Final Code

#Loading nessesary packages

library(MASS)

library(car)

library(dplyr)

library(tidyr)

library(ggplot2)

library(caret)

library(e1071)

library(randomForest)

library(DMwR)

library(gridExtra)

library(Information)

library(scorecard)

library(cowplot)

library(caTools)

#Loading Data

demographic <- read.csv("Demographic data.csv",header = TRUE, na.strings = "NA",stringsAsFactors = FALSE)

credit\_bureau <- read.csv("Credit Bureau data.csv",header = TRUE, na.strings = "NA", stringsAsFactors = FALSE)

# having a peek at the structure of data inside both the files

str(demographic)

str(credit\_bureau)

# business understanding

# write something about the data present in both the files

# Application Id seems to be a common key between both the datasets, so we will be

# collating the data on Application Id

# making a pre-check whether the Application Id is same in both the Data Sets

length(unique(demographic$Application.ID)) # 71292 suggests that there are present 3 duplicates as well

length(unique(credit\_bureau$Application.ID)) # 71292 suggests that there are present 3 duplicates as well

setdiff(demographic$Application.ID,credit\_bureau$Application.ID)

setdiff(demographic$Performance.Tag,credit\_bureau$Performance.Tag)

# 0 suggests that Application Ids and Performance Tag are identical in both the data sets

# merging both the data sets around the Application Id and Performance Tag column

demo\_cb\_merged<- merge(demographic,credit\_bureau)

length(unique(demo\_cb\_merged$Application.ID))

# 71292 suggests there are present some duplicate values

# having a peek at the structure of the merged file

str(demo\_cb\_merged)

# write about the columns and features available in the file

# checking the performance tag column

sum(demo\_cb\_merged$Performance.Tag,na.rm = TRUE)/nrow(demo\_cb\_merged)

sum(demo\_cb\_merged$Performance.Tag,na.rm = TRUE)

# only 4% of the values have Performance Tag are one ie 2984/71295

# Following are the continuous variables

# "Age","No.of.dependents","Income","No.of.months.in.current.residence",

# "No.of.months.in.current.company","No.of.times.90.DPD.or.worse.in.last.6.months",

# "No.of.times.60.DPD.or.worse.in.last.6.months","No.of.times.30.DPD.or.worse.in.last.6.months",

# "No.of.times.90.DPD.or.worse.in.last.12.months","No.of.times.60.DPD.or.worse.in.last.12.months",

# "No.of.times.30.DPD.or.worse.in.last.12.months","Avgas.CC.Utilization.in.last.12.months",

# "No.of.trades.opened.in.last.6.months","No.of.trades.opened.in.last.12.months",

# "No.of.PL.trades.opened.in.last.6.months","No.of.PL.trades.opened.in.last.12.months"

# "No.of.Inquiries.in.last.6.months..excluding.home...auto.loans.",

# "No.of.Inquiries.in.last.12.months..excluding.home...auto.loans.",

# "Presence.of.open.home.loan","Outstanding.Balance","Total.No.of.Trades",

# "Presence.of.open.auto.loan"

continuous <- c("Age","No.of.dependents","Income","No.of.months.in.current.residence",

"No.of.months.in.current.company","No.of.times.90.DPD.or.worse.in.last.6.months",

"No.of.times.60.DPD.or.worse.in.last.6.months","No.of.times.30.DPD.or.worse.in.last.6.months",

"No.of.times.90.DPD.or.worse.in.last.12.months","No.of.times.60.DPD.or.worse.in.last.12.months",

"No.of.times.30.DPD.or.worse.in.last.12.months","Avgas.CC.Utilization.in.last.12.months",

"No.of.trades.opened.in.last.6.months","No.of.trades.opened.in.last.12.months",

"No.of.PL.trades.opened.in.last.6.months","No.of.PL.trades.opened.in.last.12.months",

"No.of.Inquiries.in.last.6.months..excluding.home...auto.loans.",

"No.of.Inquiries.in.last.12.months..excluding.home...auto.loans.",

"Outstanding.Balance","Total.No.of.Trades")

# Following are the categorical variables

# "Gender","Marital.Status..at.the.time.of.application.","Education"

# "Profession","Type.of.residence","Presence.of.open.home.loan","Presence.of.open.auto.loan"

categorical <- c("Gender","Marital.Status..at.the.time.of.application.","Education",

