19/11/2021

Credit Risk Analysis

Internal Credit Scoring model

Submitted by: Aryan Singh Rajawat,
B.SC. Actuarial Science and Quantitative Finance
Roll NO.410

Abstract

As per the new regulations across the banking industry in the credit management sector. all banks are required to either develop an internal credit scoring model or to adapt the existing one. our bank has decided to go with the former.
the desired output was binary. that is yes(the obligor will default) and no(the obligor will not default). and considering the nature of the output, we had various possible frameworks that we could have worked with, namely Logistic regression, Decision tree, Random forest, Naive Bayes classifier and Support vector machine.
In our model building process, we decided to assess three framework

- logistic Regression
- Decision Tree
- Random Forest

And after evaluating the model under the lense of various robust statistical tests. we decided to use the random forest framework.

Introduction

The recent significant increase in NPAs across the banking industry has led the regulator to impose a new regulatory requirement for credit risk management.
As per the new requirement, all banks are expected to adopt one of the following approaches to enhance their currently employed credit risk management processes:

- Use a standardized framework designed by the regulator (“Standardized Approach”)
- Design, develop and implement robust internal credit risk models for loan underwriting and credit risk provisioning. (“Internal Model Approach”)

In light of the new regulatory requirement, the Credit Risk Board Committee of our bank has decided to go ahead with the internal model approach for credit risk assessment.

Methodology

- After collecting importing the raw data and our initial Exploratory data analysis.
- We found that the variables had numerical, integer and character data types.
- Changed the variable amount name
- Transformed character variables into factors
- Conducted mean imputation on the variable years_at_residence
- After checking the levels, we observed the presence of an unknown level in variable checking_balance and savings_balance.
- converted unknown values to NA values
- Applied MICE imputation on the variable to replace missing values with imputed values.
- Split the dataset into train and testing datasets using stratified sampling
- Iteration 1 of logistic regression framework
- Upon obtaining the summary we decided to remove all the insignificant variables and did 2 iteration
- Assessed the model with various statistical tests.
- Used test dataset to predict the values using model
- Obtained confusion Matrix and Roc curves
- Did the k fold cross-validation on the same model
- Obtained confusion matrix and ROC curves
- We moved to the second framework, that is decision trees
- Developed the model using part
- Obtained confusion matrix
- We moved to the final machine learning algorithm, random forest
- Developed the random forest model
- Obtained confusion matric and calculate ROCR curve value

Exploratory Data Analysis and Data cleaning/transformation

After creating a working directory and importing the raw data from it. we did some exploratory data analysis. :-

- we used the describe function to find the proportion of values in each variable.
- Executed the dim function to find the dimension of the dataset which was 6000 rows and 19 columns.
- And str function to find the datatype of variables in the dataset. found that 10 of the 19 variables had character datatype. namely 'checking_balance', "credit_history", "savings_balance", "employment_duration", "other_credit", "housing", "job", "phone", "percent_of_income", "dependants", "purpose"
- Changed the name of the column amount.USD. to amount
- Converted the variable default from character values "yes" (indicating that the obligor default) into 1 and "no"(indicating that the obligor do not default) into 0
- After identifying the variables with character datatype we decided to convert the variables into factor variables as it is required for the model building process since some of the frameworks that we will be using do not take character datatypes into account.
- we used lapply function to convert all the character datatypes into factor datatype.
- Checked for missing values and found that there are 1320 missing values in the dataset. and upon checking the summary function we found that there the variable years_at_residence had those missing values.
- So we decided to do the mean imputation on the variable since it was the most appropriate method.
- We checked the levels of all the variables. and found out that there is a level named "unknown" in the variable checking_balance and savings_balance.
- This does not make sense because it will have no real value but a factorial value and thus have a non-intuitional impact on the model.

Exploratory Data Analysis and Data cleaning/transformation

- Therefore we decided to perform Multivariate imputation by chained Equations using the library mice.
- We used the method polyreg
- Next we checked the proportion of defaults in the dataset and each variable.
- Finally we split the dataset into training and testing dataset using stratified sampling and maintaining the proportion of 70% and 30% in the splited dataset as it was the same in the original dataset

Now that the data is clean and transformed into what we have desired we can move on to building models using various frameworks and tests, compare them to obtain the most appropriate model.

```
> str(cra_data)
'data.frame':   6000 obs. of  19 variables:
 $ cust_id             : int  6252029 5110070 2846491 9264318 9412980 6111903 161
 $ acc_no              : num  6.25e+11 5.11e+11 2.85e+11 9.26e+11 9.41e+11 ...
 $ checking_balance    : Factor w/ 4 levels "< 0 USD","> 200 USD",..: 1 3 1 3 1 1
 $ months_loan_duration: int  12 36 11 15 10 14 24 18 24 30 ...
 $ credit_history      : Factor w/ 5 levels "critical","good",..: 2 2 1 2 2 2 1 1
 $ purpose             : Factor w/ 6 levels "business","car",..: 2 2 2 6 5 2 2 5
 $ amount              : int  1274 12389 3939 1308 1924 3973 6615 2124 11938 2406
 $ savings_balance     : Factor w/ 5 levels "< 100 USD","> 1000 USD",..: 1 1 1 1
 $ employment_duration : Factor w/ 5 levels "< 1 year","> 7 years",..: 1 3 3 2 3
 $ percent_of_income   : Factor w/ 4 levels "1","2","3","4": 3 1 1 4 1 1 2 4 2 4
 $ years_at_residence  : num  1 4 2 4 4 ...
 $ age                 : int  37 37 40 38 38 22 75 24 39 23 ...
 $ other_credit        : Factor w/ 3 levels "bank","none",..: 2 2 2 2 2 2 2 2 2 2
 $ housing             : Factor w/ 3 levels "other","own",..: 2 1 2 2 2 1 1 3 2 3
 $ existing_loans_count: int  1 1 2 2 1 1 2 2 2 1 ...
 $ job                 : Factor w/ 4 levels "management","skilled",..: 4 2 4 4 2
 $ dependants          : Factor w/ 2 levels "1","2": 1 1 2 1 1 1 1 1 2 1 ...
 $ phone               : Factor w/ 2 levels "no","yes": 1 2 1 1 2 1 2 1 2 1 ...
 $ default             : Factor w/ 2 levels "0","1": 2 2 1 1 1 1 1 2 2 2 ...
```

