Capstone Project

Taiwan: Customer Defaults

Submitted By:

- Suprasanna Pradhan
  BABI Dec -2018

Table of Contents

Final Report

1. Introduction

We presented validate heuristic approach to mine potential default accounts in advance where a risk probability is precomputed from all previous data and the risk probability for recent transactions are computed as soon, they happen. Beside our heuristic approach, we also apply a recently proposed machine learning approach as a result, we find that these applied approaches outperform existing state-of-the-art approaches.

The achievement values are indeed to get the correct prediction score ,as we know there is one dependent and many independent variables with two or more classes and with continuous or discrete frequency .

At outset we have been trying to accomplish of Variable Identification, Univariate, Bivariate Analysis, Missing Values Imputation and Outliers Treatment in EDA part.

Further we have performed hypotheses for Pay status variable using chi square test to check their dependent relationship . In additional to this we have created some dummy variables and subsequently we have checked multicollinearity , KMO test and factor analysis before processing to final model preparation for our analysis .

2. EDA

The data set has 30000 observations with 25 variables in this dataset comprises of demographic variables (gender, education level, marriage status, and age) and financial variables of 6-months' worth of payment data from April 2005 to September 2005 (amount of given credit, monthly repayment statuses, monthly amount of bill statements, and monthly amount of previous payments).

Limit balance and age are continuous frequencies and other variable are bionomical categories . The Payment history from April to September2005

Variable X12 to X17 is bill statement amount ,all are carrying amount values X18 to X23 is carried out the previous amount paid by the customer shows in discrete frequency

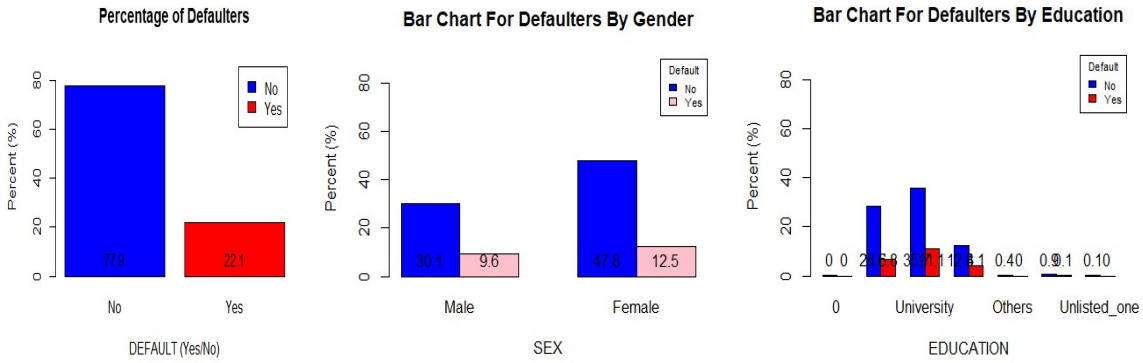
We have renamed the column names to identify easily the variable names

Header	Description	Column	Renamed -Column
X1	Limit Balance	LIMIT_BAL	LIMIT_BAL
X2	Gender	SEX	SEX
X3	Education of customer	EDUCATION	EDUCATION
X4	Married/unmarried/other	MARRIAGE	MARRIAGE
X5	Age of customer	AGE	AGE
X6	The repayment status in September, 2005	PAY_0	REPAY_SEP
X7	The repayment status in August, 2005	PAY_2	REPAY_AUG
X8	Payment delay for eight months	PAY_3	REPAY_8M
X9	Payment delay for nine months and above	PAY_4	REPAY_9M
X10	Payment delay for ten months and above	PAY_5	REPAY_10M
X11	The repayment status in April 2005. The measurement scale for the repayment status is: -1 = pay duly; 1 = payment delay for one month; 2 = payment delay for two months	PAY_6	REPAY_APR
X12	Amount of bill statement in September, 2005	BILL_AMT1	BILL_AMT_SEP
X13	Amount of bill statement in August, 2005	BILL_AMT2	BILL_AMT_AUG
X14	Amount of bill statement in July, 2005	BILL_AMT3	BILL_AMT_JUL
X15	Amount of bill statement in June, 2005	BILL_AMT4	BILL_AMT_JUN
X16	Amount of bill statement in May 2005	BILL_AMT5	BILL_AMT_MAY
X17	Amount of bill statement in April, 2005	BILL_AMT6	BILL_AMT_APR
X18	Amount paid in September, 2005	PAY_AMT1	PAID_AMT_SEP
X19	Amount paid in August, 2005	PAY_AMT2	PAID_AMT_AUG
X20	Amount paid in July , 2005	PAY_AMT3	PAID_AMT_JUL

X21	Amount paid in June, 2005	PAY_AMT4	PAID_AMT_JUN
X22	Amount paid in May, 2005	PAY_AMT5	PAID_AMT_MAY
X23	Amount paid in April, 2005	PAY_AMT6	PAID_AMT_APR
Y		default payment next month	DEFAULT

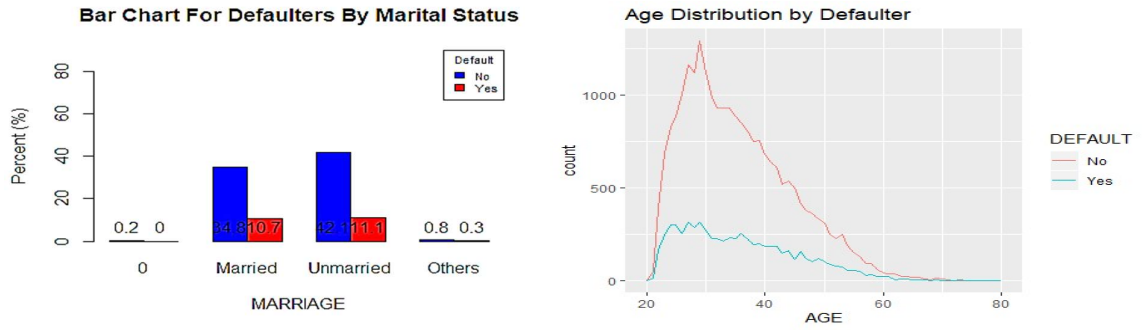
Key Observations :

- We observed that the ratio of defaulted customers is 22.12% and 77.9% are not default cases and found here Female customers are more into default category .
- The ratio shows male customer is 9.6% and female category shows 12.5% in defaulted category while we were checked gender wise the defaulters
- The customers who has been completed university level - graduate or PG is also found more into default side