"Profession","Type.of.residence","Presence.of.open.home.loan","Presence.of.open.auto.loan")

demo\_cb\_continuous <- cbind(demo\_cb\_merged[,continuous],demo\_cb\_merged$Performance.Tag)

demo\_cb\_categorical <- cbind(demo\_cb\_merged[,categorical],demo\_cb\_merged$Performance.Tag)

library(gdata)

demo\_cb\_continuous <- rename.vars(demo\_cb\_continuous, from = "demo\_cb\_merged$Performance.Tag", to = "Performance.Tag")

demo\_cb\_categorical <- rename.vars(demo\_cb\_categorical, from = "demo\_cb\_merged$Performance.Tag", to = "Performance.Tag")

str(demo\_cb\_categorical)

# "Presence.of.open.home.loan","Presence.of.open.auto.loan" needs to be changed to

# factors

demo\_cb\_categorical$Presence.of.open.auto.loan<- ifelse(demo\_cb\_categorical$Presence.of.open.auto.loan==1,"Yes","No")

demo\_cb\_categorical$Presence.of.open.home.loan<- ifelse(demo\_cb\_categorical$Presence.of.open.home.loan==1,"Yes","No")

str(demo\_cb\_categorical)

###################################################################################

########################## Data Preparation and EDA ###############################

# there were found duplicate Application Ids, lets find those and remove those

# de-duplicating the data

sum(duplicated(demo\_cb\_merged$Application.ID))

# 7 in both the data combined

duplicates <- demo\_cb\_merged$Application.ID[duplicated(demo\_cb\_merged$Application.ID)]

duplicates

# following are the duplicates

# 653287861 671989187 671989187 671989187 765011468 765011468 765011468

demo\_cb\_merged <- demo\_cb\_merged[-which(duplicated(demo\_cb\_merged$Application.ID)),]

sum(duplicated(demo\_cb\_merged$Application.ID))

# 0 , means all the duplicates are removed

# univariate analysis

# Histogram and Boxplots for numeric variables

box\_theme\_x<- theme(axis.line=element\_blank(),axis.title=element\_blank(),

axis.ticks=element\_blank(), axis.text=element\_blank())

box\_theme\_y<- theme(axis.line.y=element\_blank(),axis.title.y=element\_blank(),

axis.ticks.y=element\_blank(), axis.text.y=element\_blank(),

legend.position="none")

plot\_fun\_continuous <- function(cont\_col\_name,var\_name){

plot\_grid(ggplot(demo\_cb\_continuous, aes(cont\_col\_name))+ geom\_histogram(binwidth = 10) + labs(x = var\_name),

ggplot(demo\_cb\_continuous, aes(x="",y=cont\_col\_name))+ geom\_boxplot(width=0.1)+coord\_flip()+box\_theme\_x,

align = "v",ncol = 1)

}

plot\_fun\_continuous(demo\_cb\_continuous$Age,"Age")

plot\_fun\_continuous(demo\_cb\_continuous$No.of.dependents,"No.of.dependents")

plot\_fun\_continuous(demo\_cb\_continuous$Income,"Income")

plot\_fun\_continuous(demo\_cb\_continuous$No.of.months.in.current.residence,"No.of.months.in.current.residence")

plot\_fun\_continuous(demo\_cb\_continuous$No.of.months.in.current.company,"No.of.months.in.current.company")

plot\_fun\_continuous(demo\_cb\_continuous$No.of.times.90.DPD.or.worse.in.last.6.months,"No.of.times.90.DPD.or.worse.in.last.6.months")

plot\_fun\_continuous(demo\_cb\_continuous$No.of.times.60.DPD.or.worse.in.last.6.months,"No.of.times.60.DPD.or.worse.in.last.6.months")

plot\_fun\_continuous(demo\_cb\_continuous$No.of.times.30.DPD.or.worse.in.last.6.months,"No.of.times.30.DPD.or.worse.in.last.6.months")

plot\_fun\_continuous(demo\_cb\_continuous$No.of.times.90.DPD.or.worse.in.last.12.months,"No.of.times.90.DPD.or.worse.in.last.12.months")

plot\_fun\_continuous(demo\_cb\_continuous$No.of.times.60.DPD.or.worse.in.last.12.months,"No.of.times.60.DPD.or.worse.in.last.12.months")

plot\_fun\_continuous(demo\_cb\_continuous$No.of.times.30.DPD.or.worse.in.last.12.months,"No.of.times.30.DPD.or.worse.in.last.12.months")

plot\_fun\_continuous(demo\_cb\_continuous$Avgas.CC.Utilization.in.last.12.months,"Avgas.CC.Utilization.in.last.12.months")

plot\_fun\_continuous(demo\_cb\_continuous$No.of.trades.opened.in.last.6.months,"No.of.trades.opened.in.last.6.months")