Framework

The selection of the framework ideally depends on two things.

1. our objective
2. the type of data we are working with

Here in our case, our objective was to develop a model that upon providing the necessary input would predict whether the obligor would default or not.
And since our desired output was binary, we had plenty of potential frameworkthat were available at our disposal. namely Logistic regression, Decision tree, Random forest, Naive Bayes classifier and Support vector machine. however, we divided to use the following framework.

1. Logistic regression
2. Decision tree
3. Random forest

Let us see in detail what were the results from each model.

Framework 1 : Logistic Regression

Logistic regression is a statistical analysis method used to predict a data value based on prior observations of a data set. A logistic regression model predicts a dependent data variable by analyzing the relationship between one or more existing independent variables.
The resulting analytical model can take into consideration multiple input criteria. In the case of credit risk analysis, the model could consider factors such as the loan purpose, previous credit history etc. Based on historical data about earlier outcomes involving the same input criteria, it then scores new cases on their probability of falling into a particular outcome category.

Methodology

Iteration 1

nitially, we started with our first iteration by using the glm function, taking into account all the given variables and by regressing them against the dependent variable that is the default.

```
#iteration 1 using logistic regression
model1 = glm(default~., data = train, family = binomial)
```

Inference

And after obtaining the summary for the first iteration we checked the summary of the model and found out that at 5% level of significance only checking_balance, months_loan_duration, credit_history, purpose, amount, savings_balance, employment_duration, percent_of_income, age, other_credit, housing, existing_loan_count and phone are significant.
The residual deviance of our first iteration is still less than the null deviance which shows that our model explains the data well

Iteration 2

In our second iteration, we only considered the significant variables from the first iteration.

```
#iteration 2 tking only statistically significant variables
model3 = glm(default~.-cust_id -acc_no -years_at_residence -job -dependants,
             data = train, family = binomial)
```

Inference

After obtaining the summary of the model, we observe that the null deviance is still higher than the residual deviance which shows that the model is a good fit for the given data.

The AIC value of the second model is lower than the first, which shows that this model is better than the former.

```
    Null deviance: 5131.3  on 4199  degrees of freedom
Residual deviance: 4260.0  on 4169  degrees of freedom
AIC: 4322
```

Statistical testing

We conduct various statistical tests to test the underlying assumptions of the model.

1 Test for multicollinearity
we used the VIF(variance inflation factor) function in r from the car library to check for multicollinearity and we will conclude that there is a presence of multicollinearity if the vif crosses the conservative threshold of 2.5.

```
> vif(model3)
                        GVIF Df GVIF^(1/(2*Df))
checking_balance    1.216786  2         1.050276
months_loan_duration 1.896981  1         1.377309
credit_history      2.055527  4         1.094247
purpose             1.538162  5         1.043999
amount              2.278682  1         1.509530
savings_balance     1.193760  3         1.029958
employment_duration 1.512849  4         1.053112
percent_of_income   1.369050  3         1.053748
age                 1.433615  1         1.197337
other_credit        1.198209  2         1.046244
housing             1.345260  2         1.076965
existing_loans_count 1.613551  1         1.270256
phone               1.192540  1         1.092035
```

Inference

none of the variables has a vif of more than 2.5 hence there is no presence of multicollinearity

2. Likelihood ratio test

we will use The likelihood ratio test to compare the likelihood of data under the full model against the likelihood of data under a reduced model. This will tell us how good of the fit the model is for the given data

we will use the lrtest function from the lmtest library

```
> lrtest(model3)
Likelihood ratio test

Model 1: default ~ (cust_id + acc_no + checking_balance + months_loan_duration +
    credit_history + purpose + amount + savings_balance + employment_duration +
    percent_of_income + years_at_residence + age + other_credit +
    housing + existing_loans_count + job + dependants + phone) -
    cust_id - acc_no - years_at_residence - job - dependants
Model 2: default ~ 1
  #Df  LogLik  Df  Chisq Pr(>Chisq)
1  31 -2130.0
2   1 -2565.6 -30 871.31  < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Inference

p value = 2.2e-16

Since the pvalue is less than 0.05 we will reject Ho and conclude that the model is a good fit

3. pseudo **R2 test**

we will also use pseudo R^2 test to find how good is the model predictive the power we will use the pr2 function from the pscl library.