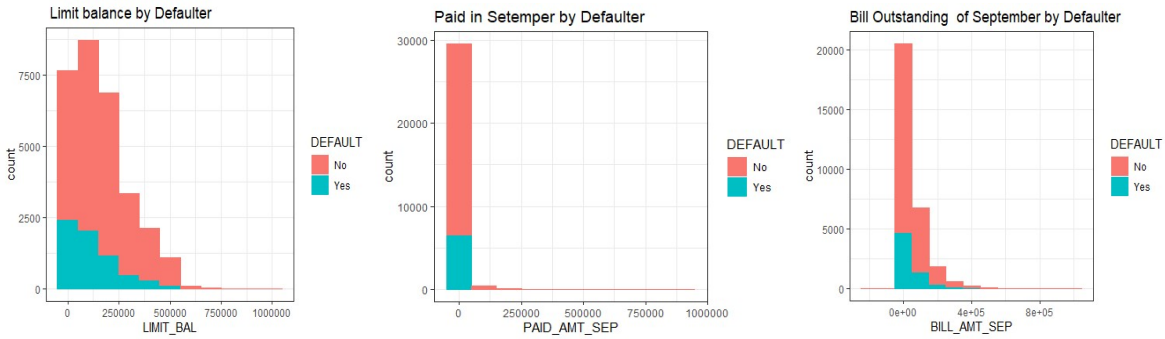


Single are paying the bills on time ,married customers somehow response is less comparing to unmarried customers

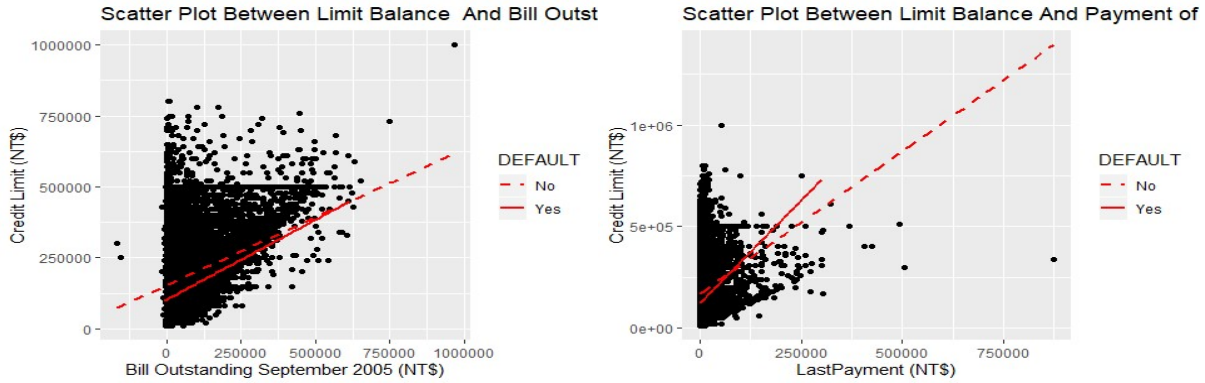
Average customers whose age category of 25 to 30 is shows highest risk .



Customers are having credit balance limit 250000 are maximum defaulters, the customers having bigger credit limit the ratio of defaulter less.



We have checked variable limit balance with outstanding bill amount and paid amount of September month where we found in the bill outstanding graph defaulter line shows long but in payment graph it has been tilted little upwards , hence we understand there is more customers in defaulter in month of September payment



Performed Chi-Square Test to check the null hypothesis whether PAY_STATUS variables are independent or not, if p-value less than 0.05 significance than we reject the null hypothesis that variables are independent.

We found the pay status variables are highly correlated to each other , checked only for April repay status randomly using chi . It looks like these pay status categorical variables are dependent on each other and impact of REPAY_ SEP to REPAY_APR variables to default. Payment DEFAULT is high.

```
Sep
## data:  REPAY_APR and REPAY_SEP
## X-squared = 26637, df = 90, p-value < 2.2e-16

Aug

## data:  REPAY_APR and REPAY_AUG
## X-squared = 29864, df = 90, p-value < 2.2e-16

Jul

## data:  REPAY_APR and REPAY_JUL
## X-squared = 51610, df = 90, p-value < 2.2e-16

Jun
```

```
## data:  REPAY_APR and REPAY_JUN
## X-squared = 81782, df = 90, p-value < 2.2e-16

May
## data:  REPAY_APR and REPAY_MAY
## X-squared = 114071, df = 81, p-value < 2.2e-16
```

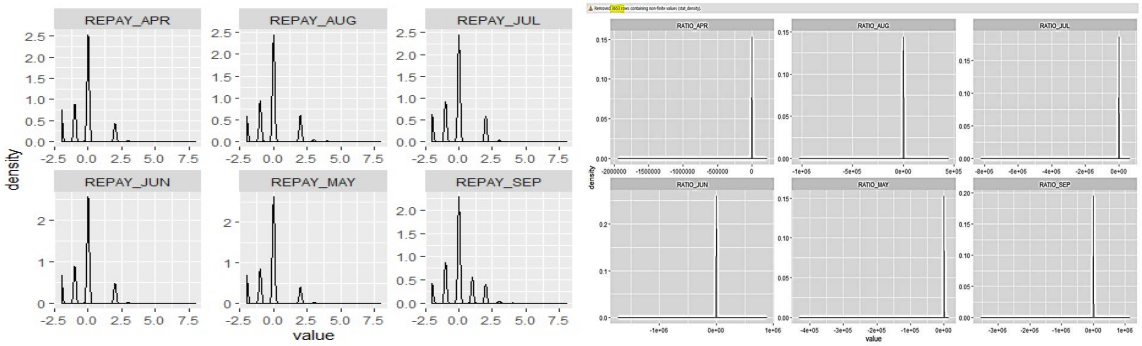
3. Data Cleaning and Pre-processing

These categorical variables with the levels from September to April REPAY_ SEP to REPAY_APR are having these classes -1=pay duly, 1=payment delay for one month, 2=payment delay for two months, . 8=payment delay for eight months, 9=payment delay for nine months and above.