plot\_fun\_continuous(demo\_cb\_continuous$No.of.trades.opened.in.last.12.months,"No.of.trades.opened.in.last.12.months")

plot\_fun\_continuous(demo\_cb\_continuous$No.of.PL.trades.opened.in.last.6.months,"No.of.PL.trades.opened.in.last.6.months")

plot\_fun\_continuous(demo\_cb\_continuous$No.of.PL.trades.opened.in.last.12.months,"No.of.PL.trades.opened.in.last.12.months")

plot\_fun\_continuous(demo\_cb\_continuous$No.of.Inquiries.in.last.6.months..excluding.home...auto.loans.,"No.of.Inquiries.in.last.6.months..excluding.home...auto.loans")

plot\_fun\_continuous(demo\_cb\_continuous$No.of.Inquiries.in.last.12.months..excluding.home...auto.loans.,"No.of.Inquiries.in.last.12.months..excluding.home...auto.loans")

plot\_fun\_continuous(demo\_cb\_continuous$Outstanding.Balance,"Outstanding.Balance")

plot\_fun\_continuous(demo\_cb\_continuous$Total.No.of.Trades,"Total.No.of.Trades")

# for categorical variables

bar\_theme1<- theme(axis.text.x = element\_text(angle = 60, hjust = 1, vjust = 0.5),

legend.position="none")

plot\_grid(ggplot(demo\_cb\_categorical, aes(x=Gender,fill=factor(Performance.Tag)))+ geom\_bar(),

ggplot(demo\_cb\_categorical, aes(x=Marital.Status..at.the.time.of.application.,fill=factor(Performance.Tag)))+ geom\_bar(),

ggplot(demo\_cb\_categorical, aes(x=Education,fill=factor(Performance.Tag)))+ geom\_bar())

plot\_grid(ggplot(demo\_cb\_categorical, aes(x=Profession,fill=factor(Performance.Tag)))+ geom\_bar(),

ggplot(demo\_cb\_categorical, aes(x=Type.of.residence,fill=factor(Performance.Tag)))+ geom\_bar(),

ggplot(demo\_cb\_categorical, aes(x=Presence.of.open.home.loan,fill=factor(Performance.Tag)))+ geom\_bar(),

ggplot(demo\_cb\_categorical, aes(x=Presence.of.open.auto.loan,fill=factor(Performance.Tag)))+ geom\_bar())

# outliers and blank values are present in some of the columns

# checking for outliers using the quantiles

quantiles\_df <- sapply(demo\_cb\_merged[,continuous],

function(x) quantile(x,seq(0,1,.01),na.rm = T))

# only the age and income column have the outliers, containing negative values

nrow(demo\_cb\_merged[which(demo\_cb\_merged$Age <= 0),])/nrow(demo\_cb\_merged)

# 0.028% of the values are <=0, so we can remove them

# removing the negative and 0 values in age column

demo\_cb\_merged <- demo\_cb\_merged[-which(demo\_cb\_merged$Age <= 0),]

# for income column

nrow(demo\_cb\_merged[which(demo\_cb\_merged$Income <= 0),])/nrow(demo\_cb\_merged)

# 0.15% are <= 0, so we can safely remove these

demo\_cb\_merged <- demo\_cb\_merged[-which(demo\_cb\_merged$Income <= 0),]

#checking for the NAs in the data

sapply(demo\_cb\_merged, function(x) sum(is.na(x)))

# performance tag has 1425

# No. of Dependents have 3

# Avgas.CC.Utilization.in.last.12.months has 1058

# No.of.trades.opened.in.last.6.months has 1

# Presence.of.open.home.loan has 272

# Outstanding.Balance has 272

sum(is.na(demo\_cb\_merged))/nrow(demo\_cb\_merged)

# 4% of the total values are NAs hence we can safely remove all the rows containing NAs

# removing all the NA values

demo\_cb\_merged <- demo\_cb\_merged[!is.na(demo\_cb\_merged$Performance.Tag),]

demo\_cb\_merged <- demo\_cb\_merged[!is.na(demo\_cb\_merged$Presence.of.open.home.loan),]

demo\_cb\_merged <- demo\_cb\_merged[!is.na(demo\_cb\_merged$Outstanding.Balance),]

demo\_cb\_merged <- demo\_cb\_merged[!is.na(demo\_cb\_merged$No.of.trades.opened.in.last.6.months),]

demo\_cb\_merged <- demo\_cb\_merged[!is.na(demo\_cb\_merged$Avgas.CC.Utilization.in.last.12.months),]

demo\_cb\_merged <- demo\_cb\_merged[!is.na(demo\_cb\_merged$No.of.dependents),]

