Predicting Bank Defaulters

Roopak Krishna

13 January 2020

Content

Chapter 1

Introduction

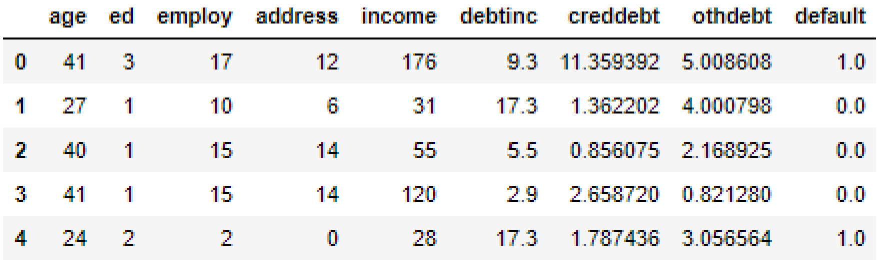
1.1 Problem Description

The objective of this project is to predict the bank loan default case with the help of various factors of customers. By achieving this goal, it would be possible to help accommodate in managing the customer for their loans on a daily basis, and providing better services to its customer.

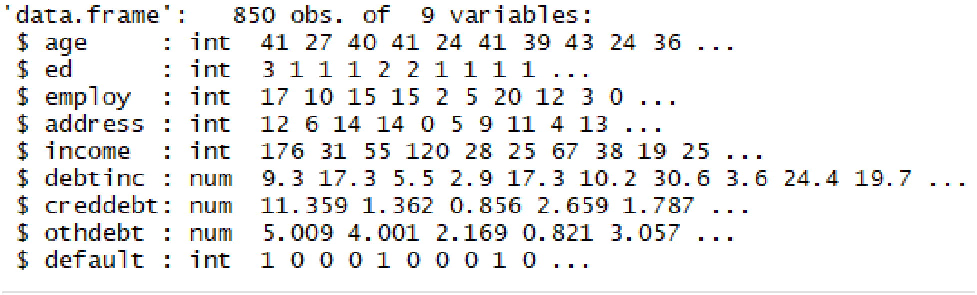
1.2 Data

Our task is to build a Classification models which will be responsible for predicting bank loan default case based on certain variables. Given below is the sample data that is used.

Fig.1 Data Overview



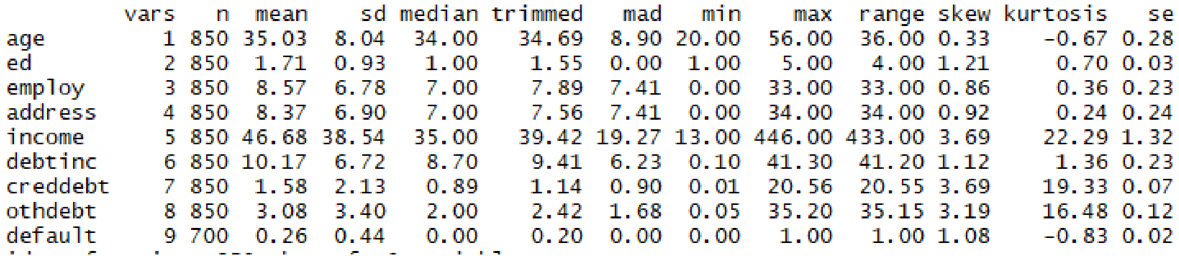
	age	ed	employ	address	income	debtinc	creddebt	othdebt	default
0	41	3	17	12	176	9.3	11.359392	5.008608	1.0
1	27	1	10	6	31	17.3	1.362202	4.000798	0.0
2	40	1	15	14	55	5.5	0.856075	2.168925	0.0
3	41	1	15	14	120	2.9	2.658720	0.821280	0.0
4	24	2	2	0	28	17.3	1.787436	3.056564	1.0



```
'data.frame':	850 obs. of  9 variables:
 $ age     : int  41 27 40 41 24 41 39 43 24 36 ...
 $ ed      : int  3 1 1 1 2 2 1 1 1 1 ...
 $ employ  : int  17 10 15 15 2 5 20 12 3 0 ...
 $ address : int  12 6 14 14 0 5 9 11 4 13 ...
 $ income  : int  176 31 55 120 28 25 67 38 19 25 ...
 $ debtinc : num  9.3 17.3 5.5 2.9 17.3 10.2 30.6 3.6 24.4 19.7 ...
 $ creddebt: num  11.359 1.362 0.856 2.659 1.787 ...
 $ othdebt : num  5.009 4.001 2.169 0.821 3.057 ...
 $ default : int  1 0 0 0 1 0 0 0 1 0 ...
```

Fig.2 Data types of variables

Here, Fig.2 shows the the datatypes of all variables that we have in our dataset and Fig.3 shows the Statistics of all variables to understand the data.



```
         vars   n  mean    sd median trimmed   mad   min    max  range skew kurtosis   se
age         1 850 35.03  8.04  34.00   34.69  8.90 20.00  56.00  36.00 0.33    -0.67 0.28
ed          2 850  1.71  0.93   1.00    1.55  0.00  1.00   5.00   4.00 1.21     0.70 0.03
employ      3 850  8.57  6.78   7.00    7.89  7.41  0.00  33.00  33.00 0.86     0.36 0.23
address     4 850  8.37  6.90   7.00    7.56  7.41  0.00  34.00  34.00 0.92     0.24 0.24
income      5 850 46.68 38.54  35.00   39.42 19.27 13.00 446.00 433.00 3.69    22.29 1.32
debtinc     6 850 10.17  6.72   8.70    9.41  6.23  0.10  41.30  41.20 1.12     1.36 0.23
creddebt    7 850  1.58  2.13   0.89    1.14  0.90  0.01  20.56  20.55 3.69    19.33 0.07
othdebt     8 850  3.08  3.40   2.00    2.42  1.68  0.05  35.20  35.15 3.19    16.48 0.12
default     9 700  0.26  0.44   0.00    0.20  0.00  0.00   1.00   1.00 1.08    -0.83 0.02
```

Fig.3 Data Stats

Chapter 2

Methodology

2.1 Pre-Processing

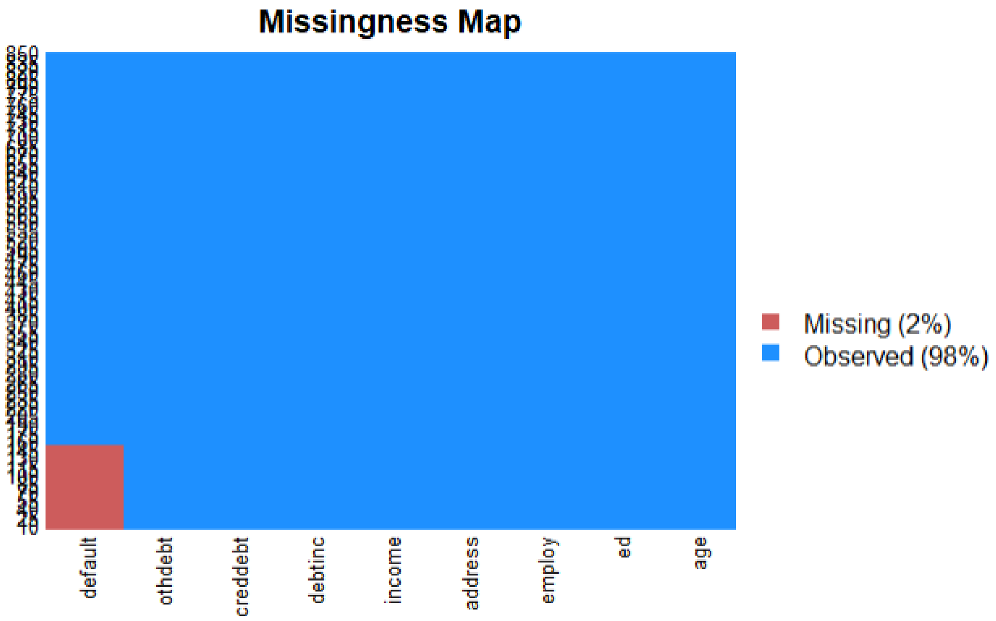
This Part is valuable to data science projects since it allows to get closer to the certainty that the future results will be *valid*, *correctly interpreted*, and *applicable* to the desired business contexts. Such level of certainty can be achieved only after raw data is validated and checked for anomalies, ensuring that the data set was collected without errors. EDA also helps to find insights that were not evident or worth investigating to business stakeholders and data scientists but can be very informative about a particular business.

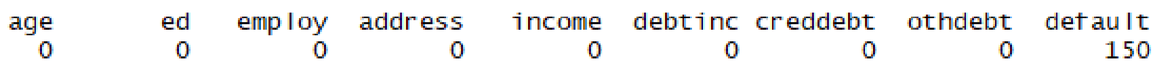
This is called Exploratory Data Analysis (EDA) which helps to answer all these questions, ensuring the best outcomes for the project. It is an approach for summarizing, visualizing, and becoming intimately familiar with the important characteristics of a data set.

2.1.1 Missing Value

It is the first step to identify the rows that contains missing data which will lead to undesired results or create some errors if not removed.

