April 18, 2020

**

THERA BANK- LOAN PURCHASE MODELLING

IDENTIFY AND TARGET THE RIGHT LIABILITY CUSTOMERS FOR PERSONAL LOAN CAMPAIGN

Shweta gupta

PGPBABI

**Index**

* Background & Objectives
* Data Dictionary
* Exploratory Data Analysis
* Model Building and model performance checks
  + CART
  + Chaid
  + Random Forest
* Choosing the Best Fit Model

**Background**

* Thera Bank 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 earn more through the interest on loans.
* The management wants to explore ways of converting its liability customers to personal loan customers *.*
* 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.
* The dataset has data on 5000 customers.
* The data include customer demographic information*,* 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.

***Objectives***

* 1. EDA - Basic data summary, Univariate, Bivariate analysis, graphs
* 2.1 Applying CART <plot the tree>
* 2.2 Interpret the CART model output <pruning, remarks on pruning, plot the pruned tree>
* 2.3 Applying Random Forests<plot the tree>
* 2.4 Interpret the RF model output <with remarks, making it meaningful for everybody>
* 3.1 Confusion matrix interpretation
* 3.2 Interpretation of other Model Performance Measures < AUC, ROC>
* 3.3 Remarks on Model validation exercise <Which model performed the best>

***Data Dictionary***

* The target or the dependent variable in the given dataset is “Personal Loan”, which has values **0 OR 1** where 1 means customer accepted the personal loan from bank and 0 means did not accept personal loan from bank
* The dataset has 5000 records with 14 different independent variables. The variables are listed below:

|  |  |
| --- | --- |
| Variable Name | Description |
| ID | Customer ID |
| Age | Customer's age in years |
| Experience | Years of professional experience |
| Income | Annual income of the customer ($000) |
| ZIPCode | Home Address ZIP code. |
| Family | Family size of the customer |
| CCAvg | Avg. spending on credit cards per month ($000) |
| Education | Education Level. 1: Undergrad; 2: Graduate; 3: Advanced/Professional |
| Mortgage | Value of house mortgage if any. ($000) |
| Personal Loan | Did this customer accept the personal loan offered in the last campaign? |
| Securities Account | Does the customer have a securities account with the bank? |
| CD Account | Does the customer have a certificate of deposit (CD) account with the bank? |
| Online | Does the customer use internet banking facilities? |
| CreditCard | Does the customer use a credit card issued by the bank? |

***Exploratory Data Analysis***

* The Data has 5000 rows and 14 variables as below mentioned:

##Viewing the structure of the data (data types)  
str(tbank2)

## 'data.frame': 5000 obs. of 14 variables:  
## $ ID : num 1 2 3 4 5 6 7 8 9 10 ...  
## $ Age..in.years. : num 25 45 39 35 35 37 53 50 35 34 ...  
## $ Experience..in.years.: num 1 19 15 9 8 13 27 24 10 9 ...  
## $ Income..in.K.year. : num 49 34 11 100 45 29 72 22 81 180 ...  
## $ ZIP.Code : Factor w/ 467 levels "9307","90005",..: 84 35  
## $ Family.members : num 4 3 1 1 4 4 2 1 3 1 ...  
## $ CCAvg : num 1.6 1.5 1 2.7 1 0.4 1.5 0.3 0.6 8.9 ...  
## $ Education : Factor w/ 3 levels "1","2","3": 1 1 1 2 2 2 2  
## $ Mortgage : num 0 0 0 0 0 155 0 0 104 0 ...  
## $ Personal.Loan : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1   
## $ Securities.Account : Factor w/ 2 levels "0","1": 2 2 1 1 1 1 1 1 1  
## $ CD.Account : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1  
## $ Online : Factor w/ 2 levels "0","1": 1 1 1 1 1 2 2 1 2

## $ CreditCard : Factor w/ 2 levels "0","1": 1 1 1 1 2 1 1 2 1

* **Check for the missing values** in the data

##Checking for missing values if any:

apply(tbank,2, function(x) sum(is.na(x)))

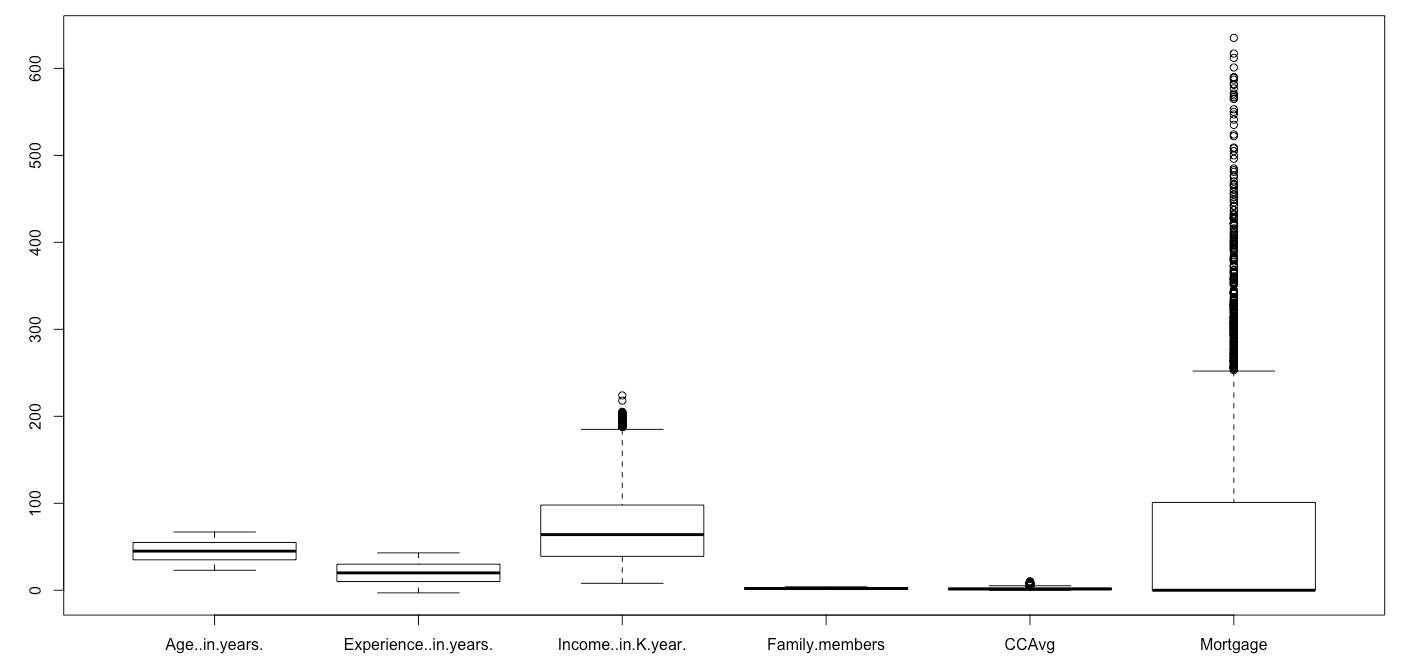
## ID Age..in.years. Experience..in.years.   
## 0 0 0   
## Income..in.K.year. ZIP.Code Family.members   
## 0 0 18   
## CCAvg Education Mortgage   
## 0 0 0   
## Personal.Loan Securities.Account CD.Account   
## 0 0 0   
## Online CreditCard   
## 0 0