# all the NAs are removed now

#checking for the missing values

sapply(demo\_cb\_categorical, function(x) table(x))

# Gender has 1

# Marital status has 5

# Education has 117

# Profession has 11

# type of residence has 8

# taking an assumption that imputing 1-2% of the data won't interfere with the Models performance

demo\_cb\_merged$Gender[which(demo\_cb\_merged$Gender == "")] <- "M"

demo\_cb\_merged$Education[which(demo\_cb\_merged$Education == "")] <- "Others"

demo\_cb\_merged$Marital.Status..at.the.time.of.application.[which(demo\_cb\_merged$Marital.Status..at.the.time.of.application. == "")] <- "Married"

demo\_cb\_merged$Profession[which(demo\_cb\_merged$Profession == "")] <- "SAL"

demo\_cb\_merged$Type.of.residence[which(demo\_cb\_merged$Type.of.residence == "")] <- "Others"

# binning the data into appropriate groups rather than considering the values present

# in them as outliers. eg: average cc utilization has values exceeding 100

# which as per the boxplots and the quantile will fall under the category of outliers

# but this could be a real life scenario where a user is excceding his card limit often

## Age binning

#demo\_cb\_merged$bin.age <- demo\_cb\_merged$Age

#demo\_cb\_merged$bin.age <- ifelse((demo\_cb\_merged$Age>=10 & demo\_cb\_merged$Age<=30) , 'Youngsters',demo\_cb\_merged$bin.age)

#demo\_cb\_merged$bin.age <- ifelse((demo\_cb\_merged$Age>30 & demo\_cb\_merged$Age<=55) , 'Adults',demo\_cb\_merged$bin.age)

#demo\_cb\_merged$bin.age <- ifelse((demo\_cb\_merged$Age>55) , 'Senior\_Citizen',demo\_cb\_merged$bin.age)

#table(demo\_cb\_merged$bin.age)

#

#ggplot(demo\_cb\_merged, aes(x=bin.age,fill=factor(Performance.Tag)))+ geom\_bar()

#

## Income binning

#demo\_cb\_merged$bin.income <- demo\_cb\_merged$Income

#demo\_cb\_merged$bin.income <- ifelse((demo\_cb\_merged$Income>=0 & demo\_cb\_merged$Income<=15) , 'Low\_Income',demo\_cb\_merged$bin.income)

#demo\_cb\_merged$bin.income <- ifelse((demo\_cb\_merged$Income>15 & demo\_cb\_merged$Income<=30) , 'Medium\_Income',demo\_cb\_merged$bin.income)

#demo\_cb\_merged$bin.income <- ifelse((demo\_cb\_merged$Income>30) , 'High\_Income',demo\_cb\_merged$bin.income)

#table(demo\_cb\_merged$bin.income)

#

#ggplot(demo\_cb\_merged, aes(x=bin.income,fill=factor(Performance.Tag)))+ geom\_bar()

#

## No of months in current company binning

#demo\_cb\_merged$bin.nomcc <- demo\_cb\_merged$No.of.months.in.current.company

#demo\_cb\_merged$bin.nomcc <- ifelse((demo\_cb\_merged$No.of.months.in.current.company>=0 & demo\_cb\_merged$No.of.months.in.current.company<=24) , 'Short\_Tenure',demo\_cb\_merged$bin.nomcc)

#demo\_cb\_merged$bin.nomcc <- ifelse((demo\_cb\_merged$No.of.months.in.current.company>24 & demo\_cb\_merged$No.of.months.in.current.company<=72) , 'Medium\_Tenure',demo\_cb\_merged$bin.nomcc)

#demo\_cb\_merged$bin.nomcc <- ifelse((demo\_cb\_merged$No.of.months.in.current.company>72) , 'High\_Tenure',demo\_cb\_merged$bin.nomcc)

#table(demo\_cb\_merged$bin.nomcc)

#

#ggplot(demo\_cb\_merged, aes(x=bin.nomcc,fill=factor(Performance.Tag)))+ geom\_bar()

#

##No of months in current residence

#demo\_cb\_merged$bin.nomcr <- demo\_cb\_merged$No.of.months.in.current.residence

#demo\_cb\_merged$bin.nomcr <- ifelse((demo\_cb\_merged$No.of.months.in.current.residence>=0 & demo\_cb\_merged$No.of.months.in.current.residence<=24) , 'Short\_Tenure',demo\_cb\_merged$bin.nomcr)