Fig.4 Missing Data Graph





```
  age        ed    employ  address    income   debtinc creddebt   othdebt   default
    0         0         0        0         0         0        0         0       150
```

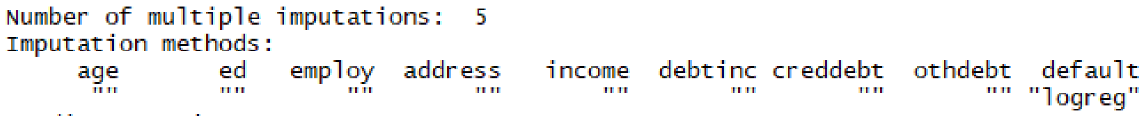
Fig.5 Missing Data Column wise

We can observe from this missingness graph Fig.4 that is provided by library(Amelia) and Fig.5 that there is some missing Data in our Dataset.
There is 17.64% of our data is missing so we can not ignore it or drop it. We need more and more data to train our model. So, we will try to fill these missing values.

2.1.2 Missing Data Imputation

MICE (Multivariate Imputation via Chained Equations) is one of the commonly used package by R users. Creating multiple imputations as compared to a single imputation (such as mean) takes care of uncertainty in missing values.

MICE assumes that the missing data are Missing at Random (MAR), which means that the probability that a value is missing depends only on observed value and can be predicted using them. It imputes data on a variable by variable basis by specifying an imputation model per variable. So the method is used “logreg” because this method is used only for binary variables.



```
Number of multiple imputations:  5
Imputation methods:
      age        ed    employ  address    income   debtinc creddebt   othdebt   default
       ""        ""        ""       ""        ""        ""       ""        "" "logreg"
```

Fig.6 Imputation

2.1.3 Outlier Analysis

Outliers are extreme values that deviate from other observations on data , they may indicate a variability in a measurement, experimental errors or a novelty. In other words, an outlier is an observation that diverges from an overall pattern on a sample.

Detecting outliers is of major importance for almost any quantitative discipline (ie: Physics, Economy, Finance, Machine Learning, Cyber Security). In machine learning and in any quantitative discipline the quality of data is as important as the quality of a prediction

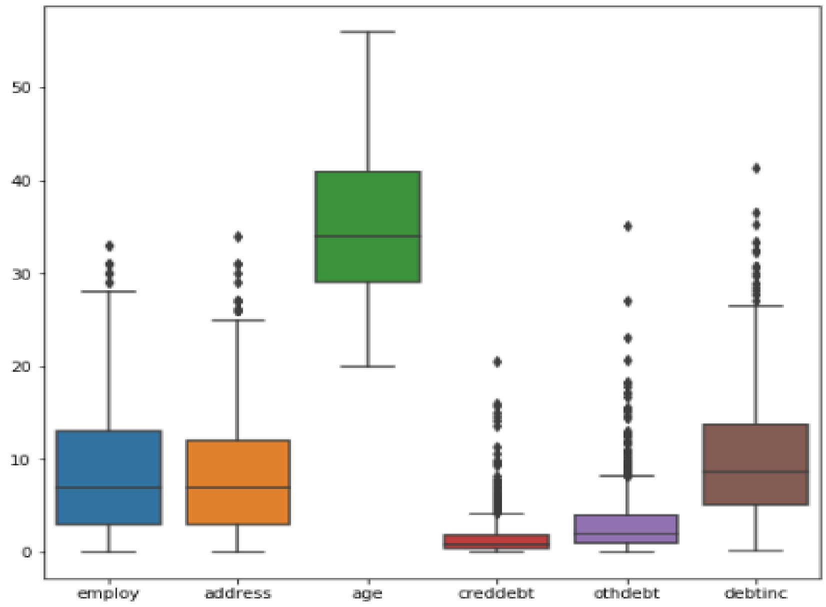


Fig.7 Boxplot for outliers(python)(a)

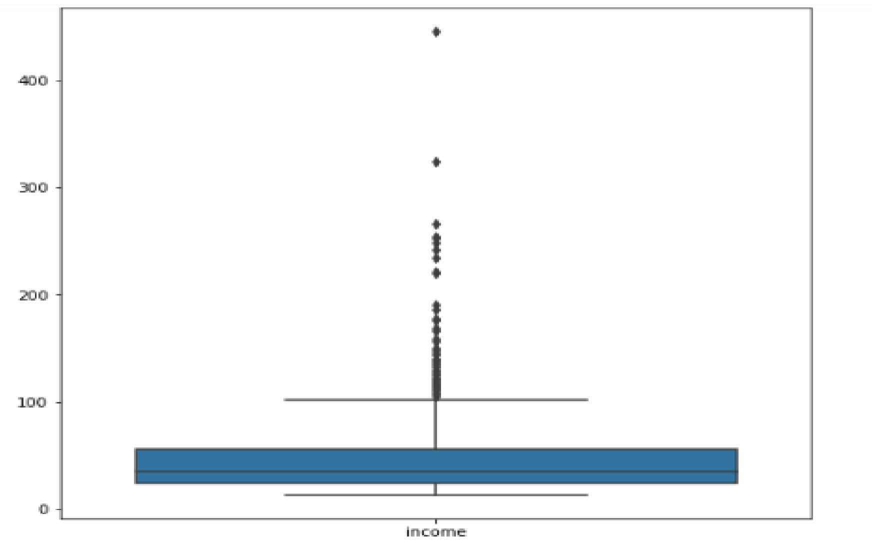
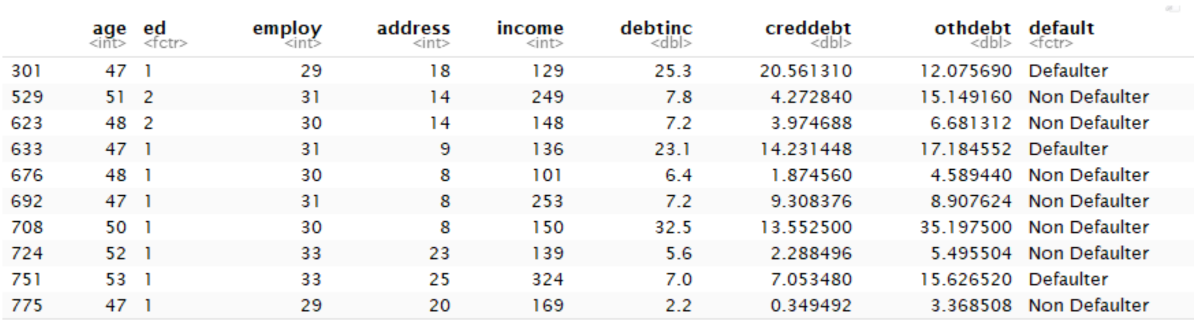


Fig.8 Boxplot for outliers(python)(b)

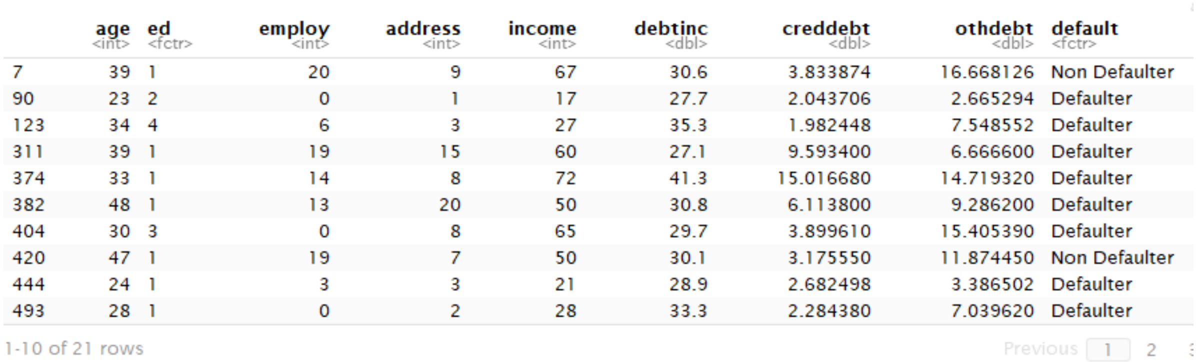
As we can see in Fig.4 that there are some outliers observed in employ, address, creddebt, othdebt, debtinc, income. We will further analyse these variables for better understanding.