#only family members has 18 missing values

* Five Point Summary

summary(tbank2)

## ID Age..in.years. Experience..in.years. Income..in.K.year.  
## Min. : 1 Min. :23.00 Min. :-3.0 Min. : 8.00   
## 1st Qu.:1251 1st Qu.:35.00 1st Qu.:10.0 1st Qu.: 39.00   
## Median :2500 Median :45.00 Median :20.0 Median : 64.00   
## Mean :2500 Mean :45.34 Mean :20.1 Mean : 73.77   
## 3rd Qu.:3750 3rd Qu.:55.00 3rd Qu.:30.0 3rd Qu.: 98.00   
## Max. :5000 Max. :67.00 Max. :43.0 Max. :224.00   
##   
## ZIP.Code Family.members CCAvg Education Mortgage   
## 94720 : 169 Min. :1.000 Min. : 0.000 1:2096 Min. : 0.0   
## 94305 : 127 1st Qu.:1.000 1st Qu.: 0.700 2:1403 1st Qu.: 0.0   
## 95616 : 116 Median :2.000 Median : 1.500 3:1501 Median : 0.0   
## 90095 : 71 Mean :2.398 Mean : 1.938 Mean : 56.5   
## 93106 : 57 3rd Qu.:3.000 3rd Qu.: 2.500 3rd Qu.:101.0   
## 92037 : 54 Max. :4.000 Max. :10.000 Max. :635.0   
## (Other):4406   
## Personal.Loan Securities.Account CD.Account Online CreditCard  
## 0:4520 0:4478 0:4698 0:2016 0:3530   
## 1: 480 1: 522 1: 302 1:2984 1:1470   
##

**

**BOXPLOT OF CONTINUOUS VARIABLES**

**Key observations/actions:**

* *The variable names were revised to R supported format.*
* *The data type of categorical variables* ***– Zipcode, Education, Personal Loan, Online, SD Account, Securities account, CD Account, CreditCard*** *was changed to factor from numeric.*
* *Family members have 18 missing values – we used mice package to impute the missing values.*
* *Experience has 52 negative values, which does not make intuitive sense and hence these values were converted to NA and further imputed using KNN function.*
* *Income, CCavg and Mortgage have outliers, which were treated using the flooring and ceiling method – we looked at their distribution and capped all outliers to the 5th percentile and 95th percentile values of these variables.*

**Analysing the dependent variable – Personal Loan**

prop.table(table(tbank\_wo\_na$Personal.Loan))

##   
## 0 1   
## 0.904 0.096

hist(tbank\_wo\_na$Personal.Loan)

# only 480 or 9.6% of customers have availed the personal loan out of total 5000 customers

**Summarising Independent variables**

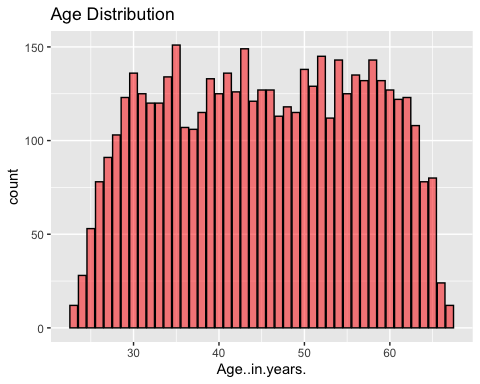
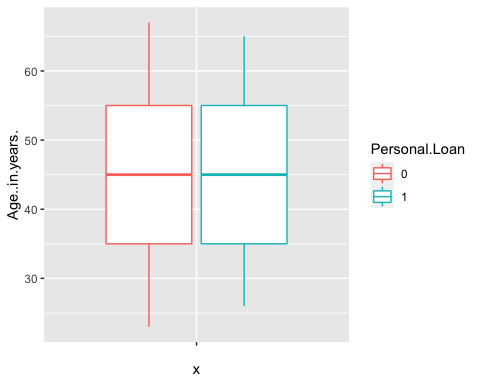
#### Age -- Continuous variable ####

summary**(Age..in.years.)**

**## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 23.00 35.00 45.00 45.34 55.00 67.00**

Boxplot Age ~ Loan

Histogram of Age



* No major difference in median age for those who availed personal loan and those who didn’t availed the personal loan. Age seem to follow near normal distribution

#### Experience-- Continuous Variable ####

summary(tbank2$Experience..in.years.)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -3.0 10.0 20.0 20.1 30.0 43.0

View(Experience..in.years.[Experience..in.years.<0])  
  
sum(tbank2$Experience..in.years.<0)

## [1] 52

# negative experience values.- total 52 values  
# We replaced the negative values with imputed values using KNN

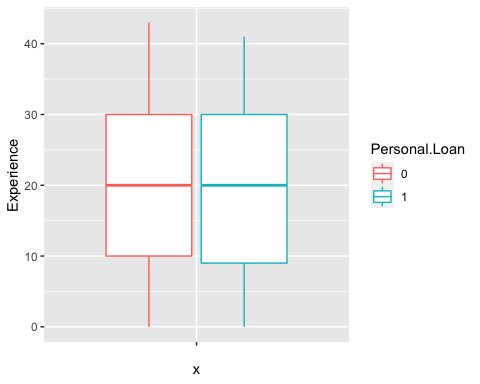
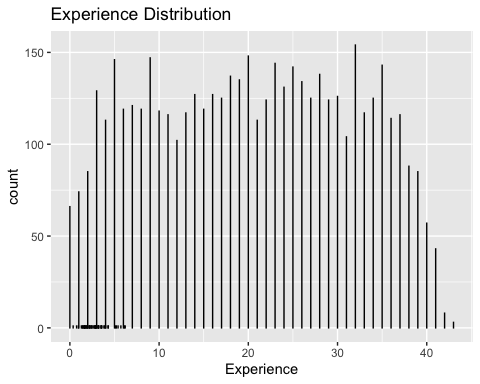
* Revised summary post removing negative Experience values

summary(tbank3$Experience)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.00 10.00 20.00 20.15 30.00 43.00

Boxplot of Experience ~ Loan

Histogram of Experience



* Experience variable follows near normal distribution
* No major difference in median experience for those who availed personal loan and who did not.