#demo\_cb\_merged$bin.nomcr <- ifelse((demo\_cb\_merged$No.of.months.in.current.residence>24 & demo\_cb\_merged$No.of.months.in.current.residence<=72) , 'Medium\_Tenure',demo\_cb\_merged$bin.nomcr)

#demo\_cb\_merged$bin.nomcr <- ifelse((demo\_cb\_merged$No.of.months.in.current.residence>72) , 'High\_Tenure',demo\_cb\_merged$bin.nomcr)

#table(demo\_cb\_merged$bin.nomcr)

#

#ggplot(demo\_cb\_merged, aes(x=bin.nomcr,fill=factor(Performance.Tag)))+ geom\_bar()

#

## Outstanding Balance binning

#demo\_cb\_merged$bin.outstanding <- demo\_cb\_merged$Outstanding.Balance

#demo\_cb\_merged$bin.outstanding <- ifelse((demo\_cb\_merged$Outstanding.Balance>=0 & demo\_cb\_merged$Outstanding.Balance<=500000) , 'Low',demo\_cb\_merged$bin.outstanding)

#demo\_cb\_merged$bin.outstanding <- ifelse((demo\_cb\_merged$Outstanding.Balance>500000 & demo\_cb\_merged$Outstanding.Balance<=1500000) , 'Medium',demo\_cb\_merged$bin.outstanding)

#demo\_cb\_merged$bin.outstanding <- ifelse((demo\_cb\_merged$Outstanding.Balance>1500000 & demo\_cb\_merged$Outstanding.Balance<=3500000) , 'High',demo\_cb\_merged$bin.outstanding)

#demo\_cb\_merged$bin.outstanding <- ifelse((demo\_cb\_merged$Outstanding.Balance>3500000 ) , 'Very\_High',demo\_cb\_merged$bin.outstanding)

#table(demo\_cb\_merged$bin.outstanding)

#

#ggplot(demo\_cb\_merged, aes(x=bin.outstanding,fill=factor(Performance.Tag)))+ geom\_bar()

#

## Total no. of trades

##demo\_cb\_merged$bin.total\_trades <- demo\_cb\_merged$Total.No.of.Trades

##demo\_cb\_merged$bin.total\_trades <- ifelse((demo\_cb\_merged$Total.No.of.Trades>=0 & demo\_cb\_merged$Total.No.of.Trades<=5) , 'Low',demo\_cb\_merged$bin.total\_trades)

##demo\_cb\_merged$bin.total\_trades <- ifelse((demo\_cb\_merged$Total.No.of.Trades>5 & demo\_cb\_merged$Total.No.of.Trades<=10) , 'Medium',demo\_cb\_merged$bin.total\_trades)

##demo\_cb\_merged$bin.total\_trades <- ifelse((demo\_cb\_merged$Total.No.of.Trades>10) , 'High',demo\_cb\_merged$bin.total\_trades)

#

## Average CC Utilization

#demo\_cb\_merged$bin.avg\_cc <- demo\_cb\_merged$Avgas.CC.Utilization.in.last.12.months

#demo\_cb\_merged$bin.avg\_cc <- ifelse((demo\_cb\_merged$Avgas.CC.Utilization.in.last.12.months>=0 & demo\_cb\_merged$Avgas.CC.Utilization.in.last.12.months<=20) , 'Low',demo\_cb\_merged$bin.avg\_cc)

#demo\_cb\_merged$bin.avg\_cc <- ifelse((demo\_cb\_merged$Avgas.CC.Utilization.in.last.12.months>20 & demo\_cb\_merged$Avgas.CC.Utilization.in.last.12.months<=50) , 'Medium',demo\_cb\_merged$bin.avg\_cc)

#demo\_cb\_merged$bin.avg\_cc <- ifelse((demo\_cb\_merged$Avgas.CC.Utilization.in.last.12.months>50 & demo\_cb\_merged$Avgas.CC.Utilization.in.last.12.months<=80) , 'High',demo\_cb\_merged$bin.avg\_cc)

#demo\_cb\_merged$bin.avg\_cc <- ifelse((demo\_cb\_merged$Avgas.CC.Utilization.in.last.12.months>80 ) , 'Very\_High',demo\_cb\_merged$bin.avg\_cc)

#table(demo\_cb\_merged$bin.avg\_cc)

#

#ggplot(demo\_cb\_merged, aes(x=bin.avg\_cc,fill=factor(Performance.Tag)))+ geom\_bar()

#to\_remove <- c("Age","Income","No.of.months.in.current.company","No.of.months.in.current.residence",