Further we have checked only the customers greater than one is only late payer customers , we considered here to classify them as late payer or paid on time

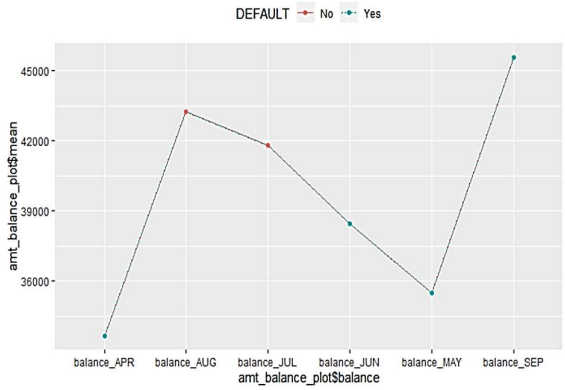
As an output of this calculation , we can classify Month of August , July and September the pay status is late paid the customer however in September month shows more compared to Aug and Jul.

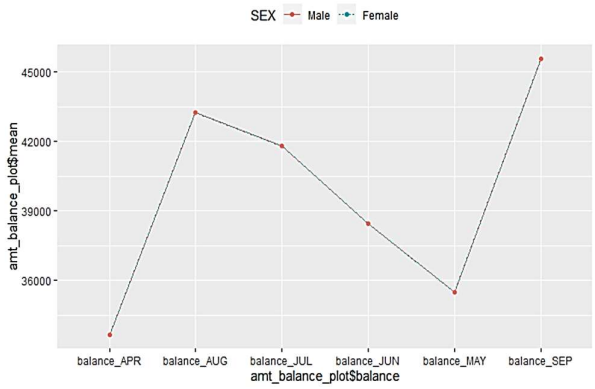
Also, we have added payment ratio a new dummy variable (outstanding bill amount /paid amount of each month)



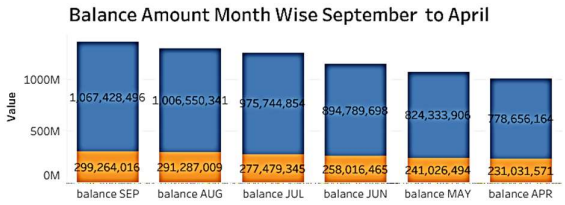
In connecting to above phase , we have added the balance amount new variable in our data set . The average of the same shows us ,month of Aug, Jul and Sep 2005 has been shown the maximum due amount which is not paid fully . The mean value of each month of the balance amount is increased continuously and for the month of September is very high while comparing with previous month.

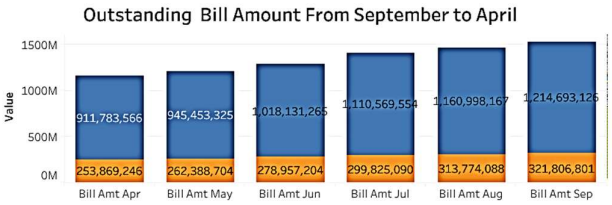
Observed more balance amount holders are male customers for whole trend September month is highest one.

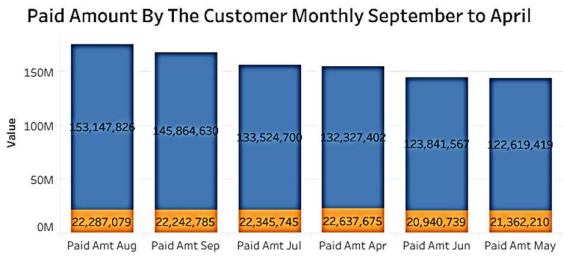


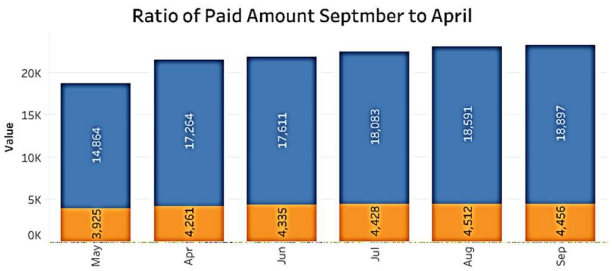


Bar chart of Balance of Amount/outstanding bill amount / paid amount /Ratio of paid amount









Further we have **converted here the RATIO_SEPT to RATIO_APR variable into positive and negative numeric values**

If Bill Amount paid <=100 : Then convert Ratio to positive , Impute NaN to 1 (Higher the ratio, lesser the chance of Default)

Month	Negative ratio"0"(amount paid less against the billed amount)	Positive ratio "1"(paid 100% of amount or More)
Sep	24834	3698
Aug	24307	3826
Jul	24177	3577
Jun	24143	3387
May	25543	1008
Apr	23091	3957

Classified again REPAY and create a dummy variable - timely paid or delayed in payment

The plot shows almost all month's tendency is very high with class "0", so it indicates moreover people are paid very minimal amount with revolving credit system. We have used the logic if the value fall less than zero then the customer is defaulted if is greater than zero than he is defaulted (-2 and -1 we have considered here that customer are paid on time other than that all 0 to 8, we have categorized as late payer)

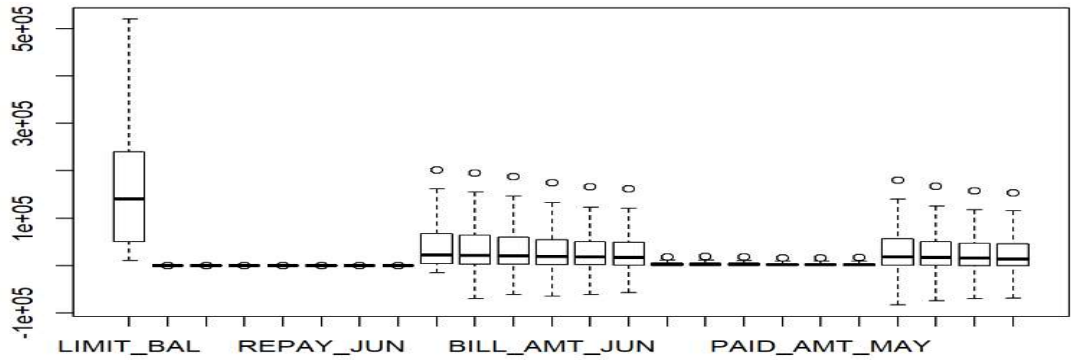
```
3688  2667   322    76    26    11     9    19

table(bank_data1$REPAY_AUG)
##    -2    -1     0     1     2     3     4     5     6     7     8
##  3782  6050 15730    28  3927   326    99    25    12    20     1
```

```
table(bank_data1$REPAY_JUL)
##    -2    -1     0     1     2     3     4     5     6     7     8
##  4085  5938 15764     4  3819   240    76    21    23    27     3

table(bank_data1$REPAY_JUN)
##    -2    -1     0     1     2     3     4     5     6     7     8
##  4348  5687 16455     2  3159   180    69    35     5    58     2

table(bank_data1$REPAY_MAY)
##    -2    -1     0     2     3     4     5     6     7     8
##  4546  5539 16947  2626   178    84    17     4    58     1

table(bank_data1$REPAY_APR)
##    -2    -1     0     2     3     4     5     6     7     8
##  4895  5740 16286  2766   184    49    13    19    46     2
```