#### Income -- Continuous Variable #####

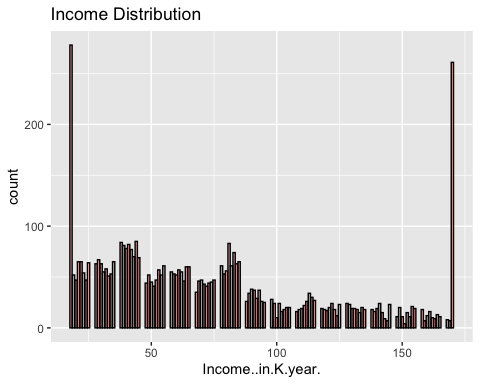
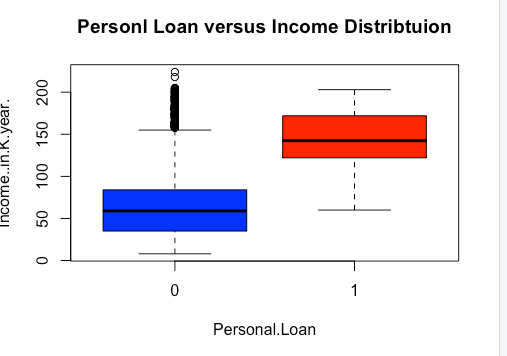
summary(tbank3$Income..in.K.year.)

Min. 1st Qu. Median Mean 3rd Qu. Max.

18.00 39.00 64.00 73.32 98.00 170.00

Boxplot of Income ~ Loan

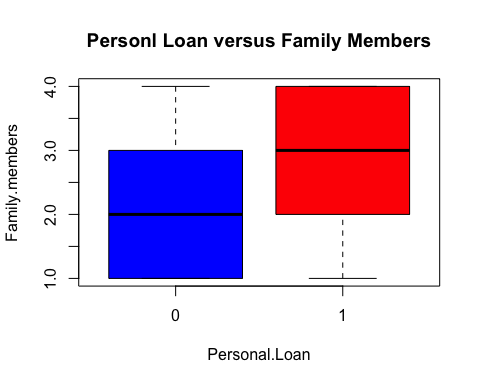
Histogram of Income



* Those availed Personal loan have a higher income compared to those who did not
* Income variable seems highly skewed

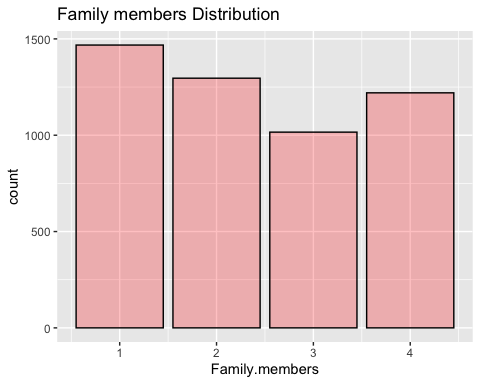
#### Family members -- Integer Variable #####

summary(Family.members)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 2.000 2.398 3.000 4.000

Boxplot of Family ~ Loan

Histogram of Family



table(Family.members, Personal.Loan)

## Personal.Loan  
## Family.members 0 1  
## 1 1362 106  
## 2 1189 107  
## 3 883 133  
## 4 1086 134

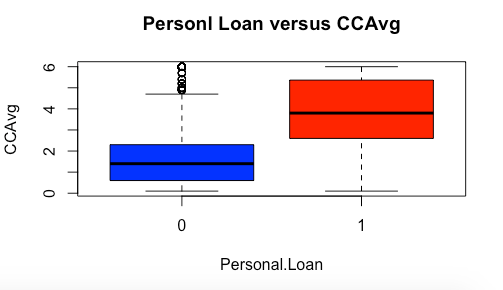
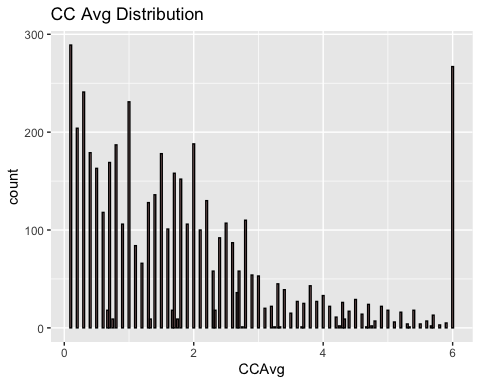
* Those with bigger family size have availed more loans

#### CC AVG -- Continuous Variable #####

summary(CCAvg)

Min. 1st Qu. Median Mean 3rd Qu. Max.

0.10 0.70 1.50 1.88 2.50 6.00



Boxplot of CC Avg ~ Loan

Histogram of CCAvg

* CCavg seems to be a skewed variable

### Education - Categorical Variable ###

summary(Education)

## 1 2 3   
## 2096 1403 1501

#Education factor varies from 1 to 3, mostly being on the lower side.  
prop.table(table(Education , Personal.Loan),margin = 1)

## Personal.Loan  
## Education 0 1  
## 1 0.95562977 0.04437023  
## 2 0.87027798 0.12972202  
## 3 0.86342438 0.13657562

* Those with higher education have availed more loans.