```
fitting null model for pseudo-r2
          llh       llhNull            G2      McFadden          r2ML          r2CU
-2129.9774515 -2565.6300686   871.3052343     0.1698034     0.1873491     0.2656380
```

Inference

The value of McFadden's pseudo R^2 is 0.1698034. although a value between 0.2 to 0.4 is considered excellent but even this value is indicating that the model is a good fit to some extent

4. Goodness of fit test Hosmer Lemeshow test

Hypotheses:

H0 : β 1 = β 2 = … = β p = 0

H1 : At least one coefficient is not zero

Reject H0 p value < 0 05

```
> hoslem.test(train$default,fitted(model3))

        Hosmer and Lemeshow goodness of fit (GOF) test

data:  train$default, fitted(model3)
X-squared = 4200, df = 8, p-value < 2.2e-16
```

Since p value< 0 05 we reject Ho and conclude that the model is a good fit

5. Goodness of fit test Somers' D

Hypotheses:

H0 : β 1 = β 2 = … = β p = 0

H1 : At least one coefficient is not zero

Test statistic:

Somers' D =

(Concordant Percent Discordant Percent) / 100

Reject:

H 0 p value <0 05

```
> somersD(train$default,fitted(model3))
[1] 0.5488673
```

Somer's D value suggests that the model has a decent predictive ability

Using test dataset to predict

We used the test dataset to predict the value of our dependent variable as part of our cross-validation using the predict function in r.

```
#Predicting values using the test data set
pred = predict(model3,data= test, type = 'response')
```

Now we obtain the optimal cut off probability by using the optimal function in r to find the probability at which the accuracy of our model is maximum.

```
#find optimal cutoff probability to use to maximize accuracy
optimal <- optimalCutoff(test$default, pred)
```

which cam out to be about 0.8923516
and then we segregated the data accordingly using the ifelse function

```
#segrigating data
fit_class = ifelse(pred>0.8918246,1,0)
```

Confusion matrix

we use the confusion matrix to test the accuracy of the model and find its sensitivity, specificity,

Confusion matrix

we use the confusion matrix to test the accuracy of the model and find its sensitivity, specificity,

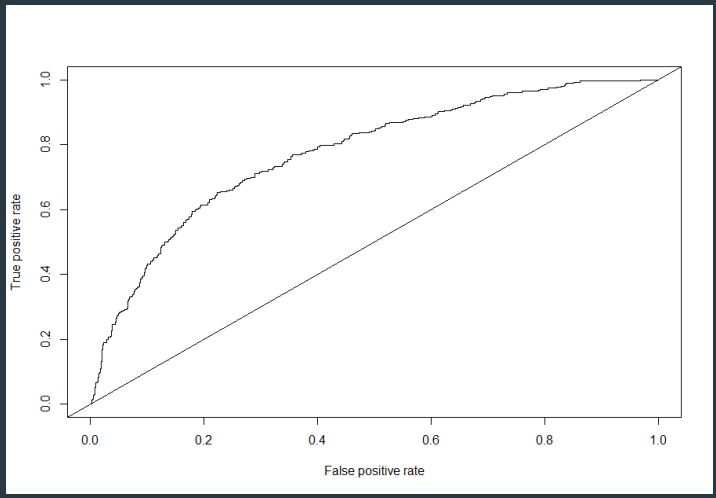
```
> caret::confusionMatrix(data=as.factor(fit_cla
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 2934 1260
         1    6    0

               Accuracy : 0.6986
                 95% CI : (0.6844, 0.7124)
    No Information Rate : 0.7
    P-Value [Acc > NIR] : 0.5875

                  Kappa : -0.0029

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.9980
            Specificity : 0.0000
         Pos Pred Value : 0.6996
         Neg Pred Value : 0.0000
             Prevalence : 0.7000
         Detection Rate : 0.6986
   Detection Prevalence : 0.9986
      Balanced Accuracy : 0.4990

       'Positive' Class : 0
```

As we can observe the accuracy isn't very promising. but the main objective of our building this model is to minimize the default rates in this time where the NPA's have increased drastically. so we should lay our focus on sensitivity. since sensitivity gives us the ratio between true positive and the sum of true positive and false negative. and we would rather focus on having more true positives than having more false-negative meaning we would like to give loans to more non-defaulters than defaulters.
we got a great sensitivity of 99.8%

$$\text{Sensitivity} = \frac{TP}{TP + FN}$$

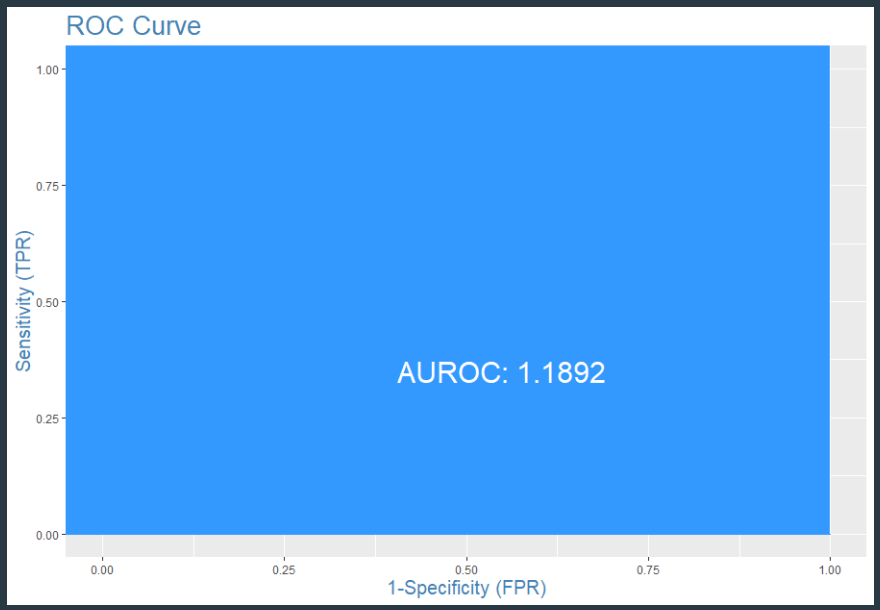
ROC curve

An ROC curve (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters: True Positive Rate. False Positive Rate.