We found in September22%, August 14%, July 14%, June 11%, May 09 and April it is 10% customer are paid on time

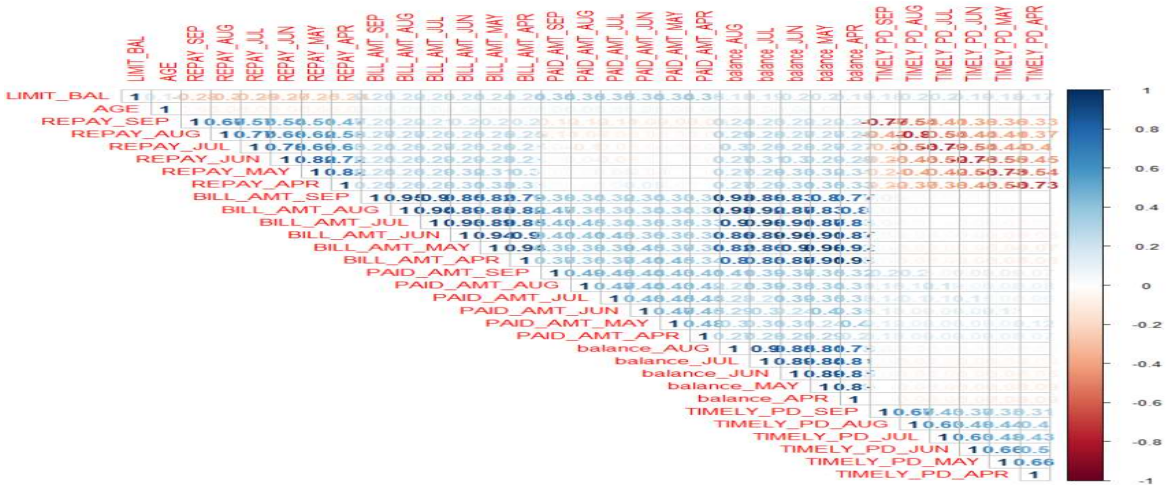
We found these are the variable are having some outliers, hence we have capped them within probability of (.05 to .95)

```
##  [1] "LIMIT_BAL"    "AGE"          "REPAY_SEP"    "REPAY_AUG"
##  [5] "REPAY_JUL"    "REPAY_JUN"    "REPAY_MAY"    "REPAY_APR"
##  [9] "BILL_AMT_SEP" "BILL_AMT_AUG" "BILL_AMT_JUL" "BILL_AMT_JUN"
## [13] "BILL_AMT_MAY" "BILL_AMT_APR" "PAID_AMT_SEP" "PAID_AMT_AUG"
## [17] "PAID_AMT_JUL" "PAID_AMT_JUN" "PAID_AMT_MAY" "PAID_AMT_APR"
## [21] "balance_SEP"  "balance_AUG"  "balance_JUL"  "balance_JUN"
## [25] "balance_MAY"  "balance_APR"
```



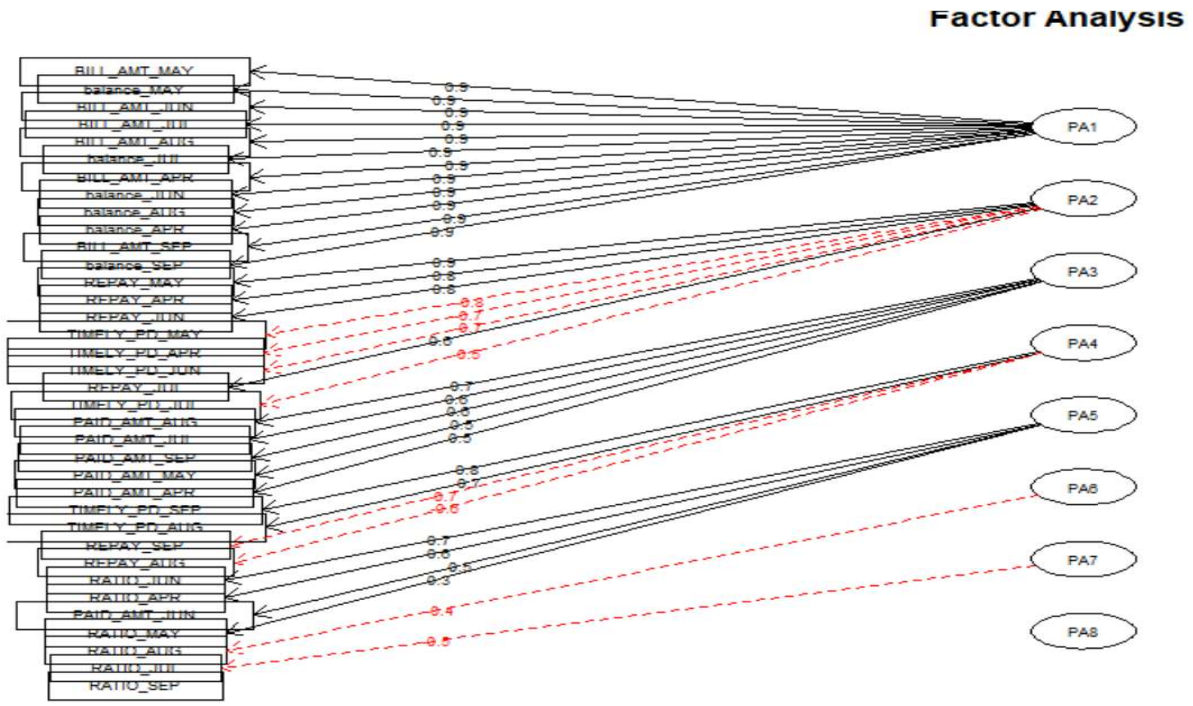
All NA are shown in the dummy variable of ratio, we understand this value are occurred due to the paid amount by the customer are not paid as the outstanding bill amount shown each month or very less against of outstanding billed amount .