### Securities.Account - Categorical Variable ###

summary(Securities.Account)

## 0 1   
## 4478 522  
prop.table(table(Securities.Account , Personal.Loan),margin = 1)

## Personal.Loan  
## Securities.Account 0 1  
## 0 0.90620813 0.09379187  
## 1 0.88505747 0.11494253

### CD.Account - Categorical Variable ###

summary(CD.Account)

## 0 1   
## 4698 302  
prop.table(table(CD.Account , Personal.Loan),margin = 1)

## Personal.Loan  
## CD.Account 0 1  
## 0 0.92762878 0.07237122  
## 1 0.53642384 0.463576

* Those with CD account and securities account with Bank have availed more personal loans

### Online - Categorical Variable ###

summary(Online)

## 0 1   
## 2016 2984.

prop.table(table(Online , Personal.Loan),margin = 1)

## Personal.Loan  
## Online 0 1  
## 0 0.90625000 0.09375000  
## 1 0.90247989 0.09752011

### CreditCard - Categorical Variable ###

summary(CreditCard)

## 0 1   
## 3530 1470  
prop.table(table(CreditCard , Personal.Loan),margin = 1)

## Personal.Loan  
## CreditCard 0 1  
## 0 0.90453258 0.09546742  
## 1 0.90272109 0.09727891

* No major difference in credit card and online variable for those who took personal loan and who did not.

### Mortgage - Continuous Variable ###

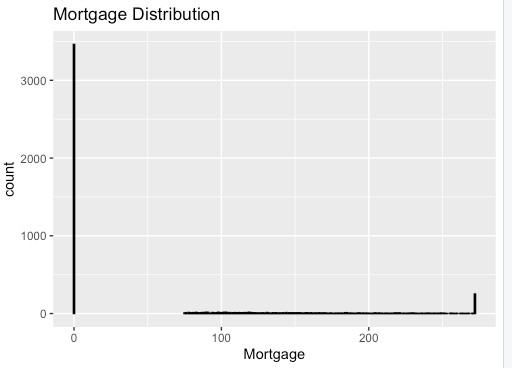
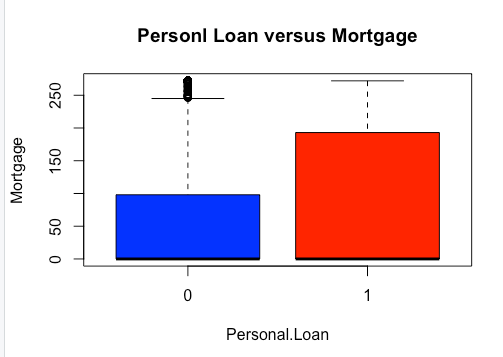
summary(tbank3$Mortgage)

Min. 1st Qu. Median Mean 3rd Qu. Max.

0.00 0.00 0.00 51.55 101.00 272.00

Boxplot of CC Mortgage ~ Loan

Histogram of Mortgage



* Obviously those who took personal loan have a higher mortgage

Performing Chi Square test on categorical Independent variables and dependent variable Personal Loan

Pearson's Chi-squared test

data: Personal.Loan and ZIP.Code

X-squared = 444.15, df = 466, p-value = 0.7597

data: Personal.Loan and Family.members

X-squared = 29.228, df = 3, p-value = 2.006e-06

data: Personal.Loan and Securities.Account

X-squared = 2.1723, df = 1, p-value = 0.1405

data: Personal.Loan and Education

X-squared = 111.24, df = 2, p-value < 2.2e-16

data: Personal.Loan and CreditCard

X-squared = 0.021144, df = 1, p-value = 0.8844

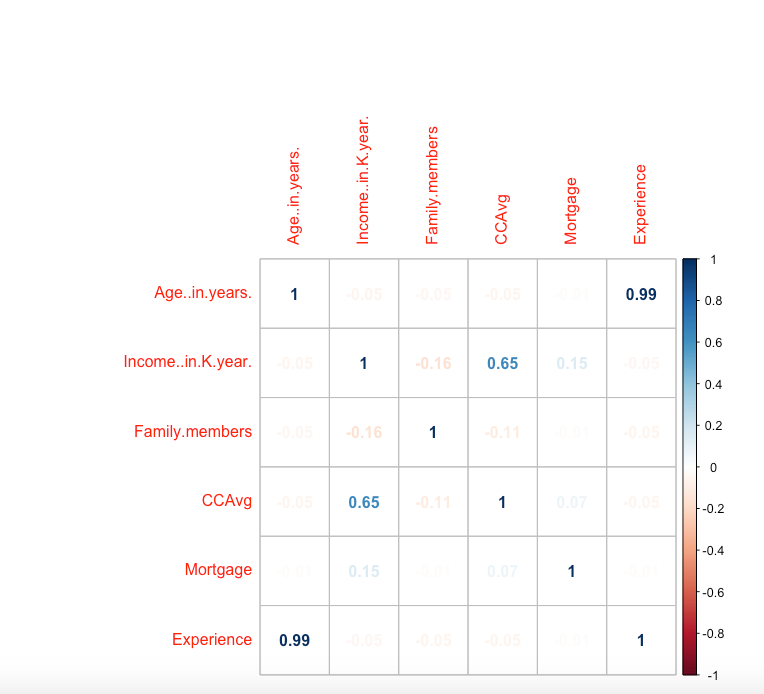
data: Personal.Loan and Online

X-squared = 0.15601, df = 1, p-value = 0.6929

data: Personal.Loan and CD.Account

X-squared = 495.9, df = 1, p-value < 2.2e-16

* The significant categorical variables in explaining our dependent variable seem to be Family members, Education and CD Account

**Correlation among Independent Continuous variables**

* Experience and Age are highly correlated

**MODEL BUILDING**

**###CART###**

Key steps

* Split the data in Test and Train in 70:30 ratio
* Build the model on the Train data set
* Set control parameters to grow the tree
  + #minsplit: if the number of records in a given node falls below a threshold, the node will not be split further.
  + #minbucket: minimum records in a terminal node. if the records are less, that bucket will not be created.
  + #Terminal node (minbucket) should not be less than 2-3% of starting population.
  + #minsplit = 3(minbucket)
  + #xval divides the entire dataset into mutually exclusive and collectively exhaustive segments.
* Predict on Train and Test dataset
* Check Model Performance measure

**CART MODEL**

*# Setting Control Parameters*

r r.ctrl = rpart.control(minsplit=90, minbucket = 30, cp = 0, xval = 10)