Here are some of the outliers that found in some of variables. Looking at every outlier will help in better understanding.



	age	ed	employ	address	income	debtinc	creddebt	othdebt	default
	<int>	<fctr>	<int>	<int>	<int>	<dbl>	<dbl>	<dbl>	<fctr>
301	47	1	29	18	129	25.3	20.561310	12.075690	Defaulter
529	51	2	31	14	249	7.8	4.272840	15.149160	Non Defaulter
623	48	2	30	14	148	7.2	3.974688	6.681312	Non Defaulter
633	47	1	31	9	136	23.1	14.231448	17.184552	Defaulter
676	48	1	30	8	101	6.4	1.874560	4.589440	Non Defaulter
692	47	1	31	8	253	7.2	9.308376	8.907624	Non Defaulter
708	50	1	30	8	150	32.5	13.552500	35.197500	Non Defaulter
724	52	1	33	23	139	5.6	2.288496	5.495504	Non Defaulter
751	53	1	33	25	324	7.0	7.053480	15.626520	Defaulter
775	47	1	29	20	169	2.2	0.349492	3.368508	Non Defaulter

Fig.9 Employ outliers



	age	ed	employ	address	income	debtinc	creddebt	othdebt	default
	<int>	<fctr>	<int>	<int>	<int>	<dbl>	<dbl>	<dbl>	<fctr>
7	39	1	20	9	67	30.6	3.833874	16.668126	Non Defaulter
90	23	2	0	1	17	27.7	2.043706	2.665294	Defaulter
123	34	4	6	3	27	35.3	1.982448	7.548552	Defaulter
311	39	1	19	15	60	27.1	9.593400	6.666600	Defaulter
374	33	1	14	8	72	41.3	15.016680	14.719320	Defaulter
382	48	1	13	20	50	30.8	6.113800	9.286200	Defaulter
404	30	3	0	8	65	29.7	3.899610	15.405390	Defaulter
420	47	1	19	7	50	30.1	3.175550	11.874450	Non Defaulter
444	24	1	3	3	21	28.9	2.682498	3.386502	Defaulter
493	28	1	0	2	28	33.3	2.284380	7.039620	Defaulter

1-10 of 21 rows

Fig.10 debtinc outliers

As we can see from Fig.9 and Fig.10, the outliers in employ will not be removed because there may be case of some rare jobs with their status so we actually don't know what kind of employment is. Similarly for address variable, there may be case of some geographic areas where people are very less or with very high density. Not removing income outlier because there may be case where people could have very less income or very high income. Not removing othdebt outlier because there may be case where customers other debt may be different from other customers. We will remove only debtinc, creddebt outlier because after analysis of debtinc outlier, Individual debt of customer can not be higher than its gross income.

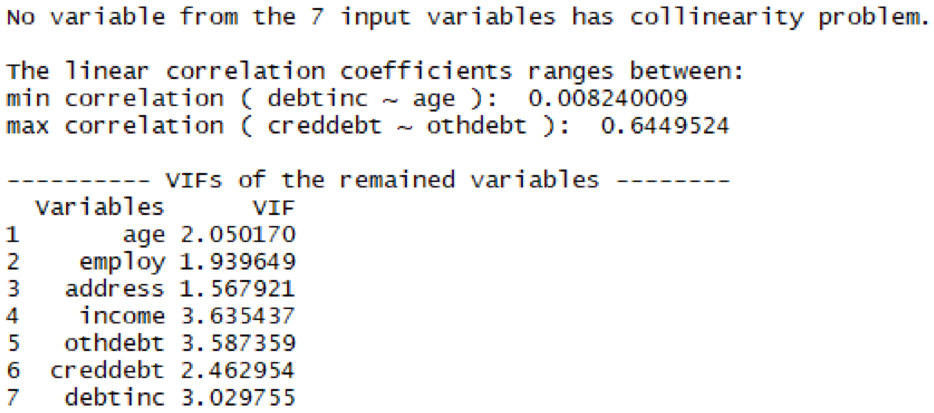
2.1.4 Multi Collinearity & Feature Selection

Multicollinearity occurs when independent variables in a Classification model are correlated. This correlation is a problem because independent variables should be *independent*. If the degree of correlation between variables is high enough, it can cause problems when you fit the model and interpret the results.

A key goal of regression analysis is to isolate the relationship between each independent variable and the dependent variable. The interpretation of a regression coefficient is that it represents the mean change in the dependent variable for each 1 unit change in an independent variable when you *hold all of the other independent variables constant*.

When independent variables are correlated, it indicates that changes in one variable are associated with shifts in another variable. The stronger the correlation, the more difficult it is to change one variable without changing another. It becomes difficult for the model to estimate the relationship between each independent variable and the dependent variable *independently* because the independent variables tend to change in unison.

Here, I have used variance inflation factor (VIF) for a set of variables and exclude the highly correlated variables from the set through a stepwise procedure. This method can be used to deal with multicollinearity problems when you fit statistical models.



```
No variable from the 7 input variables has collinearity problem.

The linear correlation coefficients ranges between:
min correlation ( debtinc ~ age ):  0.008240009
max correlation ( creddebt ~ othdebt ):  0.6449524

---------- VIFs of the remained variables --------
  Variables      VIF
1       age 2.050170
2    employ 1.939649
3   address 1.567921
4    income 3.635437
5   othdebt 3.587359
6  creddebt 2.462954
7   debtinc 3.029755
```

Fig.11 vifcor

We can observe that our data has no multicollinearity problem that means every variable carry different/unique information.

Further, I have implemented the correlation plot which is provided by library corrgram.

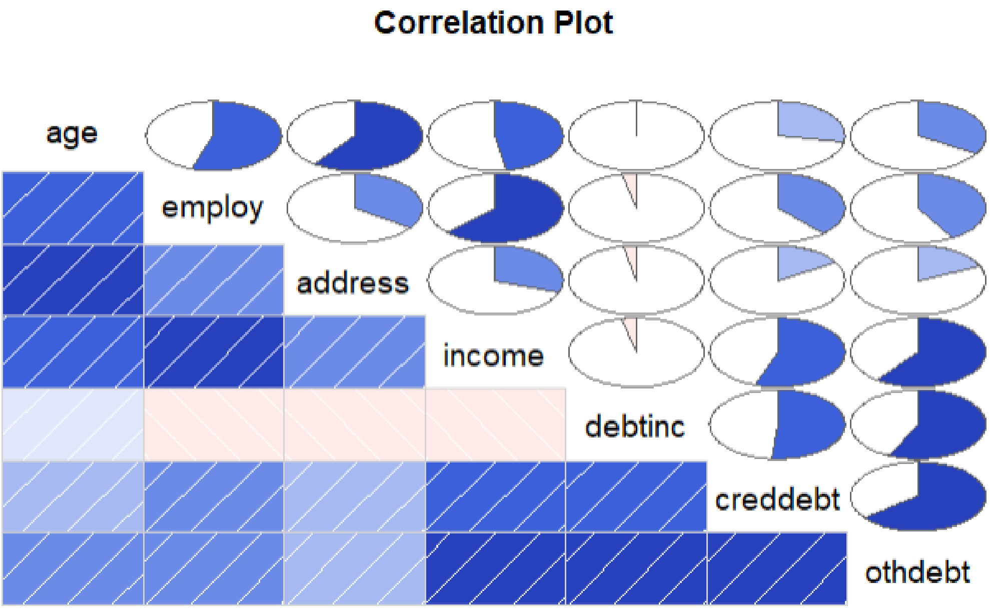


Fig.12 Correlation plot

From this plot, we can identify that all variables are nicely correlated with each other.

2.2 Modeling

2.2.1 Model Selection

First crucial step in modeling to know which algorithms we have to used based on our target variable.

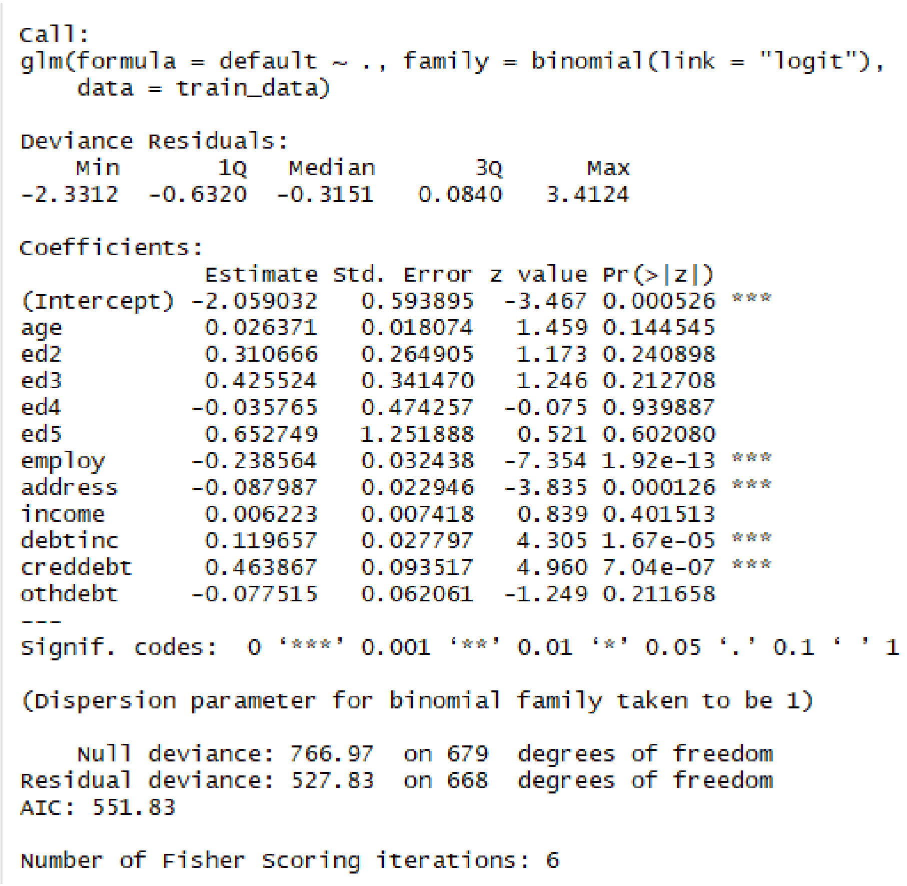
The target variable can fall in either of the four categories:

1. Nominal
2. Ordinal
3. Interval
4. Ratio

The target variable in our model is a categorical variable i.e. default. Hence the models that we choose are Logistic regression, Random Forest. The metric choosen to evaluate the performance of our model is accuracy, sensitivity, specificity, area under curve, ROC.

