Prediction of Credit Worthiness of loan for future potential customers

Pankaj Yadav

# knitr::opts\_chunk$set(echo = TRUE)  
library(readxl)  
##install.packages('xlsx')  
library(openxlsx)  
data=read.xlsx("C:\\Users\\ypank\\Downloads\\Credit\_Risk6\_final.xlsx")##reading the file  
#View(data)  
##Read the XLSX file of Sheet 1   
data1 = read.xlsx("C:\\Users\\ypank\\Downloads\\Credit\_Risk6\_final.xlsx",sheet=1)##reading the file sheetwise  
#View(data1)

**Import Package Code**

For reading specific sheet from an excel, mentioning sheet name is very important or by default sheet1 will be imported. For importing **read.xlsx** () function will be used which is under **readxl** package. **Read.xlsx()** is used to read excel files into R having extension “.xlxs”

str(data1)##structure of the dataset1, first line shows the class of imported data and number of observations and variables.

## 'data.frame': 13 obs. of 13 variables:  
## $ ID : num 781 782 783 784 785 786 787 788 789 790 ...  
## $ Checking.Acct : chr "No Acct" "Low" "No Acct" "High" ...  
## $ Credit.History : chr "All Paid" "Current" "Current" "Current" ...  
## $ Loan.Reason : chr "Car New" "Small Appliance" "Small Appliance" "Business" ...  
## $ Savings.Acct : chr "MedHigh" "Low" "Low" "Low" ...  
## $ Employment : chr "Short" "Medium" "Very Short" "Medium" ...  
## $ Personal.Status : chr "Single" "Single" "Divorced" "Single" ...  
## $ Housing : chr "Rent" "Rent" "Own" "Rent" ...  
## $ Job.Type : chr "Unskilled" "Skilled" "Skilled" "Skilled" ...  
## $ Foreign.National : chr "No" "No" "No" "Yes" ...  
## $ Months.since.Checking.Acct.opened: num 11 37 13 16 9 49 37 12 19 16 ...  
## $ Residence.Time : num 2 4 2 4 2 4 2 1 4 4 ...  
## $ Age : num 39 23 28 25 43 39 30 19 38 32 ...

summary(data1)##summary of the dataset1,

These datasets are summarised using **summary()** functions, which gives insight about the data such as there mean, median, 1st Quartile, 3rd Quartile, min , max values for numeric data, for categorical data levels are given, etc.

## ID Checking.Acct Credit.History Loan.Reason   
## Min. :781 Length:13 Length:13 Length:13   
## 1st Qu.:784 Class :character Class :character Class :character   
## Median :787 Mode :character Mode :character Mode :character   
## Mean :787   
## 3rd Qu.:790   
## Max. :793   
## Savings.Acct Employment Personal.Status   
## Length:13 Length:13 Length:13   
## Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character   
##   
##   
##   
## Housing Job.Type Foreign.National   
## Length:13 Length:13 Length:13   
## Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character   
##   
##   
##   
## Months.since.Checking.Acct.opened Residence.Time Age   
## Min. : 9.00 Min. :1.000 Min. :19.00   
## 1st Qu.:13.00 1st Qu.:2.000 1st Qu.:27.00   
## Median :16.00 Median :3.000 Median :30.00   
## Mean :21.23 Mean :2.923 Mean :31.46   
## 3rd Qu.:25.00 3rd Qu.:4.000 3rd Qu.:38.00   
## Max. :49.00 Max. :4.000 Max. :43.00

##Read the XLSX file of Sheet 2  
data2 = read.xlsx("C:\\Users\\ypank\\Downloads\\Credit\_Risk6\_final.xlsx",sheet = 2)  
str(data2)#structure of the data

## 'data.frame': 780 obs. of 14 variables:  
## $ ID : num 1 2 3 4 5 6 7 8 9 10 ...  
## $ Checking.Acct : chr "No Acct" "0Balance" "0Balance" "0Balance" ...  
## $ Credit.History : chr "All Paid" "Current" "Current" "Current" ...  
## $ Loan.Reason : chr "Car New" "Car New" "Car New" "Furniture" ...  
## $ Savings.Acct : chr "Low" "Low" "No Acct" "No Acct" ...  
## $ Employment : chr "Medium" "Short" "Long" "Long" ...  
## $ Personal.Status : chr "Single" "Divorced" "Divorced" NA ...  
## $ Housing : chr "Own" "Own" "Own" "Own" ...  
## $ Job.Type : chr "Management" "Skilled" "Skilled" "Skilled" ...  
## $ Foreign.National : chr "No" "No" "No" "No" ...  
## $ Months.since.Checking.Acct.opened : num 7 16 25 31 7 13 22 25 25 13 ...  
## $ Residence.Time.(In.current.district): num 3 2 2 4 4 2 3 4 4 4 ...  
## $ Age : num 44 28 28 30 35 22 29 33 62 40 ...  
## $ Credit.Standing : chr "Good" "Bad" "Bad" "Good" ...

summary(data2)#summary of the data

## ID Checking.Acct Credit.History Loan.Reason   
## Min. : 1.0 Length:780 Length:780 Length:780   
## 1st Qu.:195.8 Class :character Class :character Class :character   
## Median :390.5 Mode :character Mode :character Mode :character   
## Mean :390.5   
## 3rd Qu.:585.2   
## Max. :780.0   
## Savings.Acct Employment Personal.Status   
## Length:780 Length:780 Length:780   
## Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character   
##   
##   
##   
## Housing Job.Type Foreign.National   
## Length:780 Length:780 Length:780   
## Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character   
##   
##   
##   
## Months.since.Checking.Acct.opened Residence.Time.(In.current.district)  
## Min. : 5.0 Min. :-2.000   
## 1st Qu.: 13.0 1st Qu.: 2.000   
## Median : 19.0 Median : 3.000   
## Mean : 23.2 Mean : 2.868   
## 3rd Qu.: 29.5 3rd Qu.: 4.000   
## Max. :120.0 Max. :10.000   
## Age Credit.Standing   
## Min. :18.00 Length:780   
## 1st Qu.:26.00 Class :character   
## Median :32.00 Mode :character   
## Mean :34.75   
## 3rd Qu.:41.00   
## Max. :99.00

#View(data2)  
colnames(data2)##gives the column name of data

## [1] "ID"   
## [2] "Checking.Acct"   
## [3] "Credit.History"   
## [4] "Loan.Reason"   
## [5] "Savings.Acct"   
## [6] "Employment"   
## [7] "Personal.Status"   
## [8] "Housing"   
## [9] "Job.Type"   
## [10] "Foreign.National"   
## [11] "Months.since.Checking.Acct.opened"   
## [12] "Residence.Time.(In.current.district)"  
## [13] "Age"   
## [14] "Credit.Standing"

This command gives the sum of NA values are present into the attributes.

sum(is.na(data2$Checking.Acct))#sum of the NA value present in the column

## [1] 0

sum(is.na(data2$Credit.History))

## [1] 0

sum(is.na(data2$Loan.Reason))

## [1] 0

sum(is.na(data2$Savings.Acct))

## [1] 0

sum(is.na(data2$Employment))

## [1] 33

sum(is.na(data2$Personal.Status))

## [1] 6

sum(is.na(data2$Housing))

## [1] 5

sum(is.na(data2$Job.Type))

## [1] 0

sum(is.na(data2$Foreign.National))

## [1] 0

sum(is.na(data2$Months.since.Checking.Acct.opened))

## [1] 0

sum(is.na(data2$Age))

## [1] 0

sum(is.na(data2$Credit.Standing))