Observed there is multicollinearity issue billed outstanding amounts, the dummy variables balance amount and paid on time or late paid are highly correlated ,we will have to take out these variables from the module while we are process the final module evaluation



We have also checked Kaiser-Meyer-Olkin (KMO) to find the Test for Sampling Adequacy whereas the values in this case is greater than .5 , hence the data set is occurred with enough samples .

Further We have performed FA variables , the output of FA 8 variables and 6 variables form the main data set , we have merged and considered the data set for the modelling processing further



```
  LIMIT_BAL SEX EDUCATION MARRIAGE AGE DEFAULT        PA1          PA2        PA3        PA4        PA5        PA6       PA7       PA8
1     20000   2         2        1  24       1 -0.65505048 -1.566069537 -0.5853008 -2.7396537 -0.28868694 -0.3336919 0.4391205 0.1522075
2    120000   2         2        2  26       1 -0.85452952  1.053568856 -0.4161835  0.1636740 -0.03377417 0.4646900 -0.1320396 1.4610357
3     90000   2         2        2  34       0 -0.52269375  0.096085124 -0.3000248  0.4873752 -0.43826364 0.4397578 0.2643110 0.7402958
4     50000   2         2        1  37       0 -0.04871894 -0.002544749 -0.4353623  0.5266916 -0.39476438 0.4034434 0.5480870 0.5670916
5     50000   1         2        1  57       0 -0.85332389  0.266568820  1.9285488 -0.1913143 -0.19966378 -1.9722762 0.8661627 1.0896000
6     50000   1         1        2  37       0 -0.08186224 -0.065224365 -0.3937278  0.5364438 -0.32483433 0.6391223 0.9395406 0.7666797
'data.frame':   30000 obs. of  14 variables:
 $ LIMIT_BAL      : num  20000 120000 90000 50000 50000 50000 500000 100000 140000 20000 ...
 $ SEX            : int  2 2 2 2 1 1 1 2 2 1 ...
 $ EDUCATION      : int  2 2 2 2 2 1 1 2 3 3 ...
 $ MARRIAGE       : int  1 2 2 1 1 2 2 2 1 2 ...
 $ AGE            : int  24 26 34 37 57 37 29 23 28 35 ...
 $ DEFAULT        : int  1 1 0 0 0 0 0 0 0 0 ...
 $ BILLED _AMT    : num  -0.6551 -0.8545 -0.5227 -0.0487 -0.8533 ...
 $ REPAY_STATUS   : num  -1.56607 1.05357 0.09609 -0.00254 0.26657 ...
 $ PAID_AMT       : num  -0.585 -0.416 -0.3 -0.435 1.929 ...
 $ TIMELY_PAID_AMT: num  -2.74 0.164 0.487 0.527 -0.191 ...
 $ RATIO_PADI_AMT1: num  -0.2887 -0.0338 -0.4383 -0.3948 -0.1997 ...
 $ RATIO_PADI_AMT2: num  -0.334 0.465 0.44 0.403 -1.972 ...
 $ RATIO_PADI_AMT3: num  0.439 -0.132 0.264 0.548 0.866 ...
 $ RATIO_PADI_AMT4: num  0.152 1.461 0.74 0.567 1.09 ...
```

4. Model building

We should be looking into stratified sampling since the proportions class “yes” or ” no “the class imbalance problem is caused by there not being enough patterns belonging to the minority class, not by the ratio of positive and negative patterns itself per se. so let us randomizing the samples with proper balance for both class

```
nrow(subset(train_new, DEFAULT == 1))/nrow(train_new)
## [1] 0.5012372
nrow(subset(test_data, DEFAULT == 1))/nrow(test_data)
## [1] 0.2196667
table(train_new$DEFAULT)
##
##    0    1
## 4636 4659

str(train_bank)
## 'data.frame':    9291 obs. of  14 variables:
##  $ LIMIT_BAL      : num  20000 450000 100000 30000 20000 20000 270000 50000
120000 200000 ...
##  $ SEX            : int  1 1 1 2 2 1 2 1 1 1 ...
##  $ EDUCATION      : int  2 1 2 2 2 2 2 3 2 2 ...
##  $ MARRIAGE       : int  2 1 1 2 1 1 2 1 2 1 ...
##  $ AGE            : int  33 45 30 22 24 31 26 53 28 32 ...
##  $ DEFAULT        : int  0 1 1 1 1 1 1 0 0 1 ...
##  $ BILLED._AMT    : num  -0.391 -0.492 0.455 -0.208 -0.647 ...
##  $ REPAY_STATUS.  : num  0.1827 0.3363 0.0201 -0.2731 2.5181 ...
##  $ PAID_AMT       : num  -0.7188 1.278 -0.0873 -0.4301 -0.5607 ...
##  $ TIMELY_PAID_AMT: num  0.543 0.523 0.431 -1.422 -1.695 ...
##  $ RATIO_PADI_AMT1: num  0.419 -0.546 -0.411 -0.44 -0.027 ...
##  $ RATIO_PADI_AMT2: num  0.37743 -2.66432 0.00851 -0.35402 -0.03372 ...
## $ RATIO_PADI_AMT3: num  0.3778 1.6226 -0.1 -0.0637 0.1059 ...
## $ RATIO_PADI_AMT4: num  0.62 0.416 0.325 0.922 0.141 ...
```

After doing FA ,we have been considered above 14 variables for the final model building .