2.2.2 Logistic Regression

Logistic regression is the go-to method for binary classification problems (problems with two class values). Logistic regression is named for the function used at the core of the method, the logistic function. The logistic function, also called the sigmoid function.

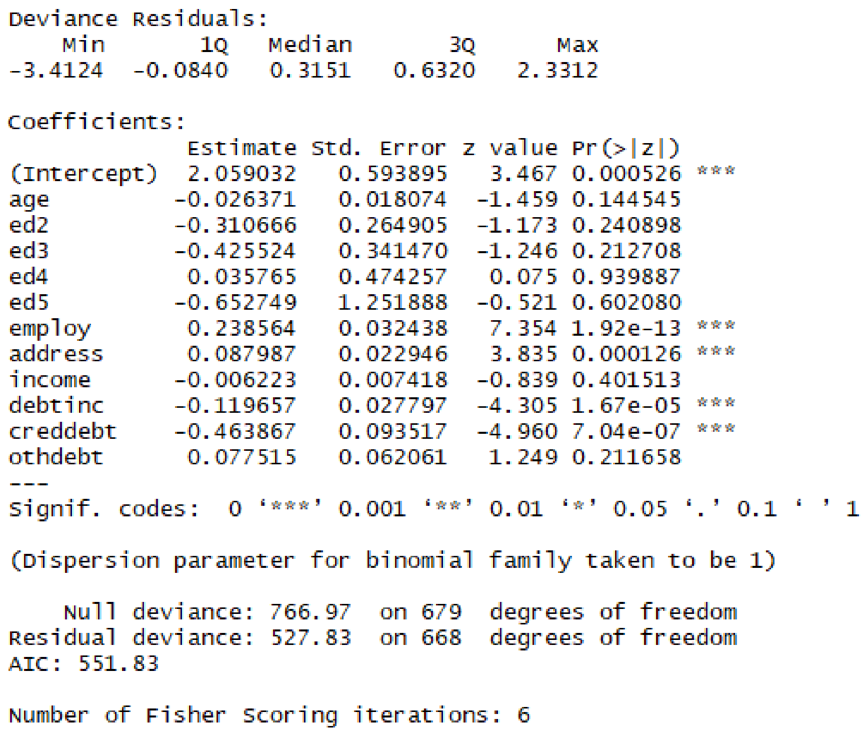


```
Call:
glm(formula = default ~ ., family = binomial(link = "logit"), 
    data = train_data)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-2.3312  -0.6320  -0.3151   0.0840   3.4124  

Coefficients:
             Estimate Std. Error z value Pr(>|z|)    
(Intercept) -2.059032   0.593895  -3.467 0.000526 ***
age          0.026371   0.018074   1.459 0.144545    
ed2          0.310666   0.264905   1.173 0.240898    
ed3          0.425524   0.341470   1.246 0.212708    
ed4         -0.035765   0.474257  -0.075 0.939887    
ed5          0.652749   1.251888   0.521 0.602080    
employ      -0.238564   0.032438  -7.354 1.92e-13 ***
address     -0.087987   0.022946  -3.835 0.000126 ***
income       0.006223   0.007418   0.839 0.401513    
debtinc      0.119657   0.027797   4.305 1.67e-05 ***
creddebt     0.463867   0.093517   4.960 7.04e-07 ***
othdebt     -0.077515   0.062061  -1.249 0.211658    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 766.97  on 679  degrees of freedom
Residual deviance: 527.83  on 668  degrees of freedom
AIC: 551.83

Number of Fisher Scoring iterations: 6
```

Fig.13 lr model

2.2.3 K- fold Cross Validation

Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample. Cross-validation is primarily used in applied machine learning to estimate the skill of a machine learning model on unseen data. That is, to use a limited sample in order to estimate how the model is expected to perform in general when used to make predictions on data not used during the training of the model. We have used 10 folds in our models.



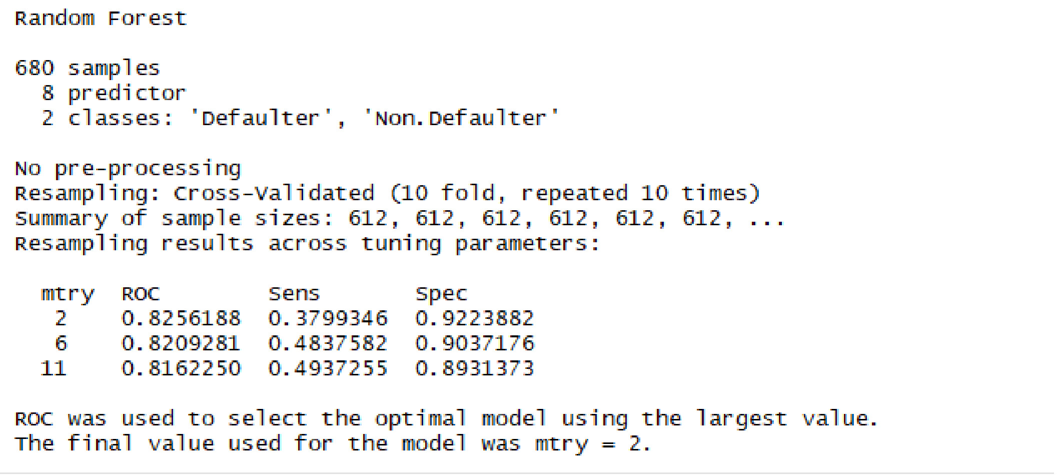
```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.4124  -0.0840   0.3151   0.6320   2.3312

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  2.059032   0.593895   3.467 0.000526 ***
age         -0.026371   0.018074  -1.459 0.144545
ed2         -0.310666   0.264905  -1.173 0.240898
ed3         -0.425524   0.341470  -1.246 0.212708
ed4          0.035765   0.474257   0.075 0.939887
ed5         -0.652749   1.251888  -0.521 0.602080
employ       0.238564   0.032438   7.354 1.92e-13 ***
address      0.087987   0.022946   3.835 0.000126 ***
income      -0.006223   0.007418  -0.839 0.401513
debtinc     -0.119657   0.027797  -4.305 1.67e-05 ***
creddebt    -0.463867   0.093517  -4.960 7.04e-07 ***
othdebt      0.077515   0.062061   1.249 0.211658
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 766.97  on 679  degrees of freedom
Residual deviance: 527.83  on 668  degrees of freedom
AIC: 551.83

Number of Fisher Scoring iterations: 6
```

Fig.14 lr cv

This is the summary of our logistic regression classifier after tuning it with the 10 fold cross validation parameters.

2.2.4 Random Forest

Random forest is like bootstrapping algorithm with Decision tree (CART) model. Random forest tries to build multiple CART models with different samples and different initial variables. Here, we have given ntree value 500.



```
Random Forest

680 samples
  8 predictor
  2 classes: 'Defaulter', 'Non.Defaulter'

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 10 times)
Summary of sample sizes: 612, 612, 612, 612, 612, 612, ...
Resampling results across tuning parameters:

  mtry  ROC        Sens       Spec
   2    0.8256188  0.3799346  0.9223882
   6    0.8209281  0.4837582  0.9037176
  11    0.8162250  0.4937255  0.8931373

ROC was used to select the optimal model using the largest value.
The final value used for the model was mtry = 2.
```

Fig.15 rf model

Chapter3

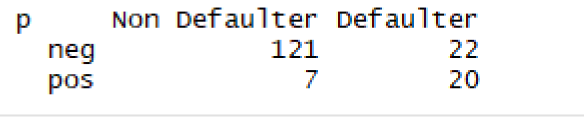
Conclusion

3.1 Model Evaluation

Now that we have a few models for predicting the target variable, we need to decide which one to choose.
There are several criteria that exist for evaluating and comparing models. We can compare the models using
any of the following criteria:
1. Predictive Performance
2. Interpretability
3. Computational Efficiency

Models will be evaluated on the basis of various metrics that can be calculated through confusion matrix and ROC curve.