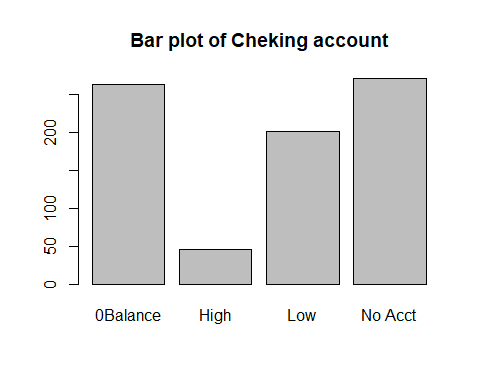
## [1] 0

table(data2$Checking.Acct)#table of the categories count of the data,

#BANK DETAILS

##   
## 0Balance High Low No Acct   
## 263 45 201 271

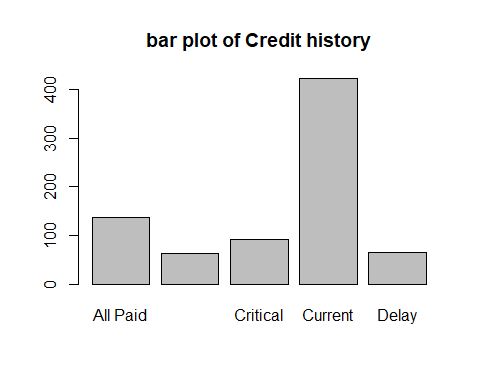
##windows(20,20)  
##Bank details  
barplot(table(data2$Checking.Acct),main = "Bar plot of Cheking account")



table(data2$Credit.History)

##   
## All Paid Bank Paid Critical Current Delay   
## 138 63 92 423 64

barplot(table(data2$Credit.History),main = "bar plot of Credit history")

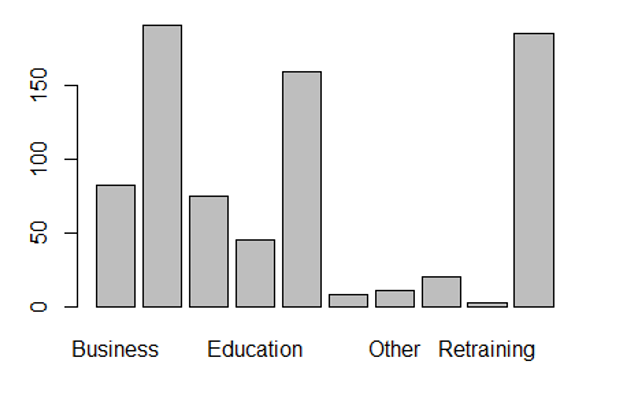


table(data2$Loan.Reason)

##   
## Business Car New Car Used Education   
## 82 191 75 45   
## Furniture Large Appliance Other Repairs   
## 160 8 11 20   
## Retraining Small Appliance   
## 2 186

barplot(table(data2$Loan.Reason),main = "Bar plot of LOANREASON")

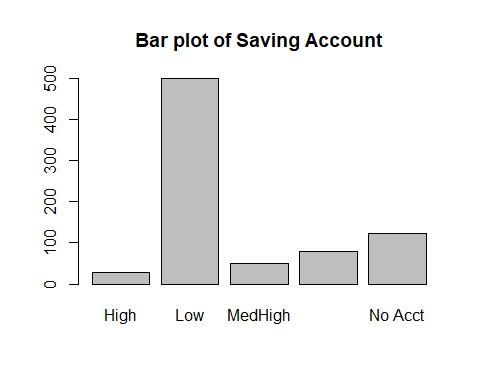
**Bar Plot of LOANREASON**



table(data2$Savings.Acct)

##   
## High Low MedHigh MedLow No Acct   
## 29 500 49 80 122

barplot(table(data2$Savings.Acct),main = "Bar plot of Saving Account")



table(data2$Employment)

##   
## Long Medium Retired Short Unemployed Very Short   
## 186 140 2 242 43 134

table(data2$Personal.Status)

##   
## Divorced Married Single   
## 273 70 431

table(data2$Housing)

##   
## Other Own Rent   
## 94 524 157

table(data2$Job.Type)

##   
## Management Skilled Unemployed Unskilled   
## 101 498 19 162

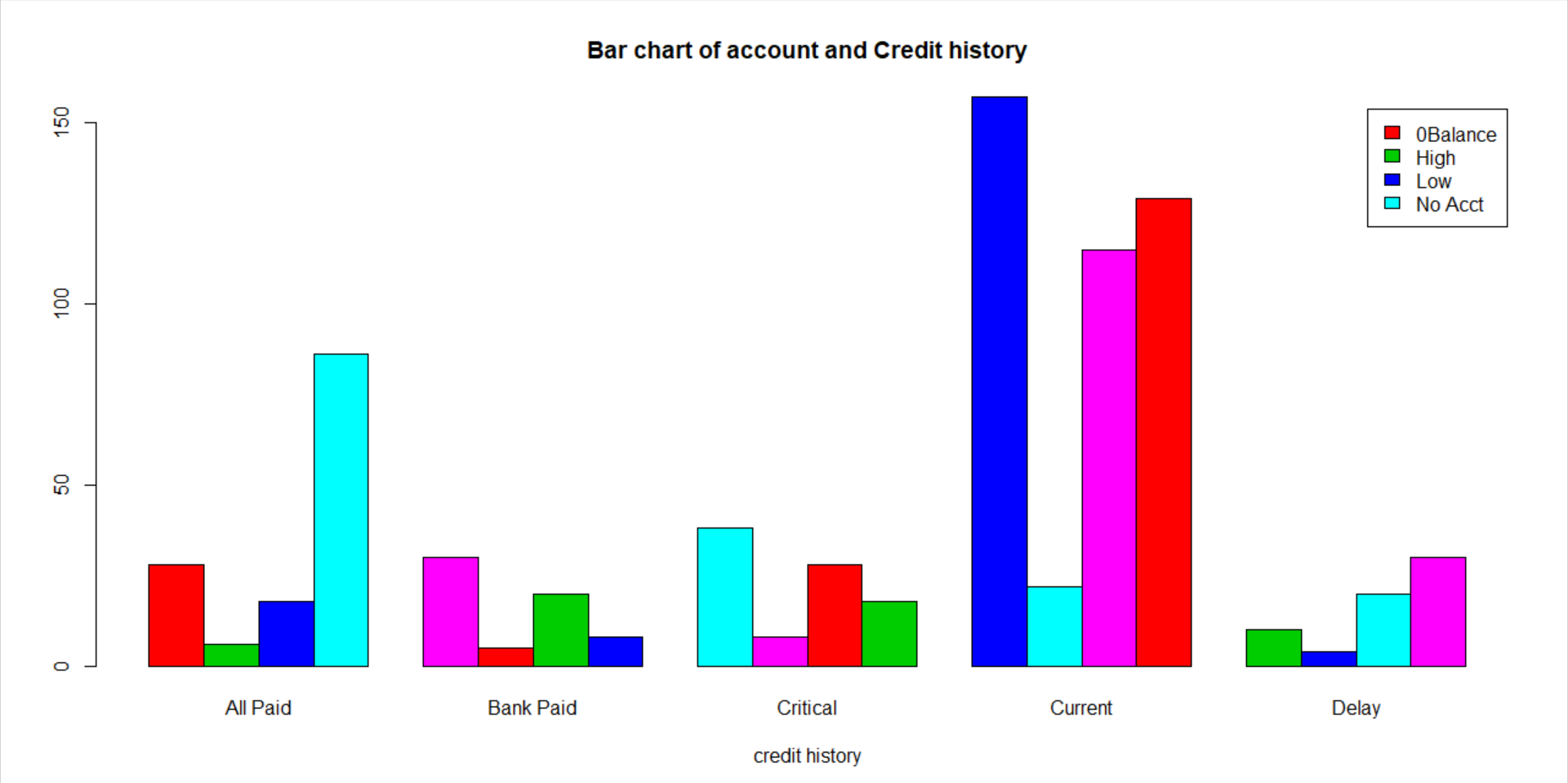
table(data2$Foreign.National)