# "Outstanding.Balance","Avgas.CC.Utilization.in.last.12.months")

#demo\_cb\_merged <- demo\_cb\_merged[,!(names(demo\_cb\_merged) %in% to\_remove)]

# creating final set of categorical and continuous variables

#cat <- c("Performance.Tag","Gender","Marital.Status..at.the.time.of.application.","Presence.of.open.home.loan","Presence.of.open.auto.loan","bin.age","bin.income","bin.nomcc","bin.nomcr","bin.outstanding","bin.avg\_cc","Education","Profession","Type.of.residence","No.of.dependents")

#cont <- c("No.of.times.90.DPD.or.worse.in.last.6.months","No.of.times.60.DPD.or.worse.in.last.6.months","No.of.times.30.DPD.or.worse.in.last.6.months","No.of.times.90.DPD.or.worse.in.last.12.months","No.of.times.60.DPD.or.worse.in.last.12.months","No.of.times.30.DPD.or.worse.in.last.12.months","No.of.trades.opened.in.last.6.months","No.of.trades.opened.in.last.12.months","No.of.PL.trades.opened.in.last.6.months","No.of.PL.trades.opened.in.last.12.months","No.of.Inquiries.in.last.6.months..excluding.home...auto.loans.","No.of.Inquiries.in.last.12.months..excluding.home...auto.loans.","Total.No.of.Trades")

cat <- c("Performance.Tag","Gender","Marital.Status..at.the.time.of.application.","No.of.dependents","Education","Profession","Type.of.residence","Presence.of.open.home.loan","Presence.of.open.auto.loan")

cont <- c("Age","Income","No.of.months.in.current.residence","No.of.months.in.current.company","No.of.times.90.DPD.or.worse.in.last.6.months","No.of.times.60.DPD.or.worse.in.last.6.months","No.of.times.30.DPD.or.worse.in.last.6.months","No.of.times.90.DPD.or.worse.in.last.12.months","No.of.times.60.DPD.or.worse.in.last.12.months","No.of.times.30.DPD.or.worse.in.last.12.months","Avgas.CC.Utilization.in.last.12.months","No.of.trades.opened.in.last.6.months","No.of.trades.opened.in.last.12.months","No.of.PL.trades.opened.in.last.6.months","No.of.PL.trades.opened.in.last.12.months","No.of.Inquiries.in.last.6.months..excluding.home...auto.loans.","No.of.Inquiries.in.last.12.months..excluding.home...auto.loans.","Outstanding.Balance","Total.No.of.Trades")

final\_cat <- demo\_cb\_merged[,cat]

final\_cont <- demo\_cb\_merged[,cont]

# Normalising continuous features

#final\_cont <- data.frame(sapply(final\_cont, function(x) scale(x)))

#str(final\_cont)

# converting categorical attributes to factor

final\_cat<- data.frame(sapply(final\_cat, function(x) factor(x)))

str(final\_cat)

summary(final\_cat$Gender)

# F M

# 16234 52483

summary(final\_cat$Marital.Status..at.the.time.of.application.)

#Married Single

# 58548 10169

# creating dummy variables for factor attributes

dummies<- data.frame(sapply(final\_cat,

function(x) data.frame(model.matrix(~x-1,data =final\_cat))[,-1]))

sum(dummies$Gender)

# suggest Male == 1 and Female == 0

sum(dummies$Marital.Status..at.the.time.of.application.)

# suggest Single == 1 and Married == 0

final\_data <- cbind(dummies,final\_cont)

########################################################################

######################### splitting the data between train and test #############################

set.seed(100)

indices = sample.split(final\_data$Performance.Tag, SplitRatio = 0.7)

train = final\_data[indices,]

test = final\_data[!(indices),]

#####################################################################################

#Model Building

#####################################################################################

############################# Logistic Regression####################################

#model containing all the variables

model\_1 <- glm(Performance.Tag~.,data=train,family = "binomial")

summary(model\_1)

# used STEPAIC to find the best model

model\_2 <- stepAIC(model\_1,direction = "both")

summary(model\_2)

sort(vif(model\_2))

# removing 60 dpd in last 6 months

model\_3 <- glm(formula = Performance.Tag ~ No.of.dependents.x2 + Profession.xSE +

Income + No.of.months.in.current.residence + No.of.months.in.current.company +

No.of.times.30.DPD.or.worse.in.last.6.months +

No.of.times.90.DPD.or.worse.in.last.12.months + Avgas.CC.Utilization.in.last.12.months +

No.of.PL.trades.opened.in.last.12.months + No.of.Inquiries.in.last.12.months..excluding.home...auto.loans. +