Evaulation of 2.2.2 Logistic Regression:



```
p      Non Defaulter Defaulter
   neg           121        22
   pos             7        20
```

Fig.16 CM LR

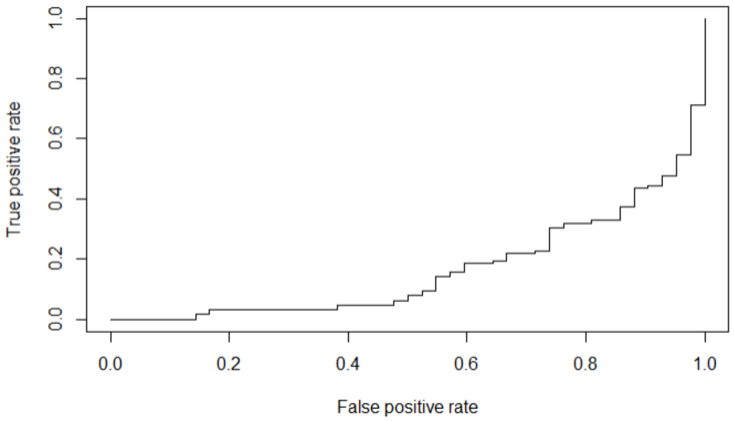
True Positive (TP) : We predicted 20 cases will be defaulter and turns out to be true.
True Negative (TN) : We predicted 121 cases are Non defaulter and it turns out to be true.
False Positive (FP) : We predicted 22 cases are defaulter but turns out to be false.
False Negative (FN) : We predicted 7 cases are Non defaulter but turns out to be false.

Accuracy : how often is the classifier correct.
(TP+TN)/(TP+TN+FP+FN) = 82.94%

Sensitivity/Recall/True Positive rate : When it's actually Defaulter, how often does it predict yes.
TP/(TP+FN) = 74.07%

Precision : The precision metric shows the accuracy of the positive class. It measures how likely the prediction of the positive class is correct.
TP/(TP+FP) = 47.61%

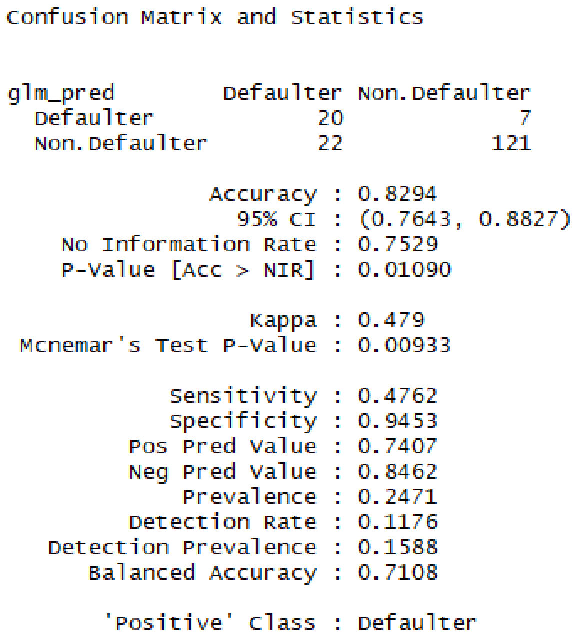
Specificity/ True Negative rate : When its actually Non defaulter, how often does it predict the same.
TN/(TN+FP) = 84.61%



Area Under Curve = 16.24%

Fig.17 ROC1

Evaulation of 2.2.3 K- fold Cross Validation:



```
Confusion Matrix and Statistics


glm_pred        Defaulter Non.Defaulter
  Defaulter            20             7
  Non.Defaulter        22           121

               Accuracy : 0.8294
                 95% CI : (0.7643, 0.8827)
    No Information Rate : 0.7529
    P-Value [Acc > NIR] : 0.01090

                  Kappa : 0.479
 Mcnemar's Test P-Value : 0.00933

            Sensitivity : 0.4762
            Specificity : 0.9453
         Pos Pred Value : 0.7407
         Neg Pred Value : 0.8462
             Prevalence : 0.2471
         Detection Rate : 0.1176
   Detection Prevalence : 0.1588
      Balanced Accuracy : 0.7108

       'Positive' Class : Defaulter
```

Fig.18 LR k-fold

True Positive (TP) : We predicted 20 cases will be defaulter and turns out to be true.
True Negative (TN) : We predicted 121 cases are Non defaulter and it turns out to be true.
False Positive (FP) : We predicted 22 cases are defaulter but turns out to be false.
False Negative (FN) : We predicted 7 cases are Non defaulter but turns out to be false.

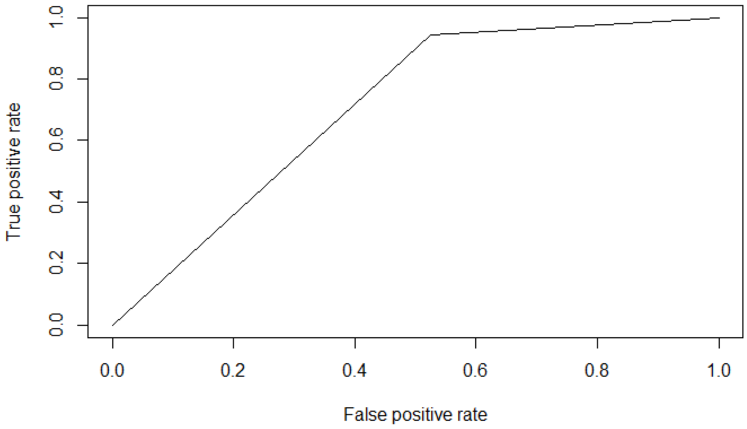
Accuracy : how often is the classifier correct.
(TP+TN)/(TP+TN+FP+FN) = 82.94%

Sensitivity/Recall/True Positive rate : When it's actually Defaulter, how often does it predict yes.

TP/(TP+FN) = 47.62%

Precision : The precision metric shows the accuracy of the positive class. It measures how likely the prediction of the positive class is correct.
TP/(TP+FP) = 74.07%

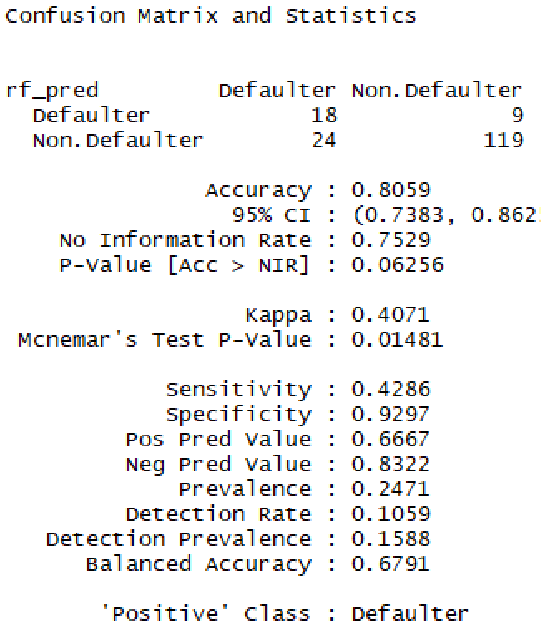
Specificity/ True Negative rate : When its actually Non defaulter, how often does it predict the same.
TN/(TN+FP) = 94.53%



Area under curve = 71.07%

Fig.19 ROC2

Evaulation of 2.2.4 Random Forest:



```
Confusion Matrix and Statistics

rf_pred         Defaulter Non.Defaulter
  Defaulter            18             9
  Non.Defaulter        24           119

               Accuracy : 0.8059
                 95% CI : (0.7383, 0.862
    No Information Rate : 0.7529
    P-Value [Acc > NIR] : 0.06256

                  Kappa : 0.4071
 Mcnemar's Test P-Value : 0.01481

            Sensitivity : 0.4286
            Specificity : 0.9297
         Pos Pred Value : 0.6667
         Neg Pred Value : 0.8322
             Prevalence : 0.2471
         Detection Rate : 0.1059
   Detection Prevalence : 0.1588
      Balanced Accuracy : 0.6791

       'Positive' Class : Defaulter
```

Fig.20 rf cm

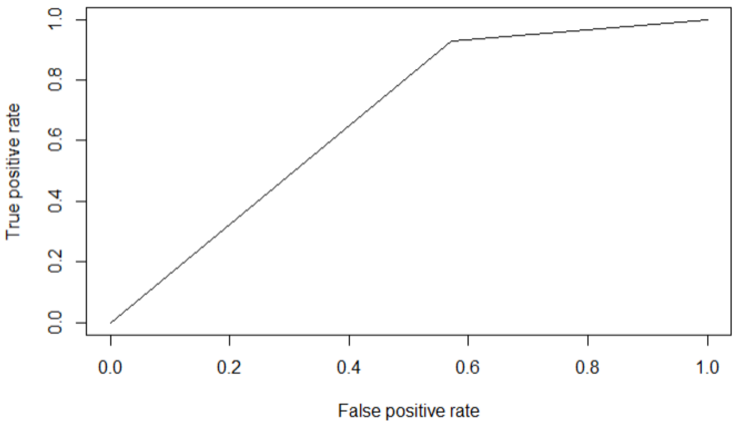
True Positive (TP) : We predicted 18 cases will be defaulter and turns out to be true.
True Negative (TN) : We predicted 119 cases are Non defaulter and it turns out to be true.
False Positive (FP) : We predicted 24 cases are defaulter but turns out to be false.
False Negative (FN) : We predicted 9 cases are Non defaulter but turns out to be false.