##   
## No Yes   
## 253 527

table(data2$Credit.Standing)

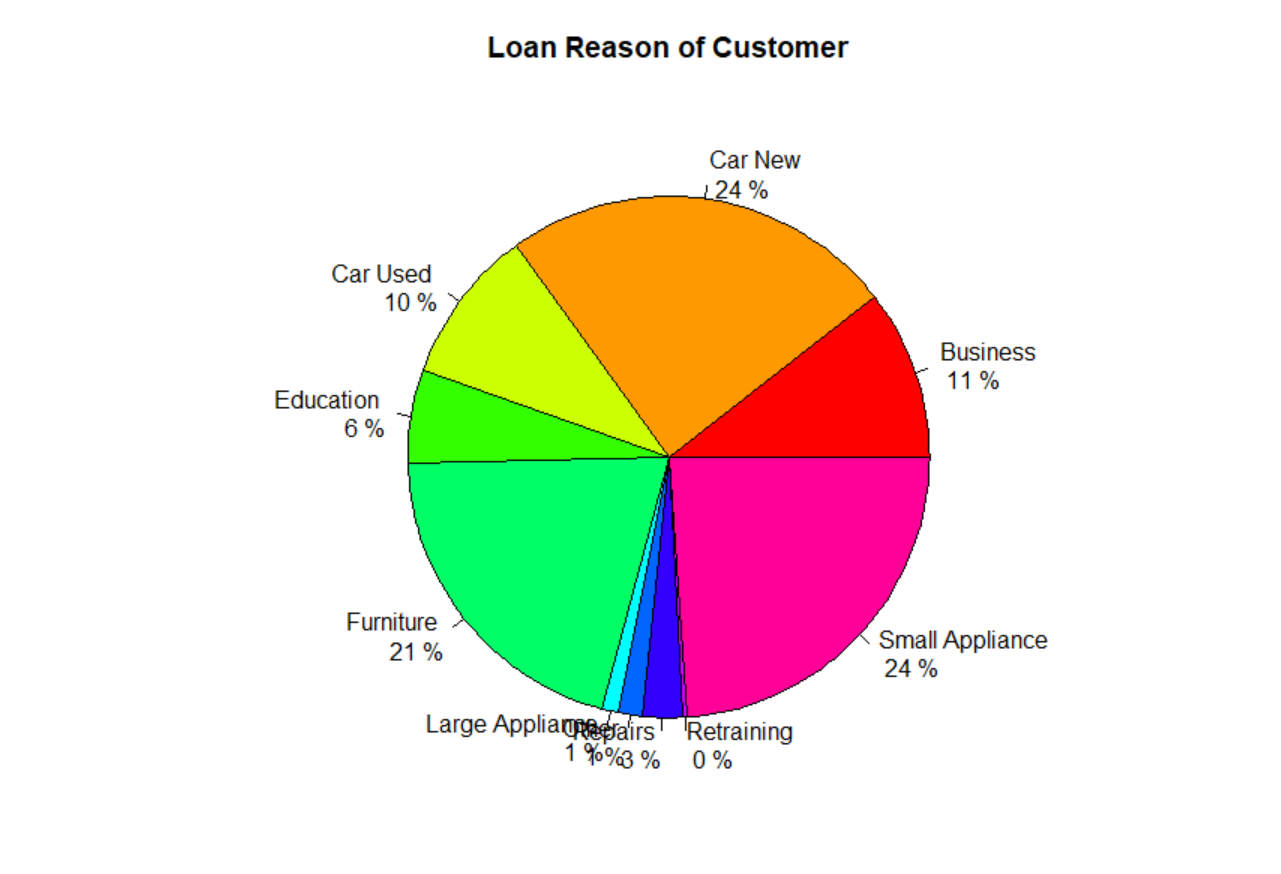
##   
## Bad Good   
## 319 461

##barchart of account and Credit history  
act\_credit<-table(data2$Checking.Acct,data2$Credit.History)  
barplot(act\_credit,main = 'Bar chart of account and Credit history',xlab = 'credit history',col=c(2,3,4,5,6),legend=rownames(act\_credit),beside = TRUE)



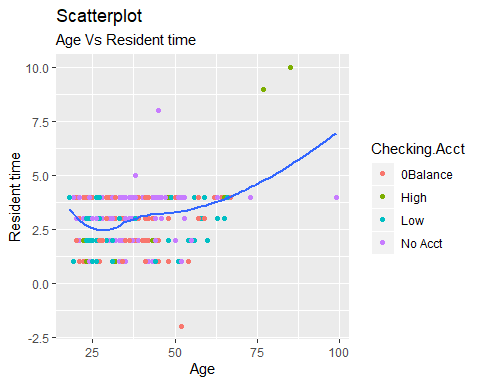
From the above Histogram we can observe most customer are holds current account where in current most customer is having Low balance and after that 0 balance and no account is very low. After that All paid( no credit taken) is second major customer where in most customer holds no account after that 0 balance and High balance customer is very less.

##pie\_chart of loan reason  
Loan\_reason<-table(data2$Loan.Reason)  
lbls<-paste(names(Loan\_reason),"\n",sep=" ")  
pct<-round(Loan\_reason/sum(Loan\_reason)\*100)  
lbls<-paste(lbls,pct,"%",sep=" ")  
pie(Loan\_reason,main = "Loan Reason of Customer",labels = lbls,col = rainbow(length(lbls)))



Pie chart of the Loan reason of the customer where in we can observe that New car and Small Appliance are the belongs to 24% each in the whole category so we can say Major reason for loan is New car, small Appliance and Furniture. Least reason for the Loan Large Appliance , repair and Retraining.

##windows(20,20)  
gg <- ggplot(data2, aes(x=Age, y=`Residence.Time.(In.current.district)`)) +   
 geom\_point(aes(col=Checking.Acct)) + ##draw point  
 geom\_smooth(method="loess", se=F) +   
 labs(subtitle="Age Vs Resident time",   
 y="Resident time",   
 x="Age",   
 title="Scatterplot")##scatterplot  
  
plot(gg)