Total.No.of.Trades, family = "binomial", data = train)

summary(model\_3)

sort(vif(model\_3))

# removing no of times 90 dpd based on the p-value

model\_4 <- glm(formula = Performance.Tag ~ No.of.dependents.x2 + Profession.xSE +

Income + No.of.months.in.current.residence + No.of.months.in.current.company +

No.of.times.30.DPD.or.worse.in.last.6.months +

Avgas.CC.Utilization.in.last.12.months +

No.of.PL.trades.opened.in.last.12.months + No.of.Inquiries.in.last.12.months..excluding.home...auto.loans. +

Total.No.of.Trades, family = "binomial", data = train)

summary(model\_4)

sort(vif(model\_4))

# removing income based on the p-value

model\_5 <- glm(formula = Performance.Tag ~ No.of.dependents.x2 + Profession.xSE +

No.of.months.in.current.residence + No.of.months.in.current.company +

No.of.times.30.DPD.or.worse.in.last.6.months +

Avgas.CC.Utilization.in.last.12.months +

No.of.PL.trades.opened.in.last.12.months + No.of.Inquiries.in.last.12.months..excluding.home...auto.loans. +

Total.No.of.Trades, family = "binomial", data = train)

summary(model\_5)

sort(vif(model\_5))

# removing no of months in current residence based on the p-value

model\_6 <- glm(formula = Performance.Tag ~ No.of.dependents.x2 + Profession.xSE +

No.of.months.in.current.company +

No.of.times.30.DPD.or.worse.in.last.6.months +

Avgas.CC.Utilization.in.last.12.months +

No.of.PL.trades.opened.in.last.12.months + No.of.Inquiries.in.last.12.months..excluding.home...auto.loans. +

Total.No.of.Trades, family = "binomial", data = train)

summary(model\_6)

sort(vif(model\_6))

# removing profession based on the p-value

model\_7 <- glm(formula = Performance.Tag ~ No.of.dependents.x2 +

No.of.months.in.current.company +

No.of.times.30.DPD.or.worse.in.last.6.months +

Avgas.CC.Utilization.in.last.12.months +

No.of.PL.trades.opened.in.last.12.months + No.of.Inquiries.in.last.12.months..excluding.home...auto.loans. +

Total.No.of.Trades, family = "binomial", data = train)

summary(model\_7)

sort(vif(model\_7))

#predicted probabilities of default for test data

test\_pred = predict(model\_7, type = "response",

newdata = test[,-1])

# Let's see the summary

summary(test\_pred)

test$prob <- test\_pred

View(test)

# Let's use the probability cutoff of 50%.

test\_pred\_def <- factor(ifelse(test\_pred >= 0.50, "Yes", "No"))

test\_actual\_def <- factor(ifelse(test$Performance.Tag==1,"Yes","No"))

table(test\_actual\_def,test\_pred\_def)

test\_conf <- confusionMatrix(test\_pred\_def, test\_actual\_def, positive = "Yes")

test\_conf

#######################################################################

perform\_fn <- function(cutoff)

{

predicted\_def <- factor(ifelse(test\_pred >= cutoff, "Yes", "No"))

conf <- confusionMatrix(predicted\_def, test\_actual\_def, positive = "Yes")

acc <- conf$overall[1]

sens <- conf$byClass[1]

spec <- conf$byClass[2]

out <- t(as.matrix(c(sens, spec, acc)))

colnames(out) <- c("sensitivity", "specificity", "accuracy")

return(out)

}

# Creating cutoff values from 0.003575 to 0.812100 for plotting and initiallizing a matrix of 100 X 3.

# Summary of test probability

summary(test\_pred)

s = seq(.01,.80,length=100)

OUT = matrix(0,100,3)

for(i in 1:100)

{

OUT[i,] = perform\_fn(s[i])

}

plot(s, OUT[,1],xlab="Cutoff",ylab="Value",cex.lab=1.5,cex.axis=1.5,ylim=c(0,1),type="l",lwd=2,axes=FALSE,col=2)

axis(1,seq(0,1,length=5),seq(0,1,length=5),cex.lab=1.5)

axis(2,seq(0,1,length=5),seq(0,1,length=5),cex.lab=1.5)

lines(s,OUT[,2],col="darkgreen",lwd=2)

lines(s,OUT[,3],col=4,lwd=2)

box()

legend(0,.50,col=c(2,"darkgreen",4,"darkred"),lwd=c(2,2,2,2),c("Sensitivity","Specificity","Accuracy"))

cutoff <- s[which(abs(OUT[,1]-OUT[,2])<0.01)]