4.1 Logistic Regression

Since the dependent variable is binary and we need check the multicollinearity of the variable before preparing the final model of logistic regression. We found six variables are carried out good coefficient and four of them are negative coefficient ,only two predictors show positive impact . Further we expected the other model may perform better

Here we have Performed the initial regression and later on we have redefined the data set after doing these steps

1. Find out all significant variables.
2. Remove non-performing variables from the module
3. Check multicollinearity with VIF function and rebuild the module

```
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -1.346e+00  1.516e-01  -8.881  < 2e-16 ***
## LIMIT_BAL       -1.199e-06  2.770e-07  -4.329 1.50e-05 ***
## SEX             -1.252e-01  5.754e-02  -2.176  0.02955 *
## AGE              7.655e-03  3.023e-03   2.532  0.01134 *
## REPAY_STATUS.    4.632e-01  2.707e-02  17.112  < 2e-16 ***
## PAID_AMT        -3.757e-01  4.169e-02  -9.011  < 2e-16 ***
## TIMELY_PAID_AMT -7.051e-01  2.598e-02 -27.136  < 2e-16 ***
## RATIO_PADI_AMT1 -9.419e-02  3.626e-02  -2.598  0.00939 **
## RATIO_PADI_AMT4 -1.508e-01  3.003e-02  -5.021 5.13e-07 ***
```

Besides REPAY_STATUS all variables are has Negative Impact ,in other words, we can see anti-incumbency implication

Check Multi-Collinearity Effect:

```
##     LIMIT_BAL         SEX         AGE   REPAY_STATUS.
##      1.256936    1.010270    1.031040    1.111527
##      PAID_AMT TIMELY_PAID_AMT RATIO_PADI_AMT1 RATIO_PADI_AMT4
##      1.096350    1.018324    1.038484    1.024806
```

No variables are having value more than five hence , there is no multi-collinearity but as can be seen, VIF is just slightly greater than 1, hence we can Conclude that our variables are moderately correlated

4.2 Classification and Regression Tree

CART method has enabled us to determine the complex interactions among variables in the final tree, in contrast to identifying and defining the interactions in a multivariable logistic regression model.

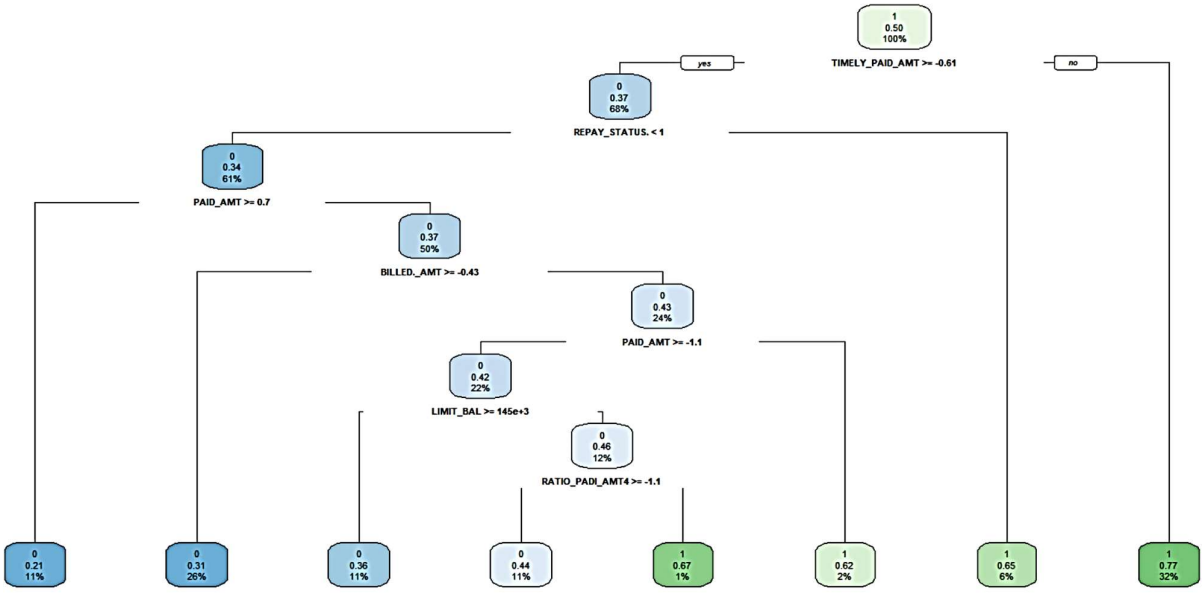
We did Pruning of the tree is using only seven Variables. TIMELY_PAID_AMT, REPAY_STATUS , PAID_AMT, BILLED._AMT , LIMIT_BAL, RATIO_PADI_AMT1

We have checked these are the variables are having highly impact to predict Y

CT_model$variable.importance

```
## TIMELY_PAID_AMT    REPAY_STATUS.         PAID_AMT      BILLED._AMT
##      806.215602       382.899461       194.203815       164.608151
## RATIO_PADI_AMT4 RATIO_PADI_AMT2 RATIO_PADI_AMT1 RATIO_PADI_AMT3
##      152.255511       125.071627       109.634039        74.565264
```

```
##        LIMIT_BAL              AGE        EDUCATION         MARRIAGE
##        68.850349        30.739832         7.163986         5.436529
```



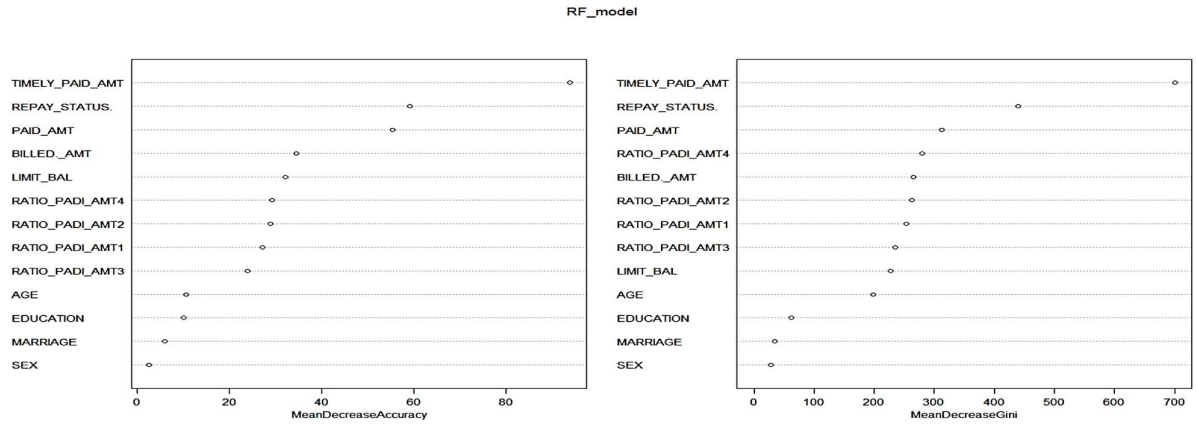
4.3 Random Forest:

RF we have used because its tree-based strategies naturally it ranks by how well the model improve the purity of the node. This mean decrease in impurity over all trees (called Gini impurity)and It reduces the complexity of a model and makes it easier to interpret.

```
##  randomForest(formula = DEFAULT ~ ., data = train_bank, mtry = 3,      nodesize = 10, ntree = 501, importance = TRUE)
##        OOB estimate of  error rate: 30.09%
## Confusion matrix:
##     0    1 class.error
## 0 3591 1052   0.2265776
 # 1 1744 2904   0.3752151
```