Accuracy : how often is the classifier correct.
(TP+TN)/(TP+TN+FP+FN) = 80.59%

Sensitivity/Recall/True Positive rate : When it's actually Defaulter, how often does it predict yes.
TP/(TP+FN) = 42.86%

Precision : The precision metric shows the accuracy of the positive class. It measures how likely the prediction of the positive class is correct.
TP/(TP+FP) = 66.66%

Specificity/ True Negative rate : When its actually Non defaulter, how often does it predict the same.
TN/(TN+FP) = 92.97%



Area Under Curve = 67.91%

Fig.21 ROC3

An ROC curve is the most commonly used way to visualize the performance of a binary classifier**,** and AUC is the best way to summarize its performance in a single number**.**

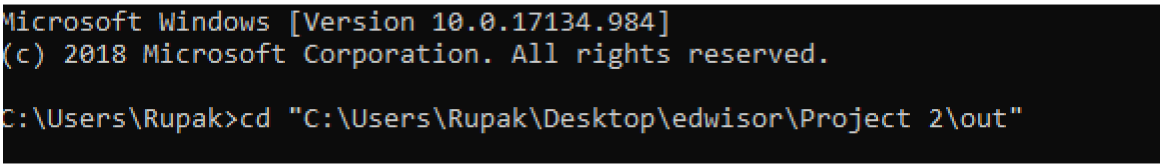
We can see from Fig.17 ROC1 and AUC is 16.24% that 2.2.2 Logistic regression has performed poorly. So, after comparing 2.2.3 and 2.2.4, 2.2.3 Logistic regression after 10 fold cross validation has performed good with accuracy of 82.94%, Precision is 74.07% and also the AUC is 71.07%.
After Concluding, we can say that after 10- fold cross validation, our Logistic regression Classifier has performed well.

Chapter 4

Deployment

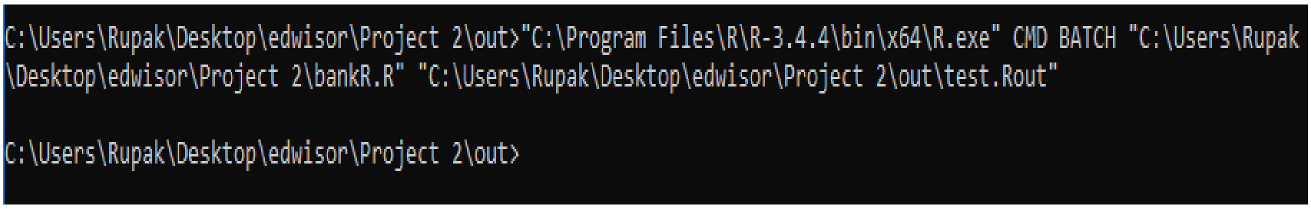
4.1 Model Deployment

At Industry level, we have to deploy our predictive model so that we will get the complete summary of model regardless of running each code again and again.



```
Microsoft Windows [Version 10.0.17134.984]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Rupak>cd "C:\Users\Rupak\Desktop\edwisor\Project 2\out"
```

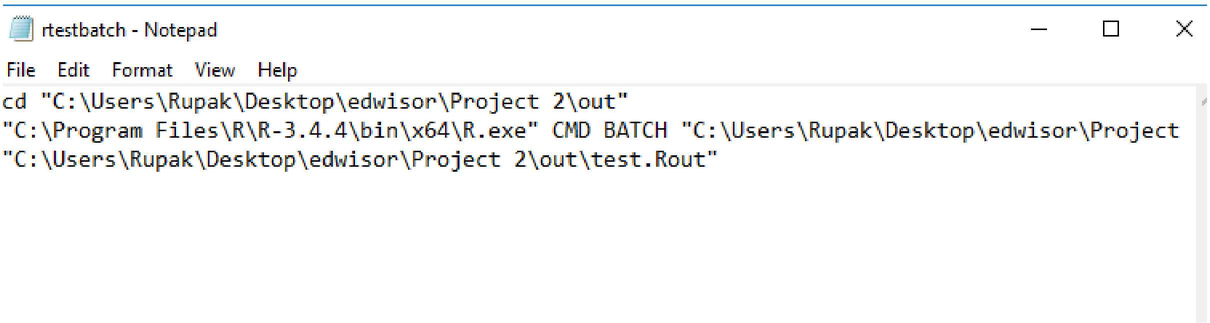
1. First, Open the command prompt, put the path of directory.



```
C:\Users\Rupak\Desktop\edwisor\Project 2\out>"C:\Program Files\R\R-3.4.4\bin\x64\R.exe" CMD BATCH "C:\Users\Rupak
\Desktop\edwisor\Project 2\bankR.R" "C:\Users\Rupak\Desktop\edwisor\Project 2\out\test.Rout"

C:\Users\Rupak\Desktop\edwisor\Project 2\out>
```

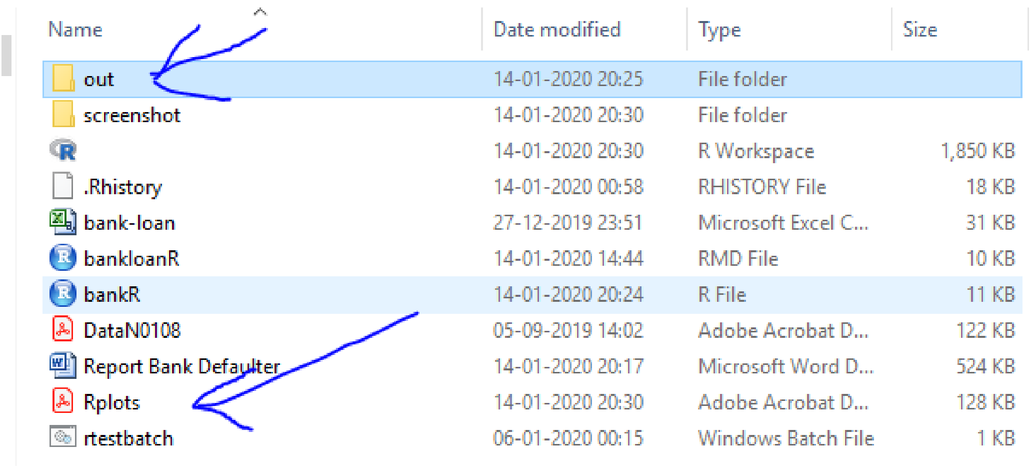
2. Put the path of the R.exe so that it will know it is a R file followed by CMD BATCH command.
3. After that, path of the code file.
4. Followed by the path and name of output file where do you want to store.

It these all these run successfully, we can create a batch file so that we don't have to do this process again and again.



```
cd "C:\Users\Rupak\Desktop\edwisor\Project 2\out"
"C:\Program Files\R\R-3.4.4\bin\x64\R.exe" CMD BATCH "C:\Users\Rupak\Desktop\edwisor\Project
"C:\Users\Rupak\Desktop\edwisor\Project 2\out\test.Rout"
```

Put all these command line instructions into a notepad file as shown in above screenshot and save it as .bat extension file.



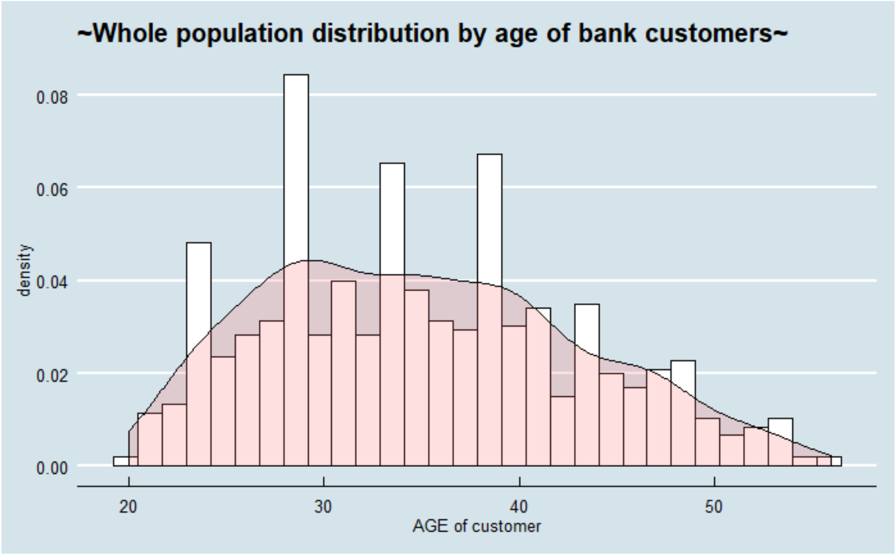
Name	Date modified	Type	Size
out	14-01-2020 20:25	File folder	
screenshot	14-01-2020 20:30	File folder	
	14-01-2020 20:30	R Workspace	1,850 KB
.Rhistory	14-01-2020 00:58	RHISTORY File	18 KB
bank-loan	27-12-2019 23:51	Microsoft Excel C...	31 KB
bankloanR	14-01-2020 14:44	RMD File	10 KB
bankR	14-01-2020 20:24	R File	11 KB
DataN0108	05-09-2019 14:02	Adobe Acrobat D...	122 KB
Report Bank Defaulter	14-01-2020 20:17	Microsoft Word D...	524 KB
Rplots	14-01-2020 20:30	Adobe Acrobat D...	128 KB
rtestbatch	06-01-2020 00:15	Windows Batch File	1 KB

We can see that after clicking on “rtestbatch file”, the summary of our bankR.R is stored in out folder and all the graphs are saved in Rplots.pdf file.