```
## accuracy
auc = performance(ROC_pred_train,"auc")
auc@y.values
```

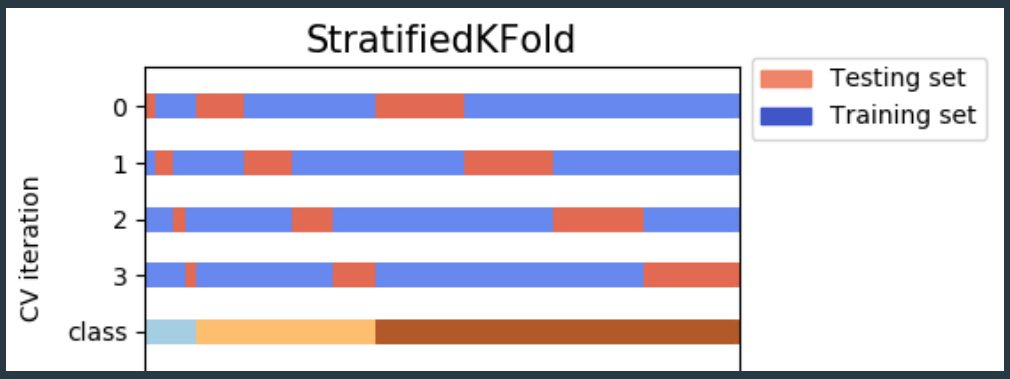
The AUROC value that we got from the model is less than 0.8. 0.7744336 to be specific. This shows that the model didn't do a good job.



The roc value of the training dataset is high and perfect

Iteration 3

Although we got high sensitivity, the accuracy wasn't that great. Therefore, we decided to make certain changes in the way we split our datasets. instead of using the normal stratified splitting, we used k fold cross-validation to split our data into training and testing datasets. this would ideally give randomness to our data. since the k fold cross validation Provides train/test indices to split data into train/test sets. This cross-validation object is a variation of KFold that returns stratified folds. The folds are made by preserving the percentage of samples for each class. . Note that the samples within each split will not be shuffled.



We follow the same procedure that we did in iteration 3 but this time we use the k fold cross-validation data sets to train and test the model.

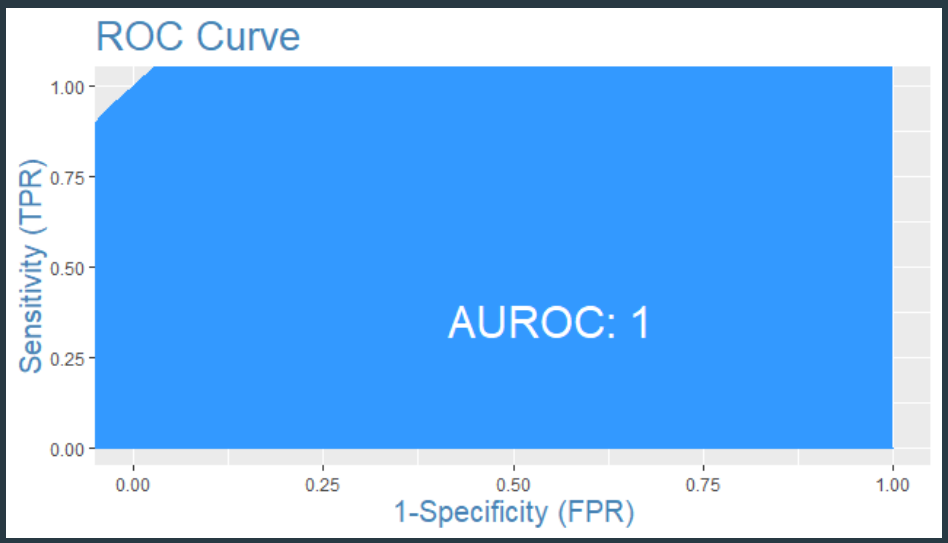
Inference

And as expected the accuracy of the model increased drastically. the accuracy jumped from 0.6986 to 0.7522. but the sensitivity decreases to 0.9063 which is still a great number.

The residual deviance is still less than the nill deviance which shows that the model fits the data good.

Iteration 3

```
> caret::confusionMatrix(data=as.factor(predic
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1142  328
         1  118  212

               Accuracy : 0.7522
                 95% CI : (0.7316, 0.772)
    No Information Rate : 0.7
    P-Value [Acc > NIR] : 4.942e-07