# Using rpart to build the tree  
  
tbank.ct <- rpart(formula = Personal.Loan ~ ., data = tbankfinal\_train, method = "class", control = r.ctrl)  
print(tbank.ct)

## n= 3500   
##   
## node), split, n, loss, yval, (yprob)  
## \* denotes terminal node  
##   
## 1) root 3500 336 0 (0.904000000 0.096000000)   
## 2) Income..in.K.year.< 101.5 2656 29 0 (0.989081325 0.010918675)   
## 4) CCAvg< 2.95 2499 0 0 (1.000000000 0.000000000) \*  
## 5) CCAvg>=2.95 157 29 0 (0.815286624 0.184713376)   
## 10) ZIP.Code=90024,90029,90041,90066,90089,90095,90245,90254,90274,90277,90717,90745,91103,91116,91311,91320,91330,91335,91361,91367,91423,91604,91709,91902,91941,92007,92028,92029,92064,92069,92096,92103,92116,92120,92123,92124,92152,92192,92373,92407,92507,92634,92647,92653,92677,92703,92870,93023,93065,93106,93117,93305,93403,93407,93460,93907,93943,93950,94002,94005,94015,94024,94025,94111,94115,94301,94304,94501,94542,94596,94606,94707,94720,94801,94901,94928,94939,94960,95020,95045,95051,95070,95133,95134,95136,95307,95403,95521,95616,95747,95762 120 3 0 (0.975000000 0.025000000) \*  
## 11) ZIP.Code=90034,90291,92037,92093,92121,92122,92220,92717,93022,93311,93555,94085,94143,94234,94305,94306,94709,95014,95032,95039,95053,95064,95814,95929 37 11 1 (0.297297297 0.702702703) \*  
## 3) Income..in.K.year.>=101.5 844 307 0 (0.636255924 0.363744076)   
## 6) Education=1 535 59 0 (0.889719626 0.110280374)   
## 12) Family.members< 2.5 463 2 0 (0.995680346 0.004319654) \*  
## 13) Family.members>=2.5 72 15 1 (0.208333333 0.791666667) \*  
## 7) Education=2,3 309 61 1 (0.197411003 0.802588997)   
## 14) Income..in.K.year.< 116.5 94 33 0 (0.648936170 0.351063830)   
## 28) ZIP.Code=90024,90025,90035,90064,90089,90095,90304,90503,91107,91125,91330,91360,91367,91380,91401,91768,92038,92093,92115,92120,92123,92521,92697,92709,92780,93305,93711,93933,93943,94005,94010,94105,94143,94301,94305,94542,94608,94611,95014,95035,95054,95351,95616,95670 56 3 0 (0.946428571 0.053571429) \*  
## 29) ZIP.Code=90049,90245,90630,91423,91711,91765,91911,92007,92037,92096,92333,92612,92677,92717,93014,93023,93106,94110,94111,94304,94501,94606,94720,94928,94949,95008,95032,95039,95051 38 8 1 (0.210526316 0.789473684) \*  
## 15) Income..in.K.year.>=116.5 215 0 1 (0.000000000 1.000000000) \*

##To see how the tree performs

printcp(tbank.ct)

## Variables actually used in tree construction:  
## [1] CCAvg Education Family.members Income..in.K.year.  
## [5] ZIP.Code   
##   
## Root node error: 336/3500 = 0.096  
##   
## n= 3500   
##

CP nsplit rel error xerror xstd

1 0.278274 0 1.00000 1.00000 0.051870

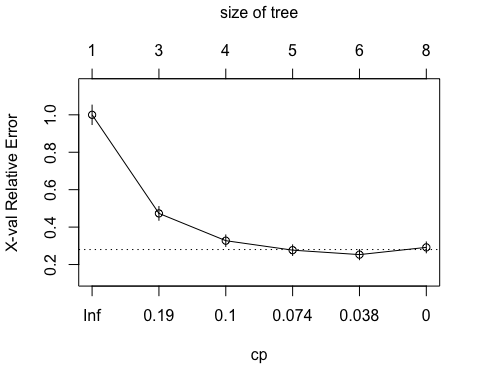
2 0.125000 2 0.44345 0.47321 0.036666

3 0.083333 3 0.31845 0.33631 0.031122

4 0.065476 4 0.23512 0.29762 0.029334

**5 0.022321 5 0.16964 0.26786 0.027869**

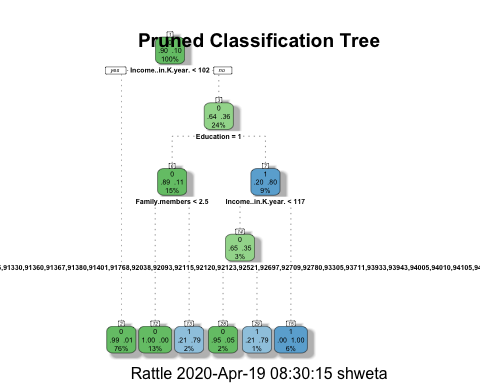
6 0.000000 7 0.12500 0.29167 0.029047



* The x val relative error starts to increase after 5 splits, therefore we will prune the tree with **Complexity parameter 0.03**

##Pruning the tree - Prune wherever xerror is min and it starts increasing after that  
tbank.ct1<- prune(tbank.ct, cp= .03 ,"CP")  
printcp(tbank.ct1)

##   
## Classification tree:  
## rpart(formula = Personal.Loan ~ ., data = tbankfinal\_train, method = "class",   
## control = r.ctrl)  
##   
## Variables actually used in tree construction:  
## [1] Education Family.members Income..in.K.year. ZIP.Code   
##   
## Root node error: 336/3500 = 0.096  
##   
## n= 3500   
##   
## CP nsplit rel error xerror xstd  
## 1 0.278274 0 1.00000 1.00000 0.051870  
## 2 0.125000 2 0.44345 0.47321 0.036666  
## 3 0.083333 3 0.31845 0.32738 0.030720  
## 4 0.065476 4 0.23512 0.27679 0.028317  
## 5 0.030000 5 0.16964 0.25298 0.027104

INfancyRpartPlot(tbank.ct1, uniform=TRUE, main="Pruned Classification Tree"

**PRUNED CLASSIFICATION TREE**

n= 3500

node), split, n, loss, yval, (yprob)

\* denotes terminal node

1) root 3500 336 0 (0.904000000 0.096000000)

