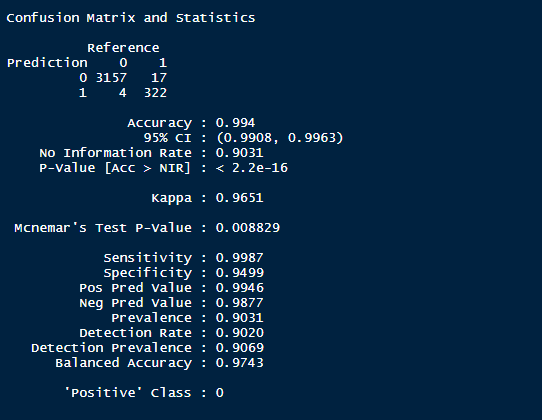
**Performance Measures of RF model on both train and test data**

Train Data

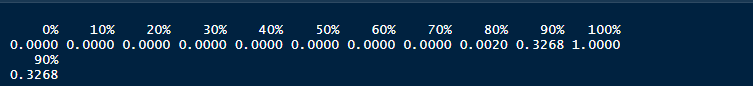
Now using the tuned Random Forest from the previous step, and redo our errors and top decile calculations for the previously identified threshold.



Confusion Matrix



We next find the probability threshold that for the top decile. The choice of what threshold you use is quite subjective and depends on the benefits of having Target=1 vs the cost of sending out, say mailers, to each customer. Since the threshold for the top decile is lower than 0.5, I will use 0.5 and measure what fraction of the top decile actually has a Target=1.



mean((loan\_train$predict[loan\_train$prob1>threshold])=="1")

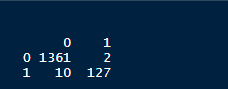
[1] 1

Since the final output is 1.It is concluded that we have captured threshold greater than 0.5. rightly as potential loan customers.

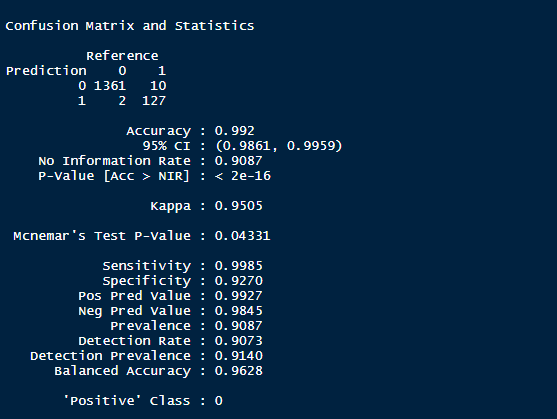
**Test Data**

Let’s make predictions on the test data and measure the prediction error rate.

Scoring Syntax



**Confusion Matrix**

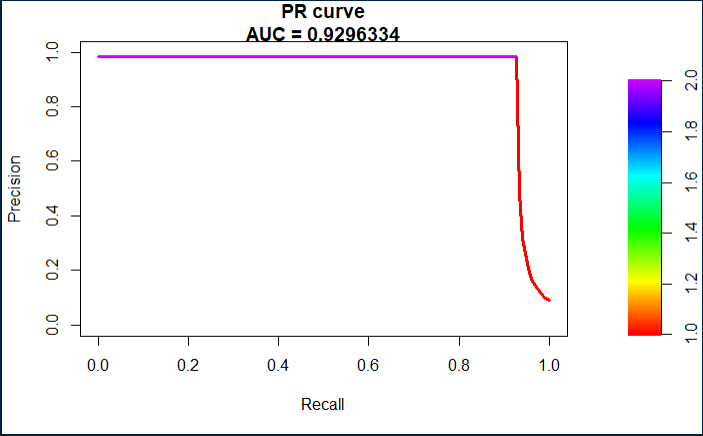


mean((loan\_test$personal\_loan[loan\_test$prob1>threshold])=="1")

## [1] 0.9844961

Since the final output is near to 1. It is concluded that we have captured threshold greater than 0.5. rightly as potential loan customers.

**PR Curve**



A PR curve is simply a graph with Precision values on the y-axis and Recall values on the x-axis. In other words, the PR curve contains TP/(TP+FN) on the y-axis and TP/(TP+FP) on the x-axis.

It is desired that the algorithm should have both high precision, and high recall. However, most machine learning algorithms often involve a trade-off between the two. A good PR curve has greater AUC (area under curve).