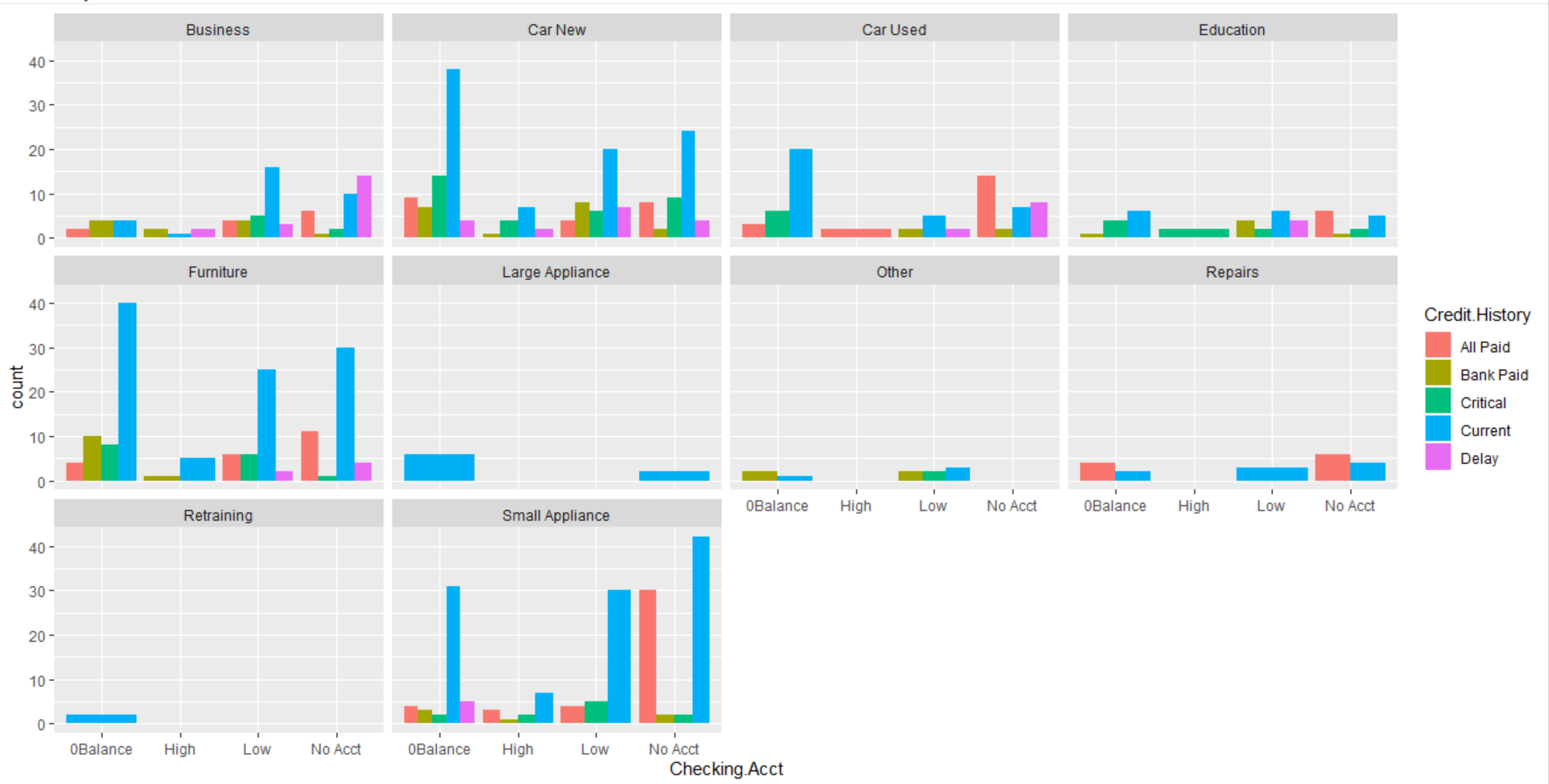


Scatterplot of Age and Resident Time with respect to account checking. We can observe from the Plot that major customer are between resident time 3 and 4.but in 3 and 4 resident time we can observe there is No account customer is very high and 0 balance customers.

colnames(data2)

## [1] "ID"   
## [2] "Checking.Acct"   
## [3] "Credit.History"   
## [4] "Loan.Reason"   
## [5] "Savings.Acct"   
## [6] "Employment"   
## [7] "Personal.Status"   
## [8] "Housing"   
## [9] "Job.Type"   
## [10] "Foreign.National"   
## [11] "Months.since.Checking.Acct.opened"   
## [12] "Residence.Time.(In.current.district)"  
## [13] "Age"   
## [14] "Credit.Standing"

#Trivariate analysis relation between Checking Account, Credit Historyand Loan reason.   
windows(20,20)  
ggplot(data2,aes(x=`Checking.Acct`,group=`Credit.History`,fill=`Credit.History`))+  
 geom\_bar(position='dodge')+facet\_wrap(factor(data2$Loan.Reason))



From the plot it is observed that major reason for loan is New car, Furniture and small Appliances where in it is observed that credit history current and All paid is mojor reason for loan. Least reason for loan is Retraining, Large Appliance and others.

library(ggplot2)  
summary(data2$Age[data2$`Residence.Time.(In.current.district)`== 1])

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 19.00 25.00 32.00 32.11 35.00 54.00

##windows(10,10)  
#install.packages('dplyr')  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(tidyr)  
  
data2 <- data2 %>% drop\_na()##remove NA values from the dataset 2   
data2$`Residence.Time`=data2$`Residence.Time.(In.current.district)`##changin the name of the column

data2$`Residence.Time.(In.current.district)`=NULL##removing dublicate column

**QUESTION B:**

##############################decision treee#####################  
set.seed(158)  
id<-sample(2,nrow(data2),prob = c(0.8,0.2),replace=TRUE)##deviding dataset2 80% as ID=1 and 20% as ID=2  
data2\_train<-data2[id==1,]  
data2\_test<-data2[id==2,]  
##install.packages("rpart")  
library(rpart)  
data2\_model<-rpart(Credit.Standing~.-ID,data=data2\_train)##creating decision tree model using rpart  
data2\_model

Splitting of decision tree on the basis of decision attributes.

## n= 593   
##   
## node), split, n, loss, yval, (yprob)  
## \* denotes terminal node  
##   
## 1) root 593 249 Good (0.41989882 0.58010118)   
## 2) Credit.History=Critical 69 1 Bad (0.98550725 0.01449275) \*  
## 3) Credit.History=All Paid,Bank Paid,Current,Delay 524 181 Good (0.34541985 0.65458015)   
## 6) Credit.History=Current,Delay 375 170 Good (0.45333333 0.54666667)   
## 12) Employment=Short 125 40 Bad (0.68000000 0.32000000)   
## 24) Credit.History=Current 110 29 Bad (0.73636364 0.26363636)   
## 48) Residence.Time>=1.5 101 23 Bad (0.77227723 0.22772277) \*  
## 49) Residence.Time< 1.5 9 3 Good (0.33333333 0.66666667) \*  
## 25) Credit.History=Delay 15 4 Good (0.26666667 0.73333333) \*  
## 13) Employment=Long,Medium,Unemployed,Very Short 250 85 Good (0.34000000 0.66000000)   
## 26) Checking.Acct=0Balance,High,Low 166 70 Good (0.42168675 0.57831325)   
## 52) Savings.Acct=No Acct 16 4 Bad (0.75000000 0.25000000) \*  
## 53) Savings.Acct=High,Low,MedHigh,MedLow 150 58 Good (0.38666667 0.61333333)   
## 106) Months.since.Checking.Acct.opened>=48.5 17 5 Bad (0.70588235 0.29411765) \*  
## 107) Months.since.Checking.Acct.opened< 48.5 133 46 Good (0.34586466 0.65413534) \*  
## 27) Checking.Acct=No Acct 84 15 Good (0.17857143 0.82142857) \*  
## 7) Credit.History=All Paid,Bank Paid 149 11 Good (0.07382550 0.92617450) \*