2) Income..in.K.year.< 101.5 2656 29 0 (0.989081325 0.010918675) \*

3) Income..in.K.year.>=101.5 844 307 0 (0.636255924 0.363744076)

6) Education=1 535 59 0 (0.889719626 0.110280374)

12) Family.members< 2.5 463 2 0 (0.995680346 0.004319654) \*

13) Family.members>=2.5 72 15 1 (0.208333333 0.791666667) \*

7) Education=2,3 309 61 1 (0.197411003 0.802588997)

14) Income..in.K.year.< 116.5 94 33 0 (0.648936170 0.351063830)

28) ZIP.Code=90024,90025,90035,90064,90089,90095,90304,90503,91107,91125,91330,91360,91367,91380,91401,91768,92038,92093,92115,92120,92123,92521,92697,92709,92780,93305,93711,93933,93943,94005,94010,94105,94143,94301,94305,94542,94608,94611,95014,95035,95054,95351,95616,95670 56 3 0 (0.946428571 0.053571429) \*

29) ZIP.Code=90049,90245,90630,91423,91711,91765,91911,92007,92037,92096,92333,92612,92677,92717,93014,93023,93106,94110,94111,94304,94501,94606,94720,94928,94949,95008,95032,95039,95051 38 8 1 (0.210526316 0.789473684) \*

15) Income..in.K.year.>=116.5 215 0 1 (0.000000000 1.000000000) \*

**Interpretation of the cart tree**

* **The prediction for root node** is 0, probability for same is 90%, total observations are 3500 and 336 are wrong predictions/cases
* **Second split** is Income< 101.5, , prediction for this node is **0**,with probability 98%, total observations are 2656 and 29 are wrong cases
* Similarly for node 7 , if education is 2 or 3 (i.e igher education) (and also satisfying the conditions from the top of the tree upto this level) the prediction is 1, with probability 80%, ,total observations in this node are 309 and no of wrong cases/prediction is 61

**Model performance on Cart Train model**

Confusion Matrix - table( Prediction, Personal.Loan)

**0 1**

**0 3141 34**

1. **23 302**

* Correct predictions of 1 = 302
* Incorrect predictions of 1 = 23
* Correct predictions of 0 = 3141
* Incorrect predictions of 0 = 34

**> sensitivity\_train - total no correct predictions of 1 out of total predictions of 1**

[1] 0.8988095

**> specificity\_train - total no correct predictions of 0 out of total predictions of o**

[1] 0.9927307

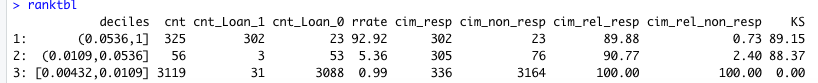
**> accuracy\_train - rati0 of correct predictions to total observations**

[1] 0.9837143 -

**> Error\_Train - rati0 of incorrect predictions to total observations**

[1] 0.01628571

**Deciles For The Cart Train Model**

****

**AUC For The Cart Train Model**

> print(auc)

[1] 0.9554997

* **The model accuracy is 98% for our Cart train model**
* **KS is 89.15% (K-S is a measure of the degree of separation between the positive and negative distributions.)**
* **Area under curve is 95.5%**

**#### CHAID ####**

**KEY STEPS**

* Split the data in Test and Train in 70:30 ratio
* Remove all continuous variables and keep all categorical variables (except Zip code), since Chaid can be run only using categorical variables
* Build the model on the Train data set
* Predict on Train and Test dataset
* Check Model Performance measure

ctrl <- chaid\_control(minbucket = 30, minsplit = 90, alpha2=.05, alpha4 = .05)

chaid.tree <-chaid(Personal.Loan ~., data=tbank\_train\_chaid[,-1], control = ctrl)

> print(chaid.tree)

Model formula:

Personal.Loan ~ Education + Securities.Account + CD.Account +

Online + CreditCard

Fitted party:

[1] root

| [2] CD.Account in 0

| | [3] Education in 1

| | | [4] Online in 0: 0 (n = 571, err = 5.3%)

| | | [5] Online in 1

| | | | [6] CreditCard in 0: 0 (n = 572, err = 2.8%)

| | | | [7] CreditCard in 1: 0 (n = 220, err = 0.0%)

| | [8] Education in 2, 3

| | | [9] CreditCard in 0

| | | | [10] Securities.Account in 0: 0 (n = 1292, err = 11.7%)

| | | | [11] Securities.Account in 1

| | | | | [12] Online in 0: 0 (n = 56, err = 8.9%)

| | | | | [13] Online in 1: 0 (n = 80, err = 0.0%)

| | | [14] CreditCard in 1

| | | | [15] Online in 0: 0 (n = 248, err = 12.1%)

| | | | [16] Online in 1: 0 (n = 247, err = 0.0%)

| [17] CD.Account in 1

| | [18] Education in 1: 0 (n = 88, err = 26.1%)

| | [19] Education in 2, 3

| | | [20] CreditCard in 0: 1 (n = 30, err = 13.3%)