PR curve has the Recall value (TPR) on the x-axis, and precision = TP/(TP+FP) on the y-axis. Precision helps highlight **how relevant the retrieved results are**, which is more important while judging an IR system.

Hence, a PR curve is often more common around problems involving information retrieval.

**Gini Coefficient**

Gini coefficient can be straight away derived from the AUC ROC number. Gini is nothing but ratio between area between the ROC curve and the diagonal line & the area of the above triangle. Following is the formulae used

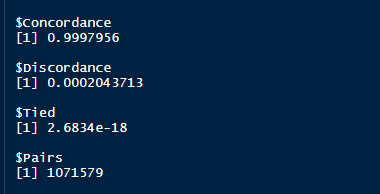
Gini = 2\*AUC – 1

Gini above 60% is a good model. For our case we get Gini as 90%.

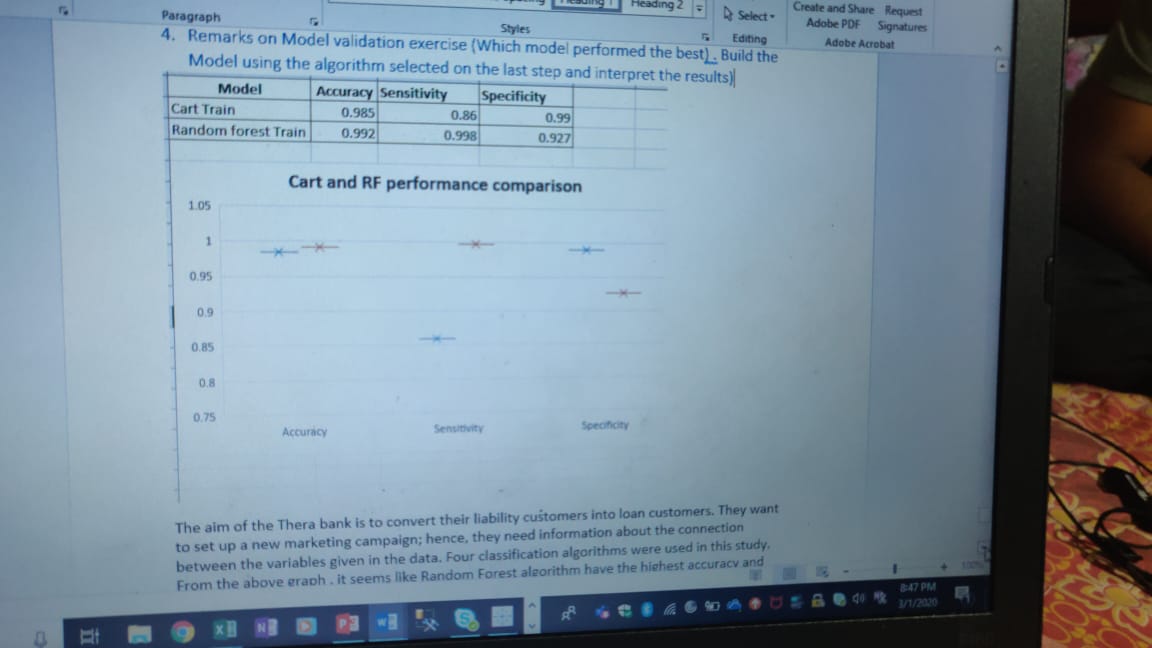
**Concordant – Discordant ratio**

This is again one of the most important metrics for any classification predictions problem. Concordant ratio of more than 60% is considered to be a good model. It is primarily used to access the model’s predictive power. In our case we got 99.9%

Finally, we use the Concordance function in the InformationValue package to find the concordance and discordcance ratios



# Remarks on Model validation exercise (Which model performed the best) /Build the Model using the algorithm selected on the last step and interpret the results)



The aim of the Thera bank is to convert their liability customers into loan customers. They want to set up a new marketing campaign; hence, they need information about the connection between the variables given in the data. CART and Random Forest classification algorithms were used in this study. From the above graph, it seems like Random Forest algorithm have the highest accuracy and we can choose that as our final model.

Various types of models were attempted Some raw, some refined and tuned to display their dissimilarity in approaching the same dataset under mostly similar conditions.  
If given a choice between low OOB (out of bag) error and Accuracy. I will go with accuracy as this case demands so.  
As financial institution we want to be more than 100% sure that there should be no tolerance for defaults and we can earn from interest income  
So under Circumstances ranger (Random Forest) performs the best on dataset with accuracy of 99%