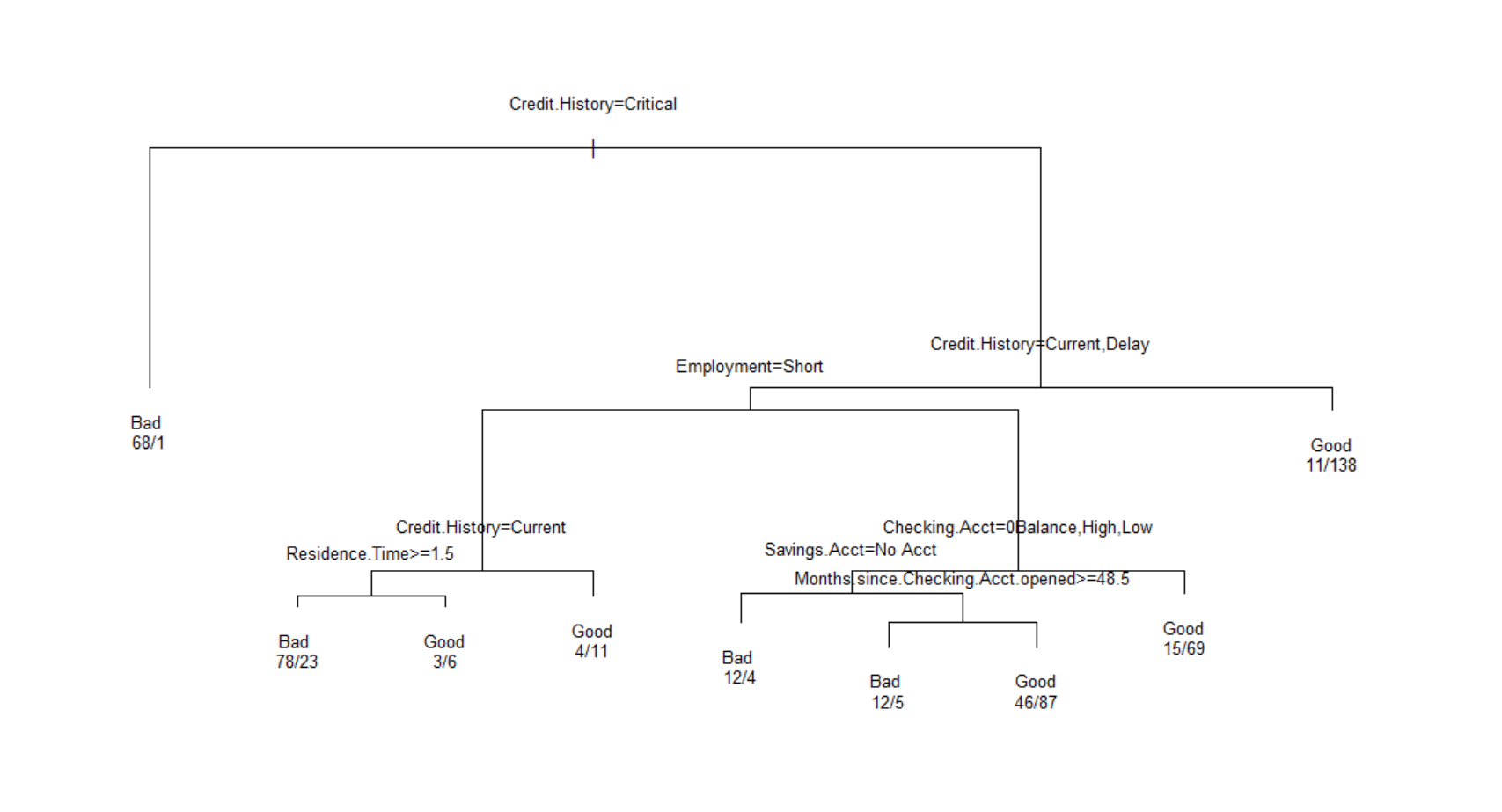
Summary of model gives the significance of variable at each level while building the model.

#install.packages("rpart.plot")  
summary(data2\_model)

## Call:  
## rpart(formula = Credit.Standing ~ . - ID, data = data2\_train)  
## n= 593   
##   
## CP nsplit rel error xerror xstd  
## 1 0.26907631 0 1.0000000 1.0000000 0.04826721  
## 2 0.09036145 1 0.7309237 0.7991968 0.04617936  
## 3 0.02811245 3 0.5502008 0.5502008 0.04122076  
## 4 0.01606426 4 0.5220884 0.5341365 0.04079229  
## 5 0.01204819 7 0.4618474 0.5180723 0.04034848  
## 6 0.01000000 8 0.4497992 0.5180723 0.04034848  
##