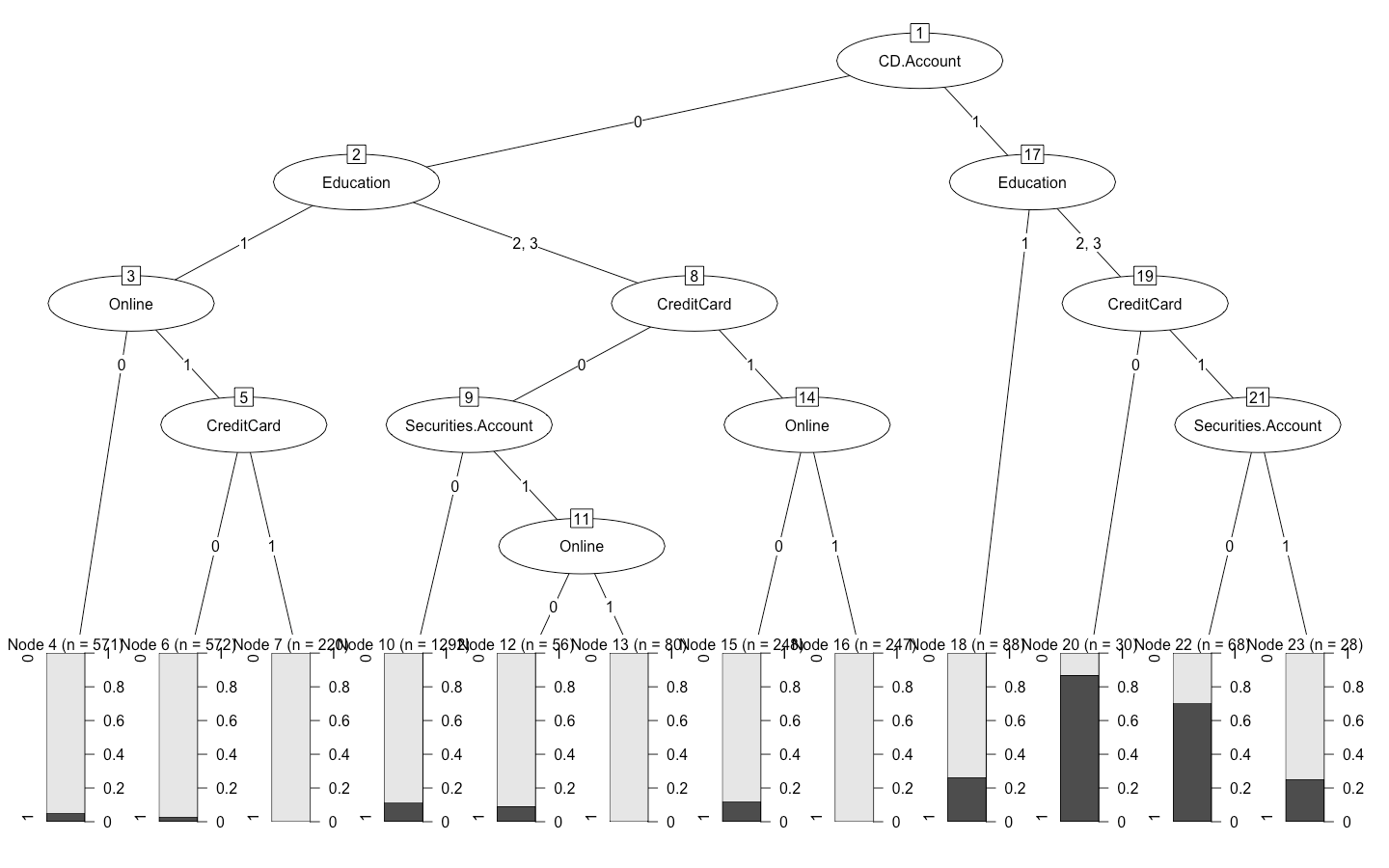
| | | [21] CreditCard in 1

| | | | [22] Securities.Account in 0: 1 (n = 68, err = 29.4%)

| | | | [23] Securities.Account in 1: 0 (n = 28, err = 25.0%)

Number of inner nodes: 11

Number of terminal nodes: 12



**Model performance on Chaid Train model**

Confusion Matrix - table( Prediction, Personal.Loan)

**0 1**

**0 3140 262**

**1 24 74**

* Correct predictions of 1 = 74
* Incorrect predictions of 1 = 24

**> sensitivity\_train\_chaid**

**[1] 0.2202381**

**> specificity\_train\_chaid**

**[1] 0.9924147**

**> accuracy\_train\_chaid**

**[1] 0.9182857**

**> Error\_Train\_chaid**

**[1] 0.08171429**

**> auc\_train\_chaid**

**[1] 0.7708832**

* **The model accuracy is 91.8% for our Chaid train model, however sensitivity is quite low at 22%.**
* **AUC is 77%, which is quite low compared to cart model**

**###RANDOM FOREST ###**

**Key Steps**

* Split the data in Test and Train in 70:30 ratio
* Remove Zip code from the data
* Build the model on the Train data set
  + Set Nodesize - no of observations in terminal node
  + Set Mtry - no of independent variables in a tree out of total Independent variables
  + nTree – no of trees to be built, ideally an odd no, to avoid ties in voting while calculating error rates
  + Importance - Give importance matrix of Independent variables
* Tune the random forest to include optimal number of trees
* Predict on Train and Test dataset
* Check Model Performance measures

**Initial Tree**

randomForest(formula = Personal.Loan ~ ., data = tbank\_train\_rforest[, -3], type = "class", mtry = 5, nodesize = 10, ntree = 501, importance = TRUE)

Type of random forest: classification

Number of trees: 501

No. of variables tried at each split: 5

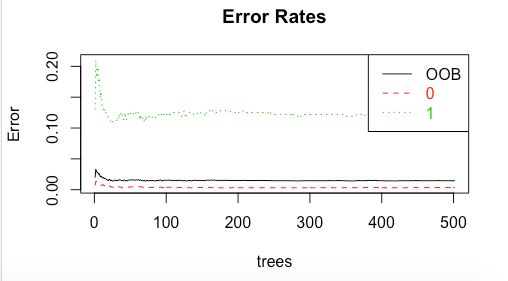
**OOB estimate of error rate: 1.46%**

Confusion matrix:

0 1 class.error

0 3153 11 0.003476612

1 40 296 0.119047619



* Our initial random forest OOB error rate is 1.46% (average error rate among all trees)
* From above plot we can see that there is not much improvement in OOB error rate after 200 + trees, therefore we will build the tuned random forest with 201 trees, and no of variables to start with = 5

**Tuned Random Forest**

tRF <- tuneRF(x = tbank\_train\_rforest[,-c(3,8)],

y=tbank\_train\_rforest$Personal.Loan,

mtryStart = 5,

ntreeTry=201,

stepFactor = 1.5,

improve = 0.0001,

trace=TRUE,

plot = TRUE,

doBest = TRUE,

nodesize = 10,

importance=TRUE

)

Call:

randomForest(x = x, y = y, mtry = res[which.min(res[, 2]), 1], nodesize = 10, importance = TRUE)

Type of random forest: classification

Number of trees: 500

No. of variables tried at each split: 5

**OOB estimate of error rate: 1.43%**

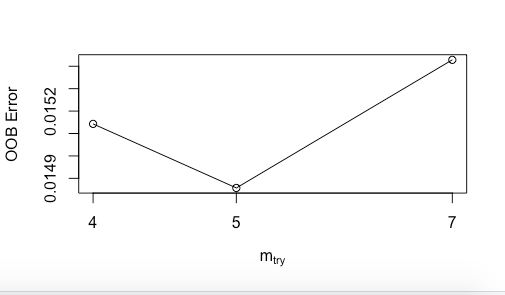
Confusion matrix:

0 1 class.error

0 3154 10 0.003160556

1 40 296 0.119047619

**OOB Error VS no of Columns**



**Importance of Independent variables**

## 0 1 MeanDecreaseAccuracy  
## Age..in.years. 3.121716e-03 -1.687536e-03 2.660577e-03  
## Income..in.K.year. 1.370821e-01 4.944042e-01 1.713328e-01  
**## Family.members 5.689349e-02 7.637999e-02 5.876891e-02**  
## CCAvg 2.276475e-02 5.376743e-02 2.575399e-02  
**## Education 7.609735e-02 1.419173e-01 8.240904e-02**  
## Mortgage 2.390744e-04 -4.783519e-04 1.720955e-04  
## Securities.Account -2.044985e-05 3.332268e-04 1.243804e-05  
## CD.Account 1.751030e-03 5.633317e-03 2.127300e-03  
## Online 9.364405e-05 1.150810e-03 1.963709e-04  
## CreditCard 1.596659e-04 -2.161194e-04 1.251240e-04  
## Experience 2.322287e-03 7.721125e-05 2.101348e-03  
## MeanDecreaseGini  
## Age..in.years. 9.2970635  
**## Income..in.K.year. 202.6010668**  
## Family.members 89.2568994  
## CCAvg 61.2941438  
**## Education 174.9432304**  
## Mortgage 3.2915162  
## Securities.Account 0.6802564  
## CD.Account 17.9387795  
## Online 1.6787059  
## CreditCard 0.9129786  
## Experience 7.4555622

* Tuned random forest gives only marginal improvement in OOB error rate from 1.46 % to 1.43% and it sticks to 500 trees and 5 variables
* From the above graph it can be seen that Random forest tries to build the model with 4 + variables but the OOB error increases when no of variables increases from 5.Hence the model sticks to 5 variables
* **Family members and Education are top 2 important** variables in random forest model based on the mean decrease in Accuracy and Gini as per above table(which is the measure of decrease in model accuracy if the values of the particular variable are shuffled up )

**MODEL PERFORMANCE ON RANDOM FOREST TRAIN MODEL**

Confusion Matrix - table( Prediction, Personal.Loan)

**0 1**

**0 3162 23**

**1 2 313**

* Correct predictions of 1 = 313
* In correct predictions of 1 = 2
* Correct predictions of 0 = 3162
* In correct predictions of 0 = 23

**> sensitivity\_train\_rf - total no correct predictions of 1 out of total predictions of 1**

[1] 0.9315476

**> specificity\_train\_rf - total no correct predictions of 0 out of total predictions of o**

[1] 0.9993679

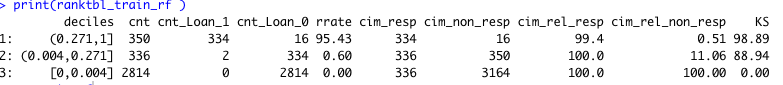
**> accuracy\_train\_rf - ratio of correct predictions to total observations**

[1] 0.9928571

**> Error\_Train\_rf - ratio of correct predictions to total observations**

[1] 0.007142857

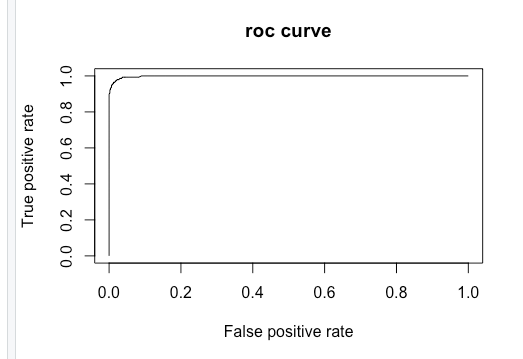
**Deciles For The Random Forest Train Model**

****

**AUC For The Random Forest Train Model**

> print(auc)

[1] 0.9998344

****

* **The model accuracy is 99% for our Random Forest train model**
* **KS is 98.89%**
* **Area under curve is 99.9%**

**CHOOSING THE BEST FIT MODEL**

**MODEL PERFORMANCE MEASURES FOR ALL THE MODELS**

Table Column explanations:-

**Test\_rf\_key - Tuned** Random Forest Test Model

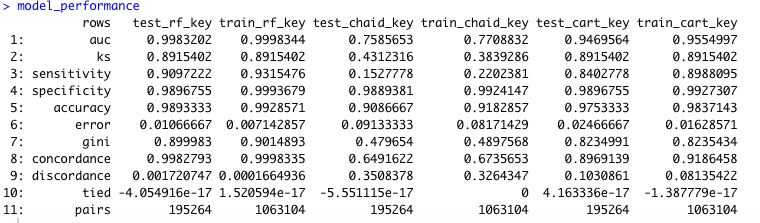
**Train\_rf\_key - Tuned** Random Forest Train Model

**Test\_chaid\_key -** Chaid Test Model

**Train\_chaid\_key -** Chaid Train Model

**Test\_cart\_key - Pruned** Cart Test Model

**Train\_cart\_key. - Pruned** Cart Train Model

****

**KEY OBSERVATIONS AND CONCLUSIONS**

* From the above table we can see that **AUC is highest for our Random Forest Model**, hence it explains the highest variation in our dependent variable.
* **Sensitivity, Specificity , Accuracy are also highest** for our **Random Forest Model.**
* The Error rate (from confusion matrix) is also lowest for our Random Forest model.
* The **difference in AUC on Test and Train Model for our RANDOM Forest Model is marginal and under 10%.** Hence the model is valid.
* The **KS is also highest for Random forest Model** – hence this model has the highest degree of separation among positive and negative distributions in the decile table.
* Even the concordance ratio (prob(right>prob(wrong) among pairs) is highest for Random Forest Model and discordance ratio (prob(right<prob(wrong) among pairs) is lowest.
* **Gini coefficient is also highest for random forest model,** hence this model can best differentiate the bad cases from good cases
* Therefore, we can conclude that our **Tuned Random Forest Model is the best model** that should be used **for predicting the prospective personal loan customers for Thera Bank.**