The Out-of-Bag Estimate of Error Rate for our given Random Forest in our case is 30.09% , so we realized this model may not help of predict well . Further we found these variable is playing important role to predict Y

```
##                      0     1 MeanDecreaseAccuracy MeanDecreaseGini
## TIMELY_PAID_AMT 76.59 41.05                 93.85           700.48
## REPAY_STATUS.   41.56 23.38                 59.15           439.35
## PAID_AMT        18.16 42.96                 55.36           312.82
```



Machine learning approach with Ensemble Methods

4.4 KNN Classifier:

K nearest neighbors is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions). KNN has been used in statistical estimation and pattern recognition , it is a non-parametric model which means that it does not make any assumptions about the data set. It finds intense application in pattern recognition, data mining and intrusion detection

Choosing the optimal value for K is best done by first inspecting the data. In general, a large K value is more precise as it reduces the overall noise but there is no guarantee. Cross-validation is another way to retrospectively determine a good K value by using an independent dataset to validate the K value. Historically, the optimal K for most datasets has been between 3-10. That produces much better results than 1NN.

We have used K=5 to predict the test data set

```
knn_fit<- knn(train = train_bank[,-6], test = test_bank[,-c(6,15)], cl=train_bank$D
EFAULT,k =5,prob=TRUE)
knn_chk= table(test_bank$DEFAULT,knn_fit)
knn_chk
```

4.5 Naive Bayes:

We have used Naive Bayes classifiers because Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other.

Naive Bayes learners and classifiers can be extremely fast compared to more sophisticated methods. The decoupling of the class conditional feature distributions means that each distribution can be independently estimated as a one-dimensional distribution. This in turn helps to alleviate problems stemming from the curse of dimensionality.

```
NB = naiveBayes(x =train_bank[-6], y =train_bank$DEFAULT)
pred.NB = predict(NB, newdata =test_bank[-6])
pred.NB

tab.NB =table(test_bank[,6], pred.NB)
tab.NB
```

The final output shows that we built a Naive Bayes classifier that can predict whether customer can be defaulted with an accuracy of approximately 70%.

The model observed to perform decent on majority of the model performance measures, indicating it to be a good model.

4.6 Bagging:

Bootstrapping is a sampling technique in which we create subsets of observations from the original dataset, **with replacement**. The size of the subsets is the same as the size of the original set.

Bagging (or Bootstrap Aggregating) technique uses these subsets (bags) to get a fair idea of the distribution (complete set). The size of subsets created for bagging may be less than the original set.

- Multiple subsets are created from the original dataset, selecting observations with replacement.
- A base model (weak model) is created on each of these subsets.
- The models run in parallel and are independent of each other.
- The final predictions are determined by combining the predictions from all the models.

```
bank_bagging <- bagging(DEFAULT ~.,data=train_bank,
                        control=rpart.control(maxdepth=5, minsplit=4))

pred_class <- predict(bank_bagging, test_bank)
```

4.7 XGBoost

XGBoost (extreme Gradient Boosting) is an advanced implementation of the gradient boosting algorithm. XGBoost has proved to be a highly effective ML algorithm, extensively used in machine learning competitions and hackathons. XGBoost has high predictive power and is almost 10 times faster than the other gradient boosting techniques. It also includes a variety of regularization which reduces overfitting and improves overall performance. Hence it is also known as 'regularized boosting' technique.

XGBoost is comparatively better than other techniques:

- Regularization: Standard GBM implementation has no regularization like XGBoost. Thus, XGBoost also helps to reduce overfitting.
- Parallel Processing: XGBoost implements parallel processing and is faster than GBM .XGBoost also supports implementation on Hadoop.

- High Flexibility: XGBoost allows users to define custom optimization objectives and evaluation criteria adding a whole new dimension to the model.
- Handling Missing Values: XGBoost has an in-built routine to handle missing values.
- Tree Pruning: XGBoost makes splits up to the adept specified and then starts pruning the tree backwards and removes splits beyond which there is no positive gain.
- Built-in Cross-Validation: XGBoost allows a user to run a cross-validation at each iteration of the boosting process and thus it is easy to get the exact optimum number of boosting iterations in a single run.

```
classifier = xgboost(data = as.matrix(train_bank[,-6]), label = train_bank$DEFAULT,
nrounds = 10)

## [1]  train-rmse:0.843655
## [2]  train-rmse:0.666488
## [3]  train-rmse:0.557688
## [4]  train-rmse:0.492921
## [5]  train-rmse:0.456142
## [6]  train-rmse:0.436011
## [7]  train-rmse:0.423342
## [8]  train-rmse:0.415939
## [9]  train-rmse:0.411865
## [10] train-rmse:0.408762
```

4.8 K-Fold Cross Validation

We know ,in the K-fold cross-validation method, the dataset is divided into k subsets, and the holdout method is repeated k times. Each time, one of the k subsets is used as the test set and the remaining k 1 subsets are put together to form a training set. The average error across all k trials is then calculated.

The advantage of this method is that every data point has one chance to be in a test set exactly once and has the chance to be in a training set k 1 times.

The variance of the resulting estimate is reduced as k is increased.

The disadvantage of this method is that it suffers from heavy computational complexity, because the training algorithm has to be rerun from scratch k times, so it takes k times as much computation to make an evaluation.