Var importance gives the important variables which are used for splitting of the decision tree.   
## Variable importance  
## Credit.History Employment   
## 67 15   
## Checking.Acct Months.since.Checking.Acct.opened   
## 5 4   
## Savings.Acct Residence.Time   
## 4 3   
## Age   
## 1   
##   
## Node number 1: 593 observations, complexity param=0.2690763  
## predicted class=Good expected loss=0.4198988 P(node) =1  
## class counts: 249 344  
## probabilities: 0.420 0.580   
## left son=2 (69 obs) right son=3 (524 obs)  
## Primary splits:  
## Credit.History splits as RRLRR, improve=49.961360, (0 missing)  
## Employment splits as RRLLRR, improve=22.690080, (0 missing)  
## Checking.Acct splits as LLLR, improve=16.610490, (0 missing)  
## Age < 24.5 to the left, improve= 7.471971, (0 missing)  
## Months.since.Checking.Acct.opened < 9.5 to the right, improve= 7.035721, (0 missing)  
##   
## Node number 2: 69 observations  
## predicted class=Bad expected loss=0.01449275 P(node) =0.1163575  
## class counts: 68 1  
## probabilities: 0.986 0.014   
##   
## Node number 3: 524 observations, complexity param=0.09036145  
## predicted class=Good expected loss=0.3454198 P(node) =0.8836425  
## class counts: 181 343  
## probabilities: 0.345 0.655   
## left son=6 (375 obs) right son=7 (149 obs)  
## Primary splits:  
## Credit.History splits as RR-LL, improve=30.715510, (0 missing)  
## Employment splits as RRLLRR, improve=20.709400, (0 missing)  
## Checking.Acct splits as LLLR, improve=10.469140, (0 missing)  
## Age < 24.5 to the left, improve= 6.589929, (0 missing)  
## Months.since.Checking.Acct.opened < 17.5 to the right, improve= 3.679744, (0 missing)  
## Surrogate splits:  
## Residence.Time < 4.5 to the left, agree=0.723, adj=0.027, (0 split)  
## Months.since.Checking.Acct.opened < 5.5 to the right, agree=0.721, adj=0.020, (0 split)  
## Employment splits as LLRLLL, agree=0.719, adj=0.013, (0 split)  
## Loan.Reason splits as LLLLLLRLLL, agree=0.718, adj=0.007, (0 split)  
## Age < 75 to the left, agree=0.718, adj=0.007, (0 split)  
##   
## Node number 6: 375 observations, complexity param=0.09036145  
## predicted class=Good expected loss=0.4533333 P(node) =0.6323777  
## class counts: 170 205  
## probabilities: 0.453 0.547   
## left son=12 (125 obs) right son=13 (250 obs)  
## Primary splits:  
## Employment splits as RR-LRR, improve=19.266670, (0 missing)  
## Checking.Acct splits as LLLR, improve= 7.848881, (0 missing)  
## Months.since.Checking.Acct.opened < 9.5 to the right, improve= 4.944655, (0 missing)  
## Savings.Acct splits as LLRRL, improve= 4.211177, (0 missing)  
## Age < 28.5 to the left, improve= 3.130530, (0 missing)  
## Surrogate splits:  
## Age < 20.5 to the left, agree=0.688, adj=0.064, (0 split)  
## Months.since.Checking.Acct.opened < 55 to the right, agree=0.675, adj=0.024, (0 split)  
## Loan.Reason splits as RRRRRLRRRR, agree=0.669, adj=0.008, (0 split)  
##   
## Node number 7: 149 observations  
## predicted class=Good expected loss=0.0738255 P(node) =0.2512648  
## class counts: 11 138  
## probabilities: 0.074 0.926   
##   
## Node number 12: 125 observations, complexity param=0.02811245  
## predicted class=Bad expected loss=0.32 P(node) =0.2107926  
## class counts: 85 40  
## probabilities: 0.680 0.320   
## left son=24 (110 obs) right son=25 (15 obs)  
## Primary splits:  
## Credit.History splits as ---LR, improve=5.824242, (0 missing)  
## Loan.Reason splits as RRRLLLL--R, improve=3.969707, (0 missing)  
## Checking.Acct splits as LRRR, improve=2.618673, (0 missing)  
## Residence.Time < 1.5 to the right, improve=2.331034, (0 missing)  
## Savings.Acct splits as LLLRL, improve=2.158028, (0 missing)  
##   
## Node number 13: 250 observations, complexity param=0.01606426  
## predicted class=Good expected loss=0.34 P(node) =0.4215852  
## class counts: 85 165  
## probabilities: 0.340 0.660   
## left son=26 (166 obs) right son=27 (84 obs)  
## Primary splits:  
## Checking.Acct splits as LLLR, improve=6.593287, (0 missing)  
## Months.since.Checking.Acct.opened < 41.5 to the right, improve=4.401236, (0 missing)  
## Savings.Acct splits as LLRLL, improve=2.746891, (0 missing)  
## Housing splits as LRR, improve=1.745455, (0 missing)  
## Job.Type splits as LLRL, improve=1.665021, (0 missing)  
## Surrogate splits:  
## Savings.Acct splits as LLRLL, agree=0.716, adj=0.155, (0 split)  
## Months.since.Checking.Acct.opened < 6.5 to the right, agree=0.676, adj=0.036, (0 split)  
## Loan.Reason splits as LLLLLRLLLL, agree=0.672, adj=0.024, (0 split)  
## Age < 70 to the left, agree=0.672, adj=0.024, (0 split)  
##   
## Node number 24: 110 observations, complexity param=0.01204819  
## predicted class=Bad expected loss=0.2636364 P(node) =0.1854975  
## class counts: 81 29  
## probabilities: 0.736 0.264   
## left son=48 (101 obs) right son=49 (9 obs)  
## Primary splits:  
## Residence.Time < 1.5 to the right, improve=3.184338, (0 missing)  
## Loan.Reason splits as LRRLLLL--R, improve=2.377526, (0 missing)  
## Savings.Acct splits as LLLRL, improve=1.792424, (0 missing)  
## Months.since.Checking.Acct.opened < 14.5 to the right, improve=1.468069, (0 missing)  
## Checking.Acct splits as LRLR, improve=1.188083, (0 missing)  
##   
## Node number 25: 15 observations  
## predicted class=Good expected loss=0.2666667 P(node) =0.02529511  
## class counts: 4 11  
## probabilities: 0.267 0.733   
##   
## Node number 26: 166 observations, complexity param=0.01606426  
## predicted class=Good expected loss=0.4216867 P(node) =0.2799325  
## class counts: 70 96  
## probabilities: 0.422 0.578   
## left son=52 (16 obs) right son=53 (150 obs)  
## Primary splits:  
## Savings.Acct splits as RRRRL, improve=3.817189, (0 missing)  
## Months.since.Checking.Acct.opened < 9.5 to the right, improve=3.117189, (0 missing)  
## Age < 41.5 to the right, improve=2.026499, (0 missing)  
## Loan.Reason splits as LRLLR-RLRL, improve=1.337693, (0 missing)  
## Employment splits as LR--LL, improve=1.191758, (0 missing)  
##   
## Node number 27: 84 observations  
## predicted class=Good expected loss=0.1785714 P(node) =0.1416526  
## class counts: 15 69  
## probabilities: 0.179 0.821   
##   
## Node number 48: 101 observations  
## predicted class=Bad expected loss=0.2277228 P(node) =0.1703204  
## class counts: 78 23  
## probabilities: 0.772 0.228   
##   
## Node number 49: 9 observations  
## predicted class=Good expected loss=0.3333333 P(node) =0.01517707  
## class counts: 3 6  
## probabilities: 0.333 0.667   
##   
## Node number 52: 16 observations  
## predicted class=Bad expected loss=0.25 P(node) =0.02698145  
## class counts: 12 4  
## probabilities: 0.750 0.250   
##   
## Node number 53: 150 observations, complexity param=0.01606426  
## predicted class=Good expected loss=0.3866667 P(node) =0.2529511  
## class counts: 58 92  
## probabilities: 0.387 0.613   
## left son=106 (17 obs) right son=107 (133 obs)  
## Primary splits:  
## Months.since.Checking.Acct.opened < 48.5 to the right, improve=3.9073920, (0 missing)  
## Credit.History splits as ---RL, improve=1.6087000, (0 missing)  
## Age < 58 to the right, improve=1.5762370, (0 missing)  
## Employment splits as RR--RL, improve=1.1759020, (0 missing)  
## Loan.Reason splits as LRRLR-RLRL, improve=0.9794102, (0 missing)  
##   
## Node number 106: 17 observations  
## predicted class=Bad expected loss=0.2941176 P(node) =0.02866779  
## class counts: 12 5  
## probabilities: 0.706 0.294   
##   
## Node number 107: 133 observations  
## predicted class=Good expected loss=0.3458647 P(node) =0.2242833  
## class counts: 46 87  
## probabilities: 0.346 0.654

#varImp(data2\_model)  
  
library(rpart.plot)  
plot(data2\_model,margin = 0.1)##plotting the decision tree model   
text(data2\_model,use.n = TRUE,pretty = 0,cex=0.8)#addding the text into decision tree model



From the Tree it can be Referred that 1st split will happen on the basis of credit history root node when Credit History is Critical then tree will move to left hand side predict Bad Credit standing.

If not then tree will move to right hand side, it will again check credit history if it will other than Current and delay tree will move to right hand side Predict Good, if not tree will move to left hand side it will check for Employment. If Employment is short then tree will move to left hand side, it will check for credit history equal to current then it will check for resident time if its greater than 1.5 then it will predict bad if not then tree will predict Good, if Credit History is not Current then Tree will predict Good.

If Employment is not short then it will check for Checking Account , if checking Account is 0 balance, high and Low then it will check for Saving Account, if saving account is no account Tree will predict Bad, if saving account is not No Account it will check for Months since Checking account, if months since Checking account is greater than 48.5 tree will predict Bad, if not then tree will predict Good.

##install.packages('caret', dependencies = TRUE)  
library(caret)

## Loading required package: lattice

confusionMatrix(table(data2\_pred,data2\_test$Credit.Standing))##creating confusionmatrix to find the accuracy of the model.