                  Kappa : 0.3363

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.9063
            Specificity : 0.3926
         Pos Pred Value : 0.7769
         Neg Pred Value : 0.6424
             Prevalence : 0.7000
         Detection Rate : 0.6344
   Detection Prevalence : 0.8167
      Balanced Accuracy : 0.6495

       'Positive' Class : 0
```



The ROC value is 1 which AUROC> 0.80 indicates that the model does a good job in discriminating between the two categories which comprise our target variable.

Framework 2 : Decision tree

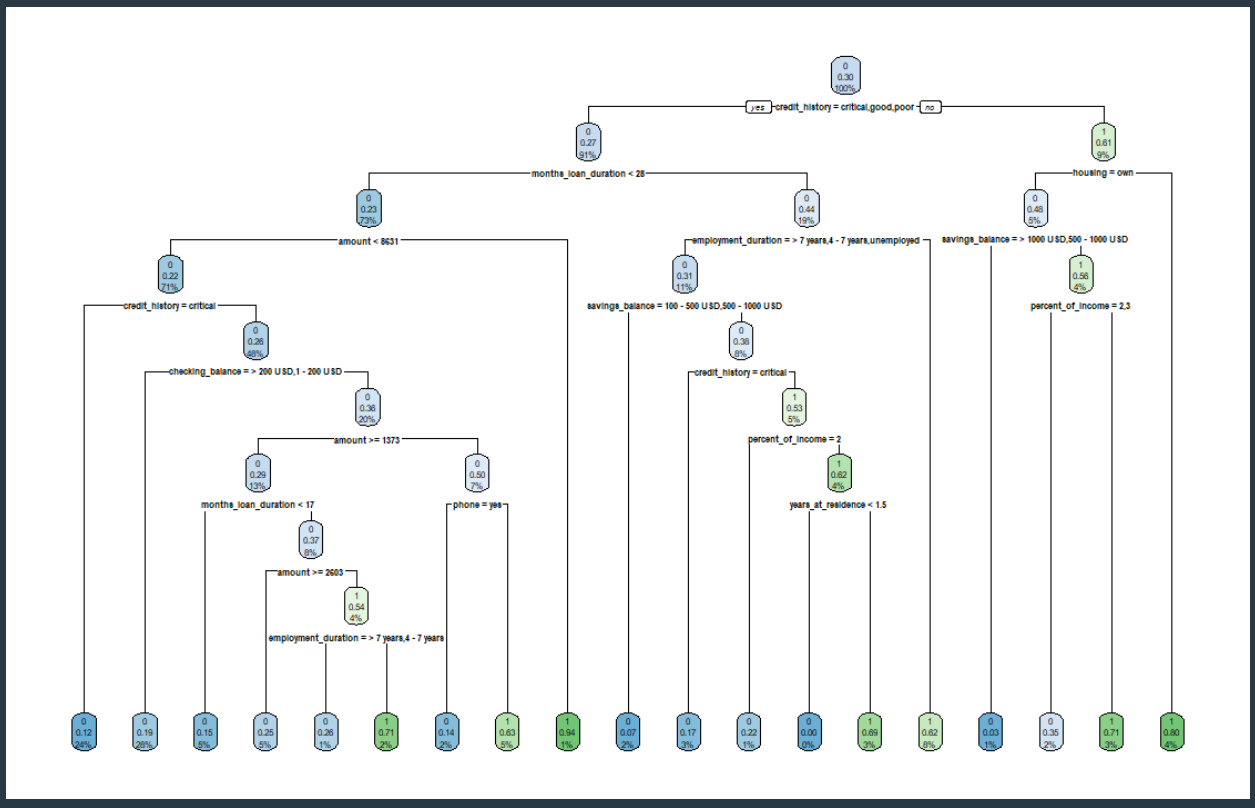
A decision tree is a flowchart-like structure in which each internal node represents a "test" on an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes).

Methodology

Iteration 1

We started off with loading the library rpart and rpart.plot. After that we made the initial model by taking default as the dependent and all the other variables except cust_id and acc_no as the independent variable.

```
#model
ct_fit=rpart(default~.-cust_id -acc_no ,data = train,method = 'class')

rpart.plot(ct_fit,extra = 106)
```



Confusion Matrix

After using the test dataset to predict we decided to look at the confusion matrix.

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1081  259
         1  179  281

               Accuracy : 0.7567
                 95% CI : (0.7362, 0.7763)
    No Information Rate : 0.7
    P-Value [Acc > NIR] : 5.126e-08

                  Kappa : 0.395

 Mcnemar's Test P-Value : 0.0001602

            Sensitivity : 0.8579
            Specificity : 0.5204
         Pos Pred Value : 0.8067
         Neg Pred Value : 0.6109
             Prevalence : 0.7000
         Detection Rate : 0.6006
   Detection Prevalence : 0.7444
      Balanced Accuracy : 0.6892

       'Positive' Class : 0
```

Inference

From the Confusion matrix that we obtained, we can observe an accuracy of about 75 % and a sensitivity of 0.8579.