For our data set we have created 12 CV folds

```
folds_bank = createFolds(train_bank$DEFAULT, k =12)
cv = lapply(folds_bank, function(x) {
tr_fold = train_bank[-x, ]
tt_fold = test_bank[x, ]
classifier = xgboost(data = as.matrix(train_bank[-6]), label = train_bank$DEFAULT,
nrounds = 10)
y_pred = predict(classifier, newdata = as.matrix(tt_fold[-c(6,15)]))
y_pred = (y_pred >= 0.5)
cmx= table(tt_fold[,6], y_pred)
```

5. Model validation

We have performed following model performance measures to calculate on entire data set to gauge the goodness of the model for Logistic regression ,CART and Random Forest

- KS
- Area Under Curve (AUC)
- Gini Coefficient
- Classification Error

Accuracy =(6672+646)/(6672+646+1342+340)= 82%

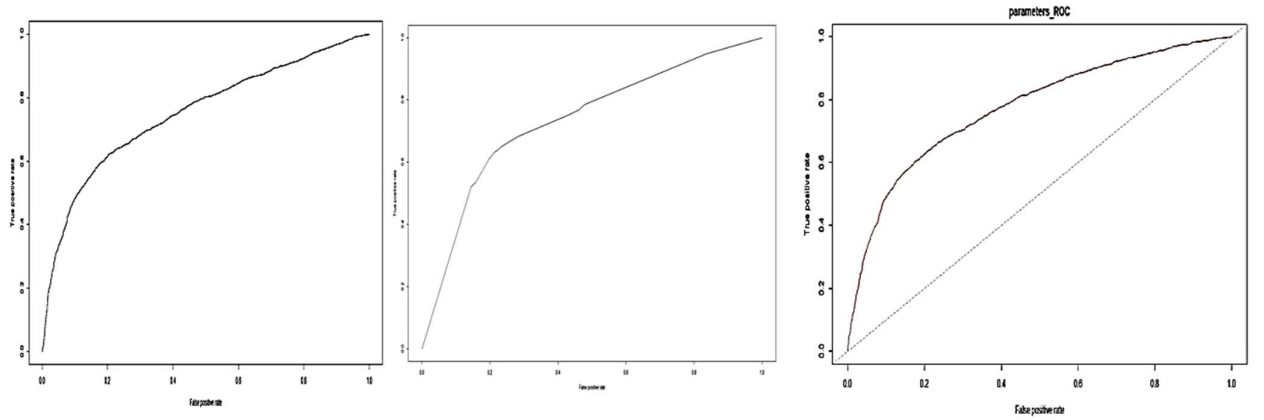
Classification Error Rate = 1- Accuracy = 18%

The lower the classification error rate, higher the model accuracy, resulting in a better model.

Machine learning models measured and validated by only checking with accuracy

We have also performed rank ordering to validate Random forest model along with KS, Gini , Accuracy

	deciles	cnt	cnt_resp	cnt_non_resp	rrate	cum_resp	cum_non_resp	cum_rel_resp	cum_rel_non_resp	ks
1	1	885	49	836	5.5%	1988	7012	100.0%	100.0%	0.0000
2	9	894	417	477	46.6%	1017	783	51.2%	11.2%	0.3999
3	8	909	243	666	26.7%	1260	1449	63.4%	20.7%	0.4272
4	7	898	161	737	17.9%	1421	2186	71.5%	31.2%	0.4030
5	10	906	600	306	66.2%	600	306	30.2%	4.4%	0.2582
6	6	929	152	777	16.4%	1573	2963	79.1%	42.3%	0.3686
7	5	904	119	785	13.2%	1692	3748	85.1%	53.4%	0.3166
8	4	885	98	787	11.1%	1790	4535	90.0%	64.7%	0.2537
9	3	902	79	823	8.8%	1869	5358	94.0%	76.4%	0.1760
10	2	888	70	818	7.9%	1939	6176	97.5%	88.1%	0.0946



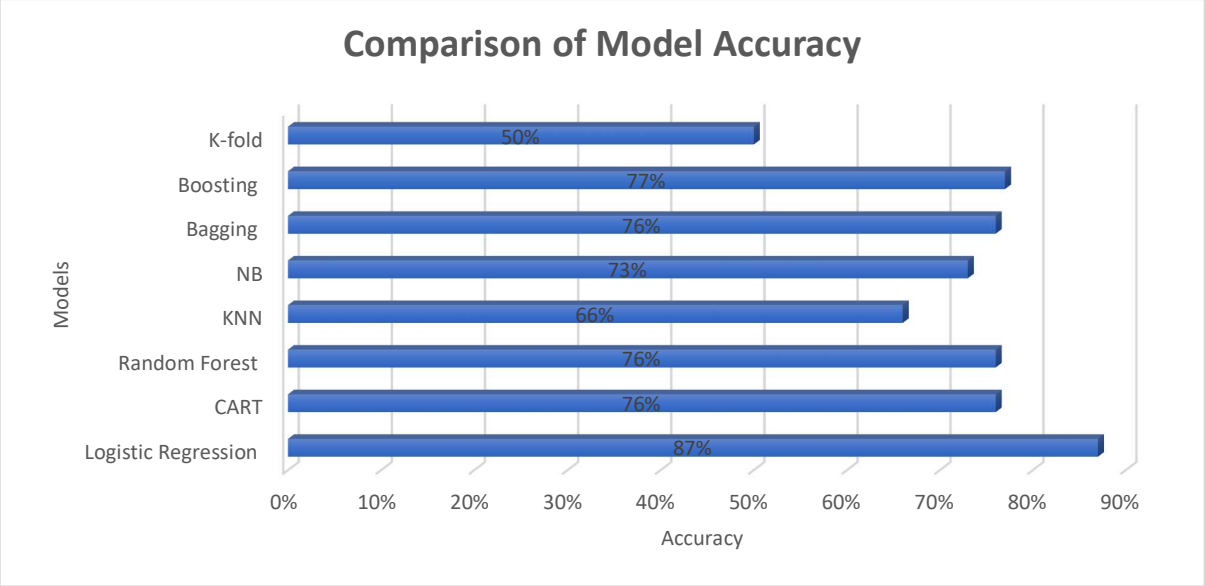
Summary of comparison of Logistic regression ,CART and Random Forest

Models	KS	AUC	GINI	Accuracy	Classification Error
Logistic Regression	41%	75%	14%	87%	18%
CART	41%	73%	14%	76%	19%
Random Forest	42%	77%	30%	76%	24%

Accuracy of Machine learning Models

Accuracy. KNN	66%
Accuracy. NB	73%
Accuracy. Bagging	76%

Accuracy. Boosting	77%
Accuracy .K Fold_ Cross Validation	50%



6. Final interpretation / recommendation

- The best models are **Logistic Regression** , by which we predicted with **87 % accuracy**, whether a customer is likely to default next month. Whereas **XGBOOST method can 77% accuracy .**
- The strongest predictors of default are the **PAY_X (i.e. the repayment status in previous months), the LIMIT_BAL & the PAY_AMTX (amount paid in previous months)**.
- We have seen also that being Female, more educated, Single and between 30-40years old means a customer is more likely to make payments on time.
- Married customer in average age of 25 to 30 is shown highest risk , may lean to be a defaulter at some point of time.

"Thank You"