## Confusion Matrix and Statistics  
##   
##   
## data2\_pred Bad Good  
## Bad 30 13  
## Good 26 75  
##   
## Accuracy : 0.7292   
## 95% CI : (0.6489, 0.7998)  
## No Information Rate : 0.6111   
## P-Value [Acc > NIR] : 0.001989   
##   
## Kappa : 0.4051   
##   
## Mcnemar's Test P-Value : 0.054664   
##   
## Sensitivity : 0.5357   
## Specificity : 0.8523   
## Pos Pred Value : 0.6977   
## Neg Pred Value : 0.7426   
## Prevalence : 0.3889   
## Detection Rate : 0.2083   
## Detection Prevalence : 0.2986   
## Balanced Accuracy : 0.6940   
##   
## 'Positive' Class : Bad   
##

data1\_pred<-predict(data2\_model,newdata = data1,type = 'class')#predicting on Scoring dataset  
data1\_pred

## 1 2 3 4 5 6 7 8 9 10 11 12 13   
## Good Good Good Good Good Bad Good Good Good Good Bad Bad Bad   
## Levels: Bad Good

**QUESTION C:**

If Checking Account is not 0 balance, high and low then tree will predict Good.



As per the customer ID 788 the tree will on 1st split check for Credit History equal to critical but here credit history is All paid, tree will move to right hand side again it will check for Credit History but in our case credit history is All paid the tree will move to right hand side and will predict GOOD.



As per customer ID 789 the tree will on 1st split check for Credit history but in our case credit history is current, so tree will move to right hand side again it will check for credit history equal to current and delayed but here credit history is current then tree will move to left hand side and will check for employment but here employment is short tree will move to right hand side and again check for checking account here checking account is no acct so tree will move to right hand side will predict Good.



As per the customer ID 790 the tree will on 1st split check for Credit History equal to critical but here credit history is Current, tree will move to right hand side again it will check for Credit History but in our case credit history is Current the tree will move to left side and will Check for Employment but here Employment is Medium the tree will move to right hand side and again check for checking account here checking account is no acct so tree will move to right hand side will predict Good.



As per the customer ID 791 the tree will on 1st split check for Credit History equal to critical and here credit history is Critical then tree will move to left hand side and predict Bad.



As per the customer ID 792 the tree will on 1st split check for Credit History equal to critical and here credit history is Critical then tree will move to left hand side and predict Bad.

**QUESTION D:**

################################randomforest############################  
controlParameter<-trainControl(method = " CV",##crossvalidation  
 number = 5,#5 partition of the data set  
 savePredictions = TRUE,##saves prediction  
 classProbs=TRUE#save probability by model  
 )  
ParameterGrid<-expand.grid(mtry=c(2,3,4))# mtry is only parameter which can be tuned in RF.  
  
library(caret)  
  
modelRandom<-train(Credit.Standing~.-ID,##train is main function   
 data = data2\_train,  
 method="rf",##random forest  
 #trControl=controlParameter,  
 trueGrid=ParameterGrid  
)  
modelRandom

## Random Forest   
##   
## 593 samples  
## 13 predictor  
## 2 classes: 'Bad', 'Good'   
##   
## No pre-processing  
## Resampling: Bootstrapped (25 reps)   
## Summary of sample sizes: 593, 593, 593, 593, 593, 593, ...   
## Resampling results across tuning parameters:  
##   
## mtry Accuracy Kappa   
## 2 0.7322384 0.4256761  
## 19 0.7505211 0.4838779  
## 36 0.7454965 0.4740077  
##   
## Accuracy was used to select the optimal model using the largest value.  
## The final value used for the model was mtry = 19.

predi<-predict(modelRandom,newdata = data2\_test)##prediction on test datset

confusionMatrix(table(predi,data2\_test$Credit.Standing))#confusion matrix to check accuracy

## Confusion Matrix and Statistics  
##   
##   
## predi Bad Good  
## Bad 33 21  
## Good 23 67  
##   
## Accuracy : 0.6944   
## 95% CI : (0.6123, 0.7684)  
## No Information Rate : 0.6111   
## P-Value [Acc > NIR] : 0.02339   
##   
## Kappa : 0.3529   
##   
## Mcnemar's Test P-Value : 0.88017   
##   
## Sensitivity : 0.5893   
## Specificity : 0.7614   
## Pos Pred Value : 0.6111   
## Neg Pred Value : 0.7444   
## Prevalence : 0.3889   
## Detection Rate : 0.2292   
## Detection Prevalence : 0.3750   
## Balanced Accuracy : 0.6753   
##   
## 'Positive' Class : Bad   
##

main\_pred<-predict(modelRandom,newdata = data1) ##prediction on scoring dataset   
main\_pred

## [1] Good Good Good Good Good Good Good Good Good Good Bad Bad Bad   
## Levels: Bad Good

summary(modelRandom)##summary of the model

## Length Class Mode   
## call 5 -none- call   
## type 1 -none- character  
## predicted 593 factor numeric   
## err.rate 1500 -none- numeric   
## confusion 6 -none- numeric   
## votes 1186 matrix numeric   
## oob.times 593 -none- numeric   
## classes 2 -none- character  
## importance 36 -none- numeric   
## importanceSD 0 -none- NULL   
## localImportance 0 -none- NULL   
## proximity 0 -none- NULL   
## ntree 1 -none- numeric   
## mtry 1 -none- numeric   
## forest 14 -none- list   
## y 593 factor numeric   
## test 0 -none- NULL   
## inbag 0 -none- NULL   
## xNames 36 -none- character  
## problemType 1 -none- character  
## tuneValue 1 data.frame list   
## obsLevels 2 -none- character  
## param 1 -none- list

varImp(modelRandom)##shows the importance of variables for model.

## rf variable importance  
##   
## only 20 most important variables shown (out of 36)  
##   
## Overall  
## Credit.HistoryCritical 100.000  
## Age 71.945  
## Months.since.Checking.Acct.opened 51.801  
## EmploymentShort 45.605  
## Credit.HistoryCurrent 40.842  
## Foreign.NationalYes 28.277  
## Residence.Time 26.905  
## Checking.AcctNo Acct 24.940  
## Savings.AcctLow 15.326  
## Credit.HistoryDelay 15.053  
## Savings.AcctNo Acct 10.349  
## EmploymentMedium 10.189  
## Personal.StatusSingle 10.138  
## Job.TypeSkilled 10.115  
## Loan.ReasonCar New 9.510  
## HousingOwn 9.077  
## Loan.ReasonSmall Appliance 8.906  
## Loan.ReasonFurniture 8.738  
## Savings.AcctMedLow 8.516  
## Personal.StatusMarried 8.415

################################converting into factor#############################  
names(data2\_train)=gsub("\\.","\_",names(data2\_train))##to remove specialcharecter from the attributes name.  
names(data2\_test)=gsub("\\.","\_",names(data2\_test))  
names(data1)=gsub("\\.","\_",names(data1))  
data2\_train$Credit\_Standing[data2\_train$Credit\_Standing=='Good']=1##converting good as 1  
data2\_train$Credit\_Standing[data2\_train$Credit\_Standing=='Bad']=0##converting bad as 0  
data2\_test$Credit\_Standing[data2\_test$Credit\_Standing=='Good']=1  
data2\_test$Credit\_Standing[data2\_test$Credit\_Standing=='Bad']=0  
data2\_train$Credit\_Standing<-as.numeric(data2\_train$Credit\_Standing)##converting attributes into numeric daat  
data2\_test$Credit\_Standing<-as.numeric(data2\_test$Credit\_Standing)  
data2\_train$Checking\_Acct<-as.factor(data2\_train$Checking\_Acct)##converting into factor  
data2\_test$Checking\_Acct<-as.factor(data2\_test$Checking\_Acct)  
data2\_train$Credit\_History<-as.factor(data2\_train$Credit\_History)  
data2\_test$Credit\_History<-as.factor(data2\_test$Credit\_History)  
data2\_train$Loan\_Reason<-as.factor(data2\_train$Loan\_Reason)  
data2\_test$Loan\_Reason<-as.factor(data2\_test$Loan\_Reason)  
data2\_train$Savings\_Acct<-as.factor(data2\_train$Savings\_Acct)  
data2\_test$Savings\_Acct<-as.factor(data2\_test$Savings\_Acct)  
data2\_train$Employment<-as.factor(data2\_train$Employment)  
data2\_test$Employment<-as.factor(data2\_test$Employment)  
data2\_train$Personal\_Status<-as.factor(data2\_train$Personal\_Status)  
data2\_test$Personal\_Status<-as.factor(data2\_test$Personal\_Status)  
data2\_train$Housing<-as.factor(data2\_train$Housing)  
data2\_test$Housing<-as.factor(data2\_test$Housing)  
data2\_train$Job\_Type<-as.factor(data2\_train$Job\_Type)  
data2\_test$Job\_Type<-as.factor(data2\_test$Job\_Type)  
data2\_train$Foreign\_National<-as.factor(data2\_train$Foreign\_National)  
data2\_test$Foreign\_National<-as.factor(data2\_test$Foreign\_National)  
View(data2\_train)  
str(data2\_train)##structure of the data