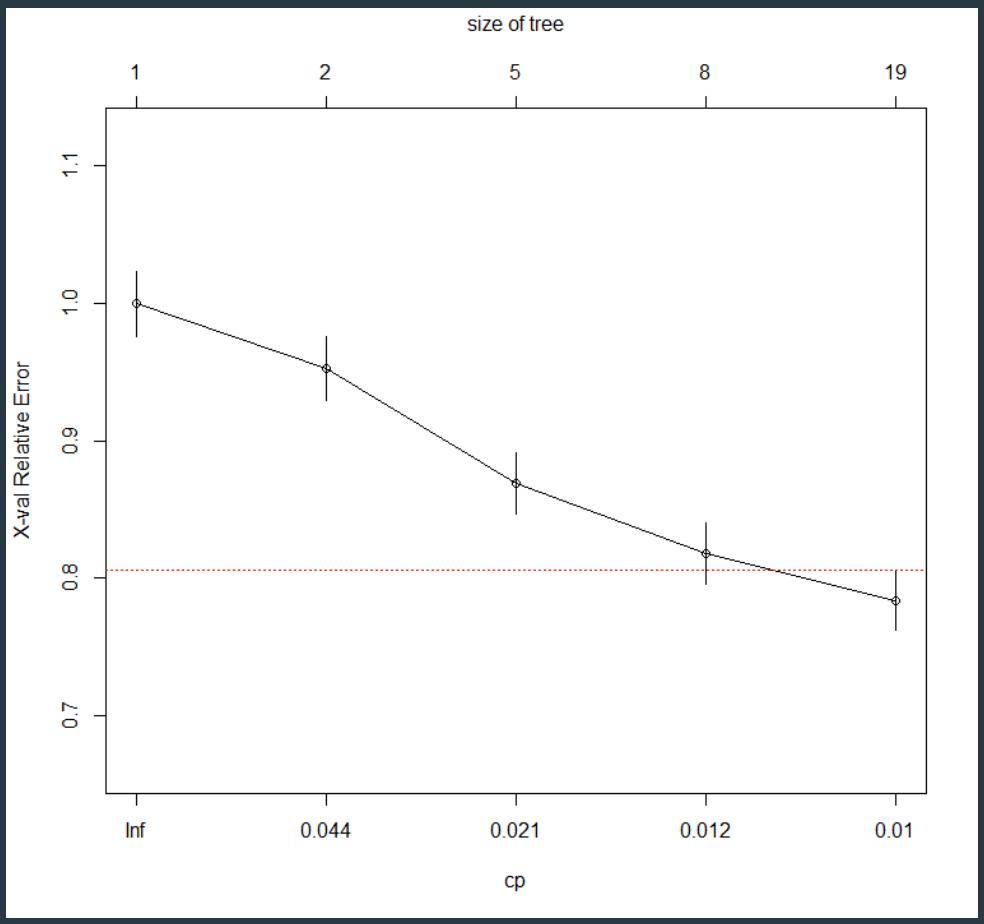
Pruning Trees

Pruning (parameter tuning) is a process of removing the paths of the tree which adds very little to the classification power of the tree.
Pruning is done with two things in mind:

- Reducing the complexity
- Reducing the chances of overfitting

For pruning trees, we used the control argument in the rpart() function.
first,we needed to find the maxdepth at which the X-val relative error is minimum. and for that we used the printcp( ) and plotcp( ) function.

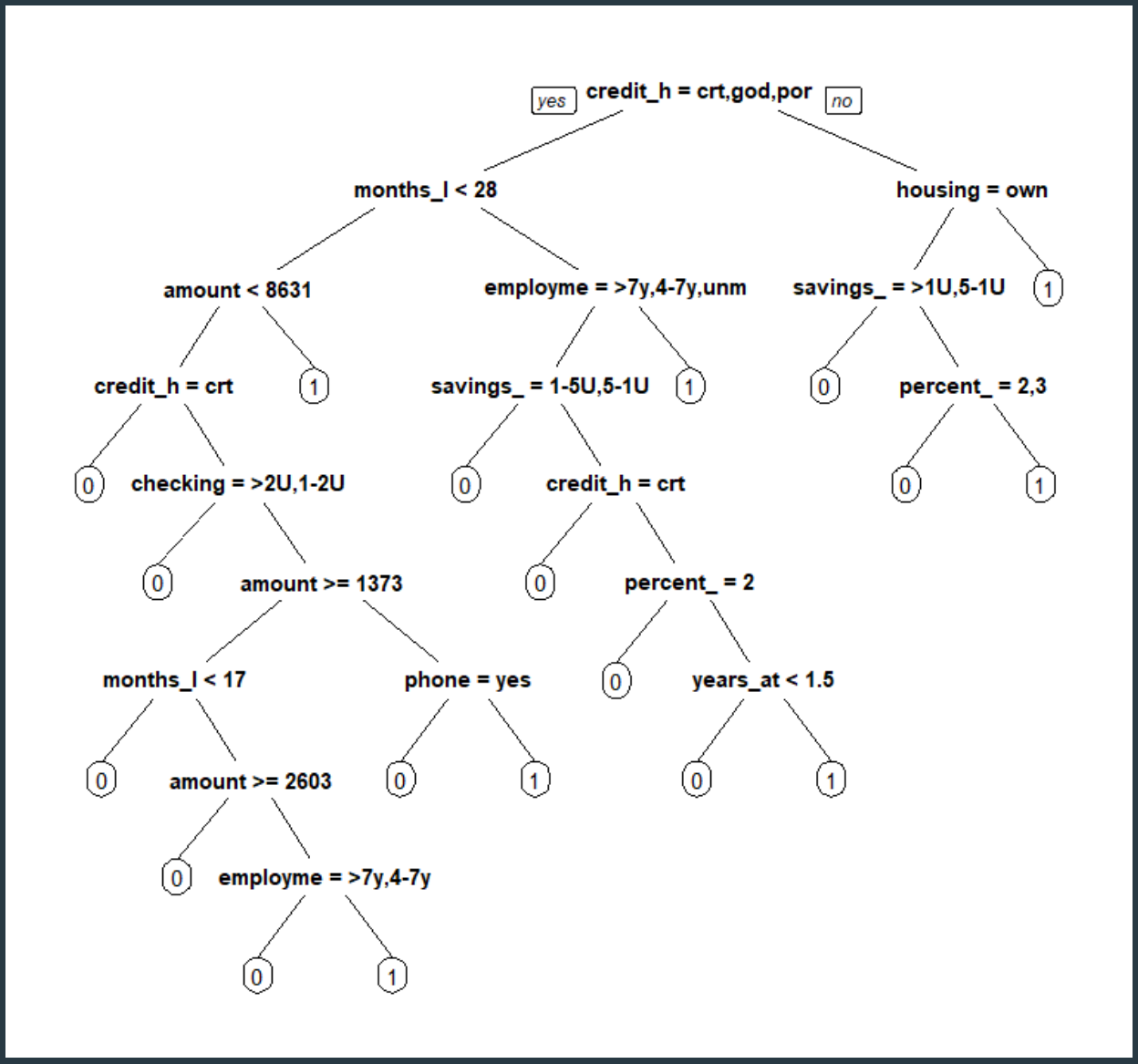
```
        CP nsplit rel error  xerror     xstd
1 0.062698      0   1.00000 1.00000 0.023570
2 0.031481      1   0.93730 0.95238 0.023236
3 0.014286      4   0.84286 0.86905 0.022581
4 0.010317      7   0.80000 0.81825 0.022136
5 0.010000     18   0.68571 0.78413 0.021816
```