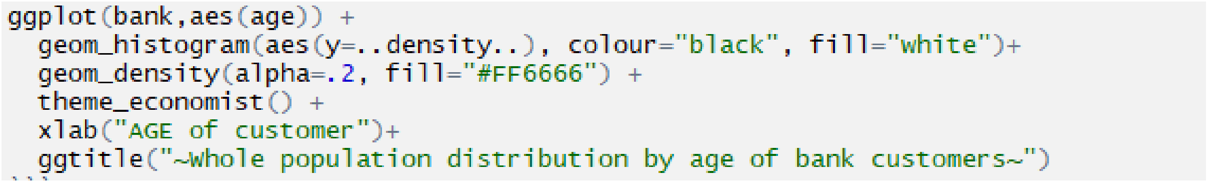
So through all this process, we just have to click on batch file to see the summary of our predictive model.

Appendix A – R Code

Histogram of age distribution:

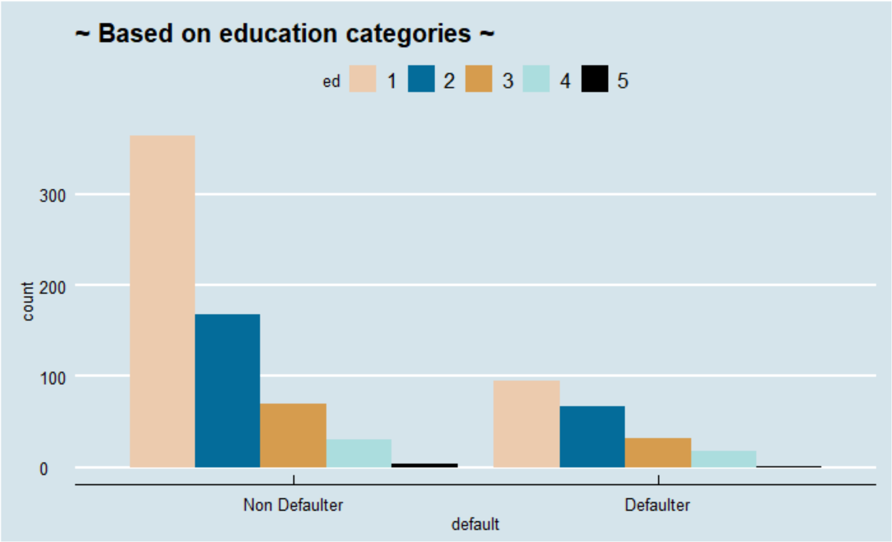


Code:

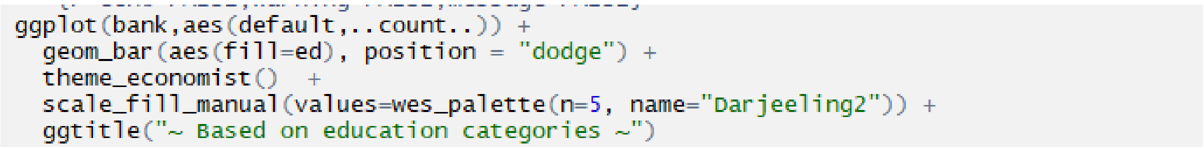


```
ggplot(bank,aes(age)) +
  geom_histogram(aes(y=..density..), colour="black", fill="white")+
  geom_density(alpha=.2, fill="#FF6666") +
  theme_economist() +
  xlab("AGE of customer")+
  ggtitle("~whole population distribution by age of bank customers~")
```

Bar graph b/w education categories and default:

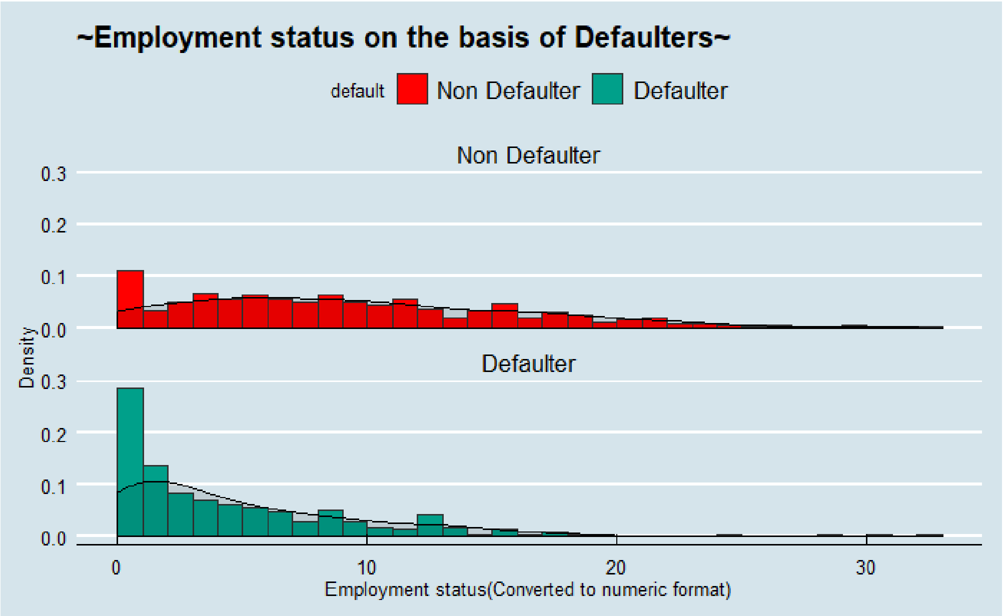


Code:

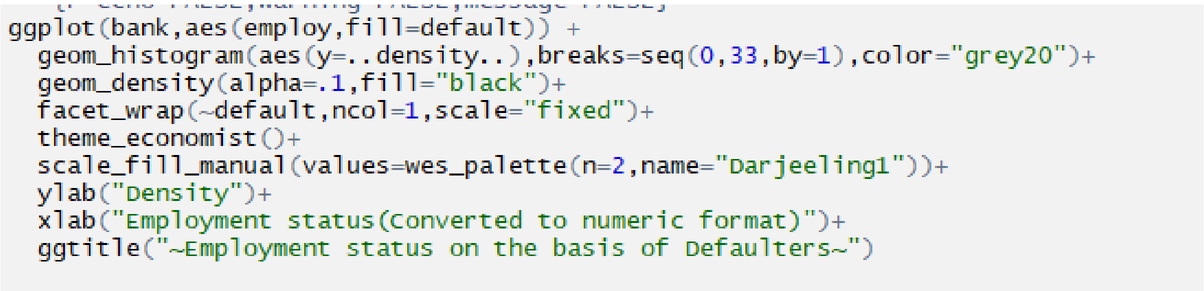


```
ggplot(bank,aes(default,..count..)) +
  geom_bar(aes(fill=ed), position = "dodge") +
  theme_economist()  +
  scale_fill_manual(values=wes_palette(n=5, name="Darjeeling2")) +
  ggtitle("~ Based on education categories ~")
```

Histogram b/w default and employment:

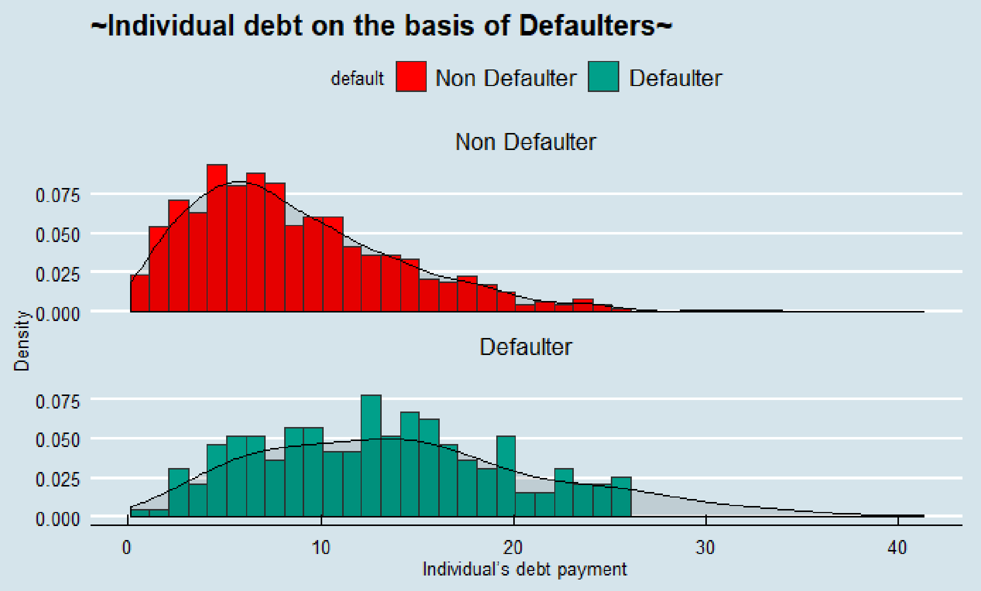


Code:

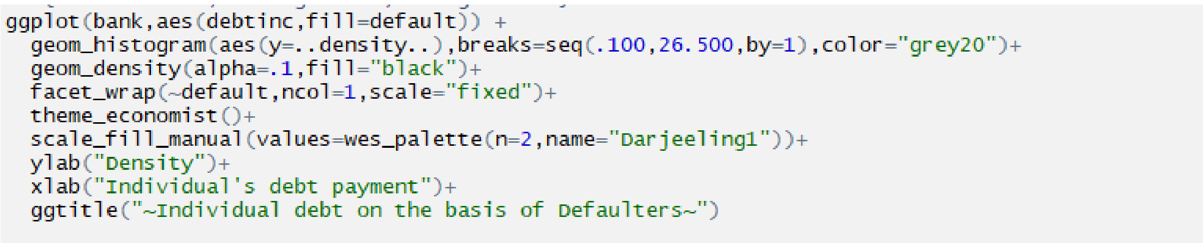


```
ggplot(bank,aes(employ,fill=default)) +
  geom_histogram(aes(y=..density..),breaks=seq(0,33,by=1),color="grey20")+
  geom_density(alpha=.1,fill="black")+
  facet_wrap(~default,ncol=1,scale="fixed")+
  theme_economist()+
  scale_fill_manual(values=wes_palette(n=2,name="Darjeeling1"))+
  ylab("Density")+
  xlab("Employment status(Converted to numeric format)")+
  ggtitle("~Employment status on the basis of Defaulters~")
```

Histogram b/w default and debtinc:

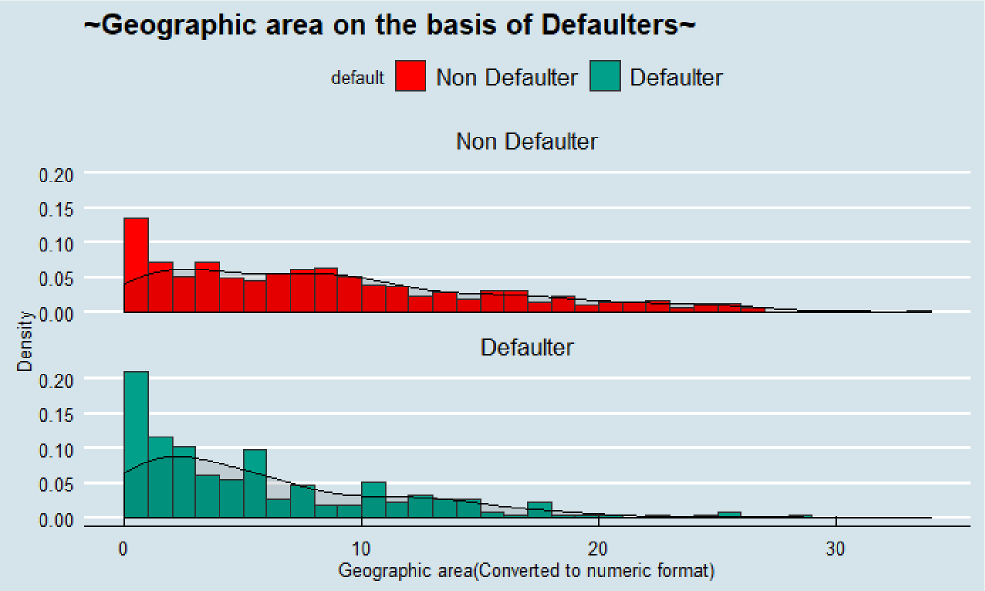


Code:

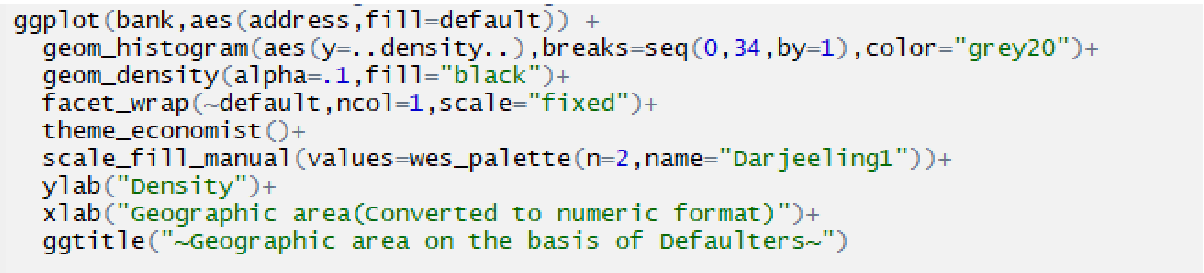


```
ggplot(bank,aes(debtinc,fill=default)) +
  geom_histogram(aes(y=..density..),breaks=seq(.100,26.500,by=1),color="grey20")+
  geom_density(alpha=.1,fill="black")+
  facet_wrap(~default,ncol=1,scale="fixed")+
  theme_economist()+
  scale_fill_manual(values=wes_palette(n=2,name="Darjeeling1"))+
  ylab("Density")+
  xlab("Individual's debt payment")+
  ggtitle("~Individual debt on the basis of Defaulters~")
```

Histogram b/w default and address:

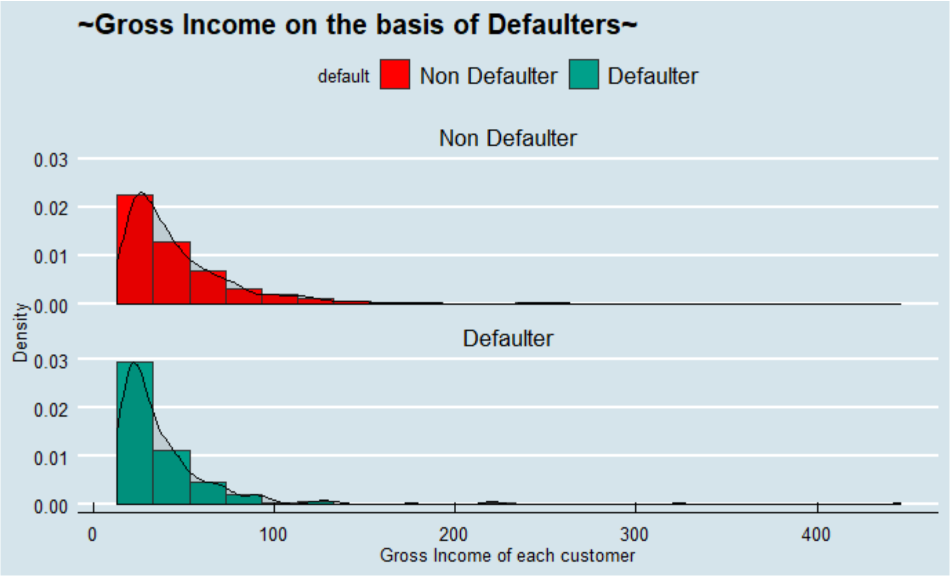


Code:

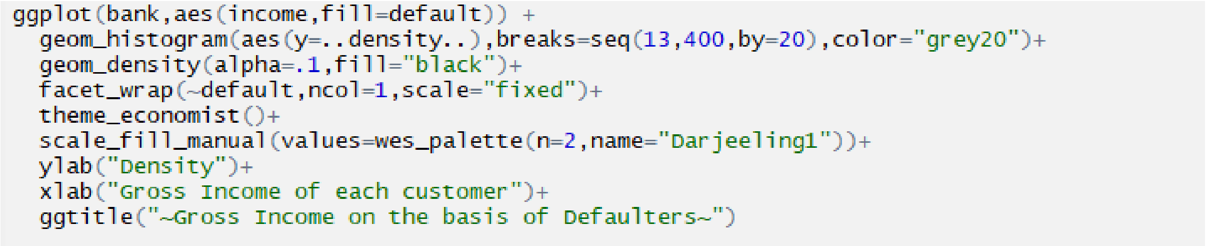


```
ggplot(bank,aes(address,fill=default)) +
  geom_histogram(aes(y=..density..),breaks=seq(0,34,by=1),color="grey20")+
  geom_density(alpha=.1,fill="black")+
  facet_wrap(~default,ncol=1,scale="fixed")+
  theme_economist()+
  scale_fill_manual(values=wes_palette(n=2,name="Darjeeling1"))+
  ylab("Density")+
  xlab("Geographic area(Converted to numeric format)")+
  ggtitle("~Geographic area on the basis of Defaulters~")
```

Histogram b/w default and income:



Code:



```
ggplot(bank,aes(income,fill=default)) +
  geom_histogram(aes(y=..density..),breaks=seq(13,400,by=20),color="grey20")+
  geom_density(alpha=.1,fill="black")+
  facet_wrap(~default,ncol=1,scale="fixed")+
  theme_economist()+
  scale_fill_manual(values=wes_palette(n=2,name="Darjeeling1"))+
  ylab("Density")+
  xlab("Gross Income of each customer")+
  ggtitle("~Gross Income on the basis of Defaulters~")
```

References

Book: R CookBook by Paul Teetor

Python for Data Analysis by Wes McKinney