## 'data.frame': 593 obs. of 14 variables:  
## $ ID : num 1 3 5 6 7 8 10 11 12 13 ...  
## $ Checking\_Acct : Factor w/ 4 levels "0Balance","High",..: 4 1 4 3 1 3 4 3 1 3 ...  
## $ Credit\_History : Factor w/ 5 levels "All Paid","Bank Paid",..: 1 4 1 4 1 2 5 3 4 1 ...  
## $ Loan\_Reason : Factor w/ 10 levels "Business","Car New",..: 2 2 10 2 2 4 3 5 5 5 ...  
## $ Savings\_Acct : Factor w/ 5 levels "High","Low","MedHigh",..: 2 5 5 4 2 2 2 2 2 2 ...  
## $ Employment : Factor w/ 6 levels "Long","Medium",..: 2 1 1 6 1 2 4 6 4 1 ...  
## $ Personal\_Status : Factor w/ 3 levels "Divorced","Married",..: 3 1 3 1 2 1 2 1 3 3 ...  
## $ Housing : Factor w/ 3 levels "Other","Own",..: 2 2 1 2 2 1 3 3 2 3 ...  
## $ Job\_Type : Factor w/ 4 levels "Management","Skilled",..: 1 2 2 4 2 4 2 2 2 2 ...  
## $ Foreign\_National : Factor w/ 2 levels "No","Yes": 1 1 2 1 2 1 2 2 1 2 ...  
## $ Months\_since\_Checking\_Acct\_opened: num 7 25 7 13 22 25 13 13 19 13 ...  
## $ Age : num 44 28 35 22 29 33 40 24 41 27 ...  
## $ Credit\_Standing : num 1 0 1 1 1 1 1 0 0 1 ...  
## $ Residence\_Time : num 3 2 4 2 3 4 4 3 3 4 ...

str(data2\_test)

## 'data.frame': 144 obs. of 14 variables:  
## $ ID : num 2 14 23 30 31 40 42 62 63 68 ...  
## $ Checking\_Acct : Factor w/ 4 levels "0Balance","High",..: 1 1 3 3 3 2 4 2 1 4 ...  
## $ Credit\_History : Factor w/ 5 levels "All Paid","Bank Paid",..: 4 4 4 4 1 3 5 3 1 1 ...  
## $ Loan\_Reason : Factor w/ 9 levels "Business","Car New",..: 2 2 3 2 2 2 3 9 8 8 ...  
## $ Savings\_Acct : Factor w/ 5 levels "High","Low","MedHigh",..: 2 2 5 2 2 3 5 5 4 5 ...  
## $ Employment : Factor w/ 5 levels "Long","Medium",..: 3 2 3 1 5 2 2 3 1 2 ...  
## $ Personal\_Status : Factor w/ 3 levels "Divorced","Married",..: 1 1 3 3 3 2 3 3 3 1 ...  
## $ Housing : Factor w/ 3 levels "Other","Own",..: 2 2 2 1 2 2 2 1 2 2 ...  
## $ Job\_Type : Factor w/ 4 levels "Management","Skilled",..: 2 2 2 1 4 2 1 2 2 2 ...  
## $ Foreign\_National : Factor w/ 2 levels "No","Yes": 1 2 2 2 2 2 2 2 1 1 ...  
## $ Months\_since\_Checking\_Acct\_opened: num 16 13 13 37 13 13 37 16 10 13 ...  
## $ Age : num 28 35 29 56 48 27 30 38 47 22 ...  
## $ Credit\_Standing : num 0 0 1 1 1 0 1 0 1 1 ...  
## $ Residence\_Time : num 2 3 2 2 3 2 2 4 4 4 ...

names(data2\_train)=gsub("\\.","\_",names(data2\_train))

##################################boosting##########################################  
##install.packages("gbm")  
library(gbm)

## Loaded gbm 2.1.5

boost\_data<-gbm(Credit\_Standing~.,data = data2\_train,distribution = "bernoulli",  
 n.trees = 500,interaction.depth = 4)##boosting model of 500 tree and depth of the as 4  
boost\_data

## gbm(formula = Credit\_Standing ~ ., distribution = "bernoulli",   
## data = data2\_train, n.trees = 500, interaction.depth = 4)  
## A gradient boosted model with bernoulli loss function.  
## 500 iterations were performed.  
## There were 13 predictors of which 13 had non-zero influence.

pred\_boost<-predict(boost\_data,newdata=data2\_test,n.trees = 500,type = 'response')#predition on testing dataset  
data1\_boostpred<-predict(boost\_data,newdata = data1,n.trees=500,type = 'response')#prediction on Scoring dataset

data1\_boostpred

## [1] 0.4880105 0.4906472 0.4403342 0.6491348 0.6923898 0.7276339 0.5539754  
## [8] 0.4941551 0.6311227 0.7052307 0.5575164 0.5972338 0.5017076

##boost\_data$data$y  
pred\_class<-ifelse(pred\_boost<0.5,"No","Yes") #creating prediction class based on probability   
table(data2\_test$Credit\_Standing,pred\_class)

## pred\_class  
## No Yes  
## 0 36 20  
## 1 18 70

Boosting makes multiple trees in series, after finishing the 1st tree then it uses the model to predict on the data which was not selected in 1st iteration, some of the data prediction works well, but for some it doesn’t those data point more likely to be selected during second training set, as long as keep going on each time adjusting model based on Previous prediction and those bad prediction most often come in 2nd prediction, Gradually number of sequences it improves the model.

QUESTION E:

data2\_pred<-predict(data2\_model,newdata = data2\_test,type='class')#predicting on testing dataset  
datamainpred<-predict(data2\_model,newdata = data2,type = 'class')##predicting on main dataset  
data2$datamainpred<-datamainpred  
#View(data2)  
#write.csv(data2,"Question\_E.csv")##creating new csv file for Question number E





To find suspiciously incorrect pattern in Credit standing it has been observed that from ID number 624 to 633 there is 10 consecutive IDS where pattern is found to be incorrect. To find the incorrect pattern in the dataset 1st step is to get the predicted credit standing in the main data set for that datamainpred column is created in the main dataset and predicted value is assigned to the datamainpred column. After that in data frame there is one reference credit standing and one predicted Credit Standing to check what’s the actual credit standing and predicted Credit. Then checked pattern in the dataset where predicted value is different from the actual Credit standing.