# Let's choose a cutoff value of 0.05 for final model

test\_cutoff\_def <- factor(ifelse(test\_pred >=0.05, "Yes", "No"))

conf\_final <- confusionMatrix(test\_cutoff\_def, test\_actual\_def, positive = "Yes")

acc <- conf\_final$overall[1]

sens <- conf\_final$byClass[1]

spec <- conf\_final$byClass[2]

conf\_final

# Accuracy = 68%

# Sensitivity = 53.2%

# Specificity = 68.9%

############################################################################################

############# Model Evaluation##################

### KS -statistic - Test Data ######

test\_cutoff\_def <- ifelse(test\_cutoff\_def=="Yes",1,0)

test\_actual\_def <- ifelse(test\_actual\_def=="Yes",1,0)

library(ROCR)

#on testing data

pred\_object\_test<- prediction(test\_cutoff\_def, test\_actual\_def)

performance\_measures\_test<- performance(pred\_object\_test, "tpr", "fpr")

ks\_table\_test <- attr(performance\_measures\_test, "y.values")[[1]] -

(attr(performance\_measures\_test, "x.values")[[1]])

max(ks\_table\_test)

# 0.22

####################################################################

# Lift & Gain Chart

# plotting the lift chart

# Loading dplyr package

require(dplyr)

library(dplyr)

lift <- function(labels , predicted\_prob,groups=10) {

if(is.factor(labels)) labels <- as.integer(as.character(labels ))

if(is.factor(predicted\_prob)) predicted\_prob <- as.integer(as.character(predicted\_prob))

helper = data.frame(cbind(labels , predicted\_prob))

helper[,"bucket"] = ntile(-helper[,"predicted\_prob"], groups)

gaintable = helper %>% group\_by(bucket) %>%

summarise\_at(vars(labels ), funs(total = n(),

totalresp=sum(., na.rm = TRUE))) %>%

mutate(Cumresp = cumsum(totalresp),

Gain=Cumresp/sum(totalresp)\*100,

Cumlift=Gain/(bucket\*(100/groups)))

return(gaintable)

}

Churn\_decile = lift(test\_actual\_def, test\_pred, groups = 10)

#####################################################################################################################

# Since there are just 2897/68717 records having performance tag value as 1, this will create an imbalance

# in the model, so we will use Smote : Synthetic Minority Oversampling Technique To Handle Class Imbalancy In Binary Classification

#####################################################################################

############################### Random Forrest ######################################

#####################################################################################

set.seed(100)

trainindices= sample(1:nrow(final\_data), 0.7\*nrow(final\_data))

train = final\_data[trainindices,]

test = final\_data[-trainindices,]

rf\_fit <- randomForest(Performance.Tag~.,train,ntree=500,importance=T)

plot(rf\_fit)

varImpPlot(rf\_fit,sort = T,main="Variable Importance",n.var=5)

var.imp <- data.frame(importance(rf\_fit,type=2))

var.imp$Variables <- row.names(var.imp)

var.imp[order(var.imp$MeanDecreaseGini,decreasing = T),]

test$predicted.prob <- predict(rf\_fit ,test[,-1], type = "prob")

s = seq(.5,.99,length=100)

OUT = matrix(0,100,3)

for(i in 1:100){

OUT[i,] = perform\_fn1(s[i])

}

#Plot to choose best cutoff

plot(s, OUT[,1],xlab="Cutoff",ylab="Value",cex.lab=1.5,cex.axis=1.5,ylim=c(0,1),type="l",lwd=2,axes=FALSE,col=2)

axis(1,seq(0,1,length=5),seq(0,1,length=5),cex.lab=1.5)

axis(2,seq(0,1,length=5),seq(0,1,length=5),cex.lab=1.5)

lines(s,OUT[,2],col="darkgreen",lwd=2)

lines(s,OUT[,3],col=4,lwd=2)

box()

legend(0.6,.80,col=c(2,"darkgreen",4,"darkred"),lwd=c(2,2,2,2),c("Sensitivity","Specificity","Accuracy"))

best\_cutoff <- s[which(abs(OUT[,1]-OUT[,2])==min(abs(OUT[,1]-OUT[,2])))]

best\_cutoff

test$Performance.Tag<- factor(ifelse(test$Performance.Tag==1,"Yes","No"))

test$predicted.response\_1<- factor(ifelse(test$predicted.prob[,1] >= best\_cutoff, "Yes", "No"))

confusionmartix\_final\_rf<-confusionMatrix(test$predicted.response\_1, test$Performance.Tag, positive = "Yes")