from the plot, we observed that at depth=19, the x-val relative error is minimum

Making a new model with a controlled max depth of 19

```
ct_prune_fit=rpart(default~.-cust_id -acc_no ,data = train,method = 'class',
                   parms = list(split="gini"),control = rpart.control(
                     maxdepth = 19))
```



We used the new pruned model to predict the values of depth using the test dataset.

We obtain the confusion matrix

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1081  259
         1  179  281

               Accuracy : 0.7567
                 95% CI : (0.7362, 0.7763)
    No Information Rate : 0.7
    P-Value [Acc > NIR] : 5.126e-08

                  Kappa : 0.395

 Mcnemar's Test P-Value : 0.0001602

            Sensitivity : 0.8579
            Specificity : 0.5204
         Pos Pred Value : 0.8067
         Neg Pred Value : 0.6109
             Prevalence : 0.7000
         Detection Rate : 0.6006
   Detection Prevalence : 0.7444
      Balanced Accuracy : 0.6892

       'Positive' Class : 0
```

From the Confusion Matrix, we observed that the pruning didn't help the model. the Accuracy and Sensitivity of the model are still the same as before.

The ROC value of the model is 0.7345334 since the va;ue is less than 0.8 it shows that the the model is not that great.

Framework 3 : Random Forest

Random forest is a supervised learning algorithm. The "forest" it builds, is an ensemble of decision trees, usually trained with the “bagging” method. The general idea of the bagging method is that a combination of learning models increases the overall result.

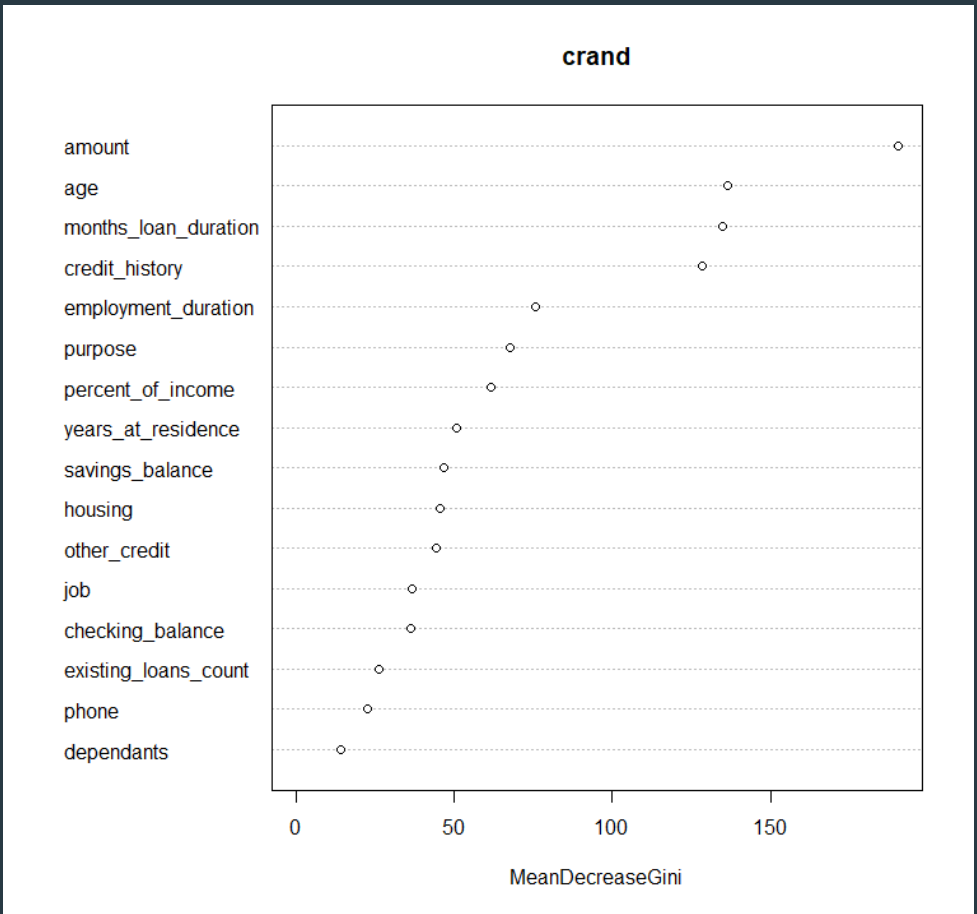
Methodology

Iteration 1

We started off with loading the library randomforest. After that we made the initial model by taking default as the dependent and all the other variables except cust_id and acc_no as the independent variable.

```
library(randomForest)
crand =  randomForest(as.factor(train$default) ~.-cust_id -acc_no, data = train,
                      nodesize =25, ntree = 200)
```

We started of with identifying the most important variables based on mean gini decrease.this Shows how each split result in low impurities or increased homogeneity



Predicting the test data using the model and obtaining the confusion matrix

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1251  105
         1    9  435

               Accuracy : 0.9367
                 95% CI : (0.9244, 0.9475)
    No Information Rate : 0.7
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8411

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.9929
            Specificity : 0.8056
         Pos Pred Value : 0.9226
         Neg Pred Value : 0.9797
             Prevalence : 0.7000
         Detection Rate : 0.6950
   Detection Prevalence : 0.7533
      Balanced Accuracy : 0.8992

       'Positive' Class : 0
```

This was by far the best accuracy and sensitivity combination that we have obtained. and after checking the ROC value which came up to be 0.9901396. this solidifies the fact that Random forest is the best framework among all the others (based on the raw data) that we can use for credit risk analysis.

RESULT

After using various framework and assesing them through various statistical tests, we cam to the following conclusions

Logistic Regression

Although the logistic Regression framework didn't provide us with a good pseudo-McFadden's R 2 value (0.16). it did give us good accuracy and high sensitivity upon changing the cross-validation method to k fold cross-validation. and

Decision tree

This was by far the poorest model as far as the accuracy and sensitivity are concerned. maybe this can be contributed to the information-hungry nature of the decision tree algorithm.

Random forest

We got by far the best results from this framework. The accuracy was a staggering 93.67% and the sensitivity was 0.9929 and not just that we also had an specificity of 0.8056 and ROC value of 0.9901396.

Final verdict

Although all the algorithms are great in their own rights. for us the Random forest algorithm gave the best results.

Suggestions and possibilities

The average bank has a variety of different departments that all work together to provide services to individual customers and businesses alike. While most customers are familiar with the retail banking department, which is what typically serves as the "face" of the bank. There are other departments in a bank depending upon the type of bank.
lets see how our credit risk model can be used in different departments of the bank with different intent.

Retail Banking Services

The retail bank division is likely the first place a person finds himself whenever he walks through the average bank's doors. It's also where he find help with:

- Checking and savings accounts
- Marketing and community relations
- Personal loans
- Credit cards
- Certificates of Deposit (CDs)
- Some types of insurance

The credit scoring models could be used extensively here. from setting the credit limit on a credit card of an obligor to assessing what interest rate that should be charged in case of loans. Credit scoring models can also be used to find the creditworthiness of an individual or a business, if the bank gives out some kind of insurance, this would help them in charging appropriate premiums.

Commercial and Business Banking

Whereas retail banking is aimed at providing services to individuals, commercial banking is catered towards businesses. Often, many mid-size and larger banks have both retail and commercial branches that operate under the same roof. That said, not every local bank branch or credit union may have a commercial business department, though most can accept commercial deposits.
Commercial banking departments work with a wide variety of companies, from local businesses to large corporations. Some of the services that fall under commercial banking include:

- Business loans
- Startup loans
- Lines of credit
- Equipment lending
- Employer services
- Commercial real estate

This is the department where most of the big transactions take place and it becomes that much more important to have a robust credit scoring system. because if the obligor defaults then that can result in a huge loss for the bank.

Loan service department

The loan servicing department of a bank takes care of communications with borrowers at any point in their loan journey, from managing the initial application process to assisting borrowers once loan funds have been disbursed. It is of utmost importance to constantly monitor the existing debt holder. because it may be possible that initially, the obligor looked creditworthy but with time the debt holders economic condition might change and he may not be eligible now to service the debt as he looked when he was applying for the loan. so a credit scoring system may save the bank from a potential loss if appropriate steps are taken.

Mortgage department

A mortgage is a loan typically used to buy a home or other piece of real estate for which that property then serves as collateral.
The major responsibilities of this department are as follows

- Assessing a potential borrower's eligibility
- Processing a mortgage application after collecting the required information and documents
- Inspecting a borrower's credit reports and other information to determine whether the bank should approve or deny a loan
- Processing mortgage payments
- Answering questions that a borrower has throughout the course of their loan

A good credit scoring system could help in deciding the appropriate mortgage rate that is charged depending on the credit worthiness of the obliigor.