8\_1.R

Lithin Reddy Marla

Thu Sep 08 17:29:40 2016

rm(list=ls(all=T))  
  
set.seed(999)  
  
setwd("C:/Users/dell/Dropbox/compitition/nasscom/BFSI")  
library(data.table)

library(mice)

library(dummies)

library(xgboost)

library(Matrix)  
library(car)

library(leaps)

library(xlsx)

library(randomForest)

require('vcd')

source("funplot.R")

#'funplot.R' has few custom defined functioned for visualization and diagnosis

library(ggplot2)

comp<-function(predicted,actual){

qplot(x=predicted,y=actual)+

stat\_smooth() +

geom\_abline(intercept = 0, slope = 1, color = "red")+

labs(x = "predicted", y = "actual")

}

res\_plot<-function(fit){

df.fit<-fortify(fit)

ggplot(data = df.fit, aes(x = .fitted, y = .resid)) +

geom\_point() +

stat\_smooth() +

labs(x = "fitted", y = "residual")+

geom\_abline(intercept = 0, slope = 0, color = "red")

# identify(x = .fitted, y = .resid)

}

res\_plot1<-function(fit,x){

df.fit<-fortify(fit)

ggplot(data = df.fit, aes(x , y = .resid)) +

geom\_point() +

stat\_smooth() +

geom\_abline(intercept = 0, slope = 0, color = "red")

labs(x = deparse(substitute(x)), y = "residual")

# identify(x,.resid)

}

res\_plot2<-function(predicted,actual){

qplot(x = predicted, y =actual-predicted)+

stat\_smooth()+

geom\_abline(intercept = 0, slope = 0, color = "red") +

labs(x = "fitted", y = "residual")

# identify(predicted,actual)

}

test<-function(fit){

plot(residuals(fit),data$index)#autocorrelation

durbinWatsonTest(fit)#autocorrelation

car::vif(fit)#collinearity

ncvTest(fit)#heteroscedasticity

residualPlots(fit,ask=T,plot=F)#non-linearity

}

dia<-function(fit){

library(car)

layout(1)

plot(fit)#diagnosis

influencePlot(fit)#outliers

residualPlots(fit,ask=T,layout=c(2,2),

type="rstandard")#non-linearity

}

RMS<-function(x,y,p){

rms<-sqrt(( sum( (x - y)^2 , na.rm = TRUE )) / (length(x)-p-1))

avg\_error<-sum(x - y)/(length(x)-p)

return(list(rms,avg\_error))

}

#1.Data Preparation

tr<-fread("BFSI Stage 1 Train data.csv",stringsAsFactors=F)  
te<-fread("BFSI Stage 1 Test data.csv",stringsAsFactors=F)  
  
sol<-fread("BFSI - Solution submission template.csv",stringsAsFactors=T)  
te<-te[,c("Behavioural\_Score","Application\_Score"):=0]  
  
c<-list(tr, te)  
a<- rbindlist(c)  
  
a$`Pre-Approved Auto Limit`<-ordered(a$`Pre-Approved Auto Limit`,levels=  
 c("MISSING",">480,000" ,"<360,000", "<480,000",  
 "<240,000" ,"<120,000" ,"ZERO"),  
 labels=c(NA,5,3,4,2,1,0))  
a$`Pre-Approved Mortgage Limit`<-ordered(a$`Pre-Approved Mortgage Limit`,levels=  
 c(">=5,000,000","<2,500,000" , "<5,000,000",   
 "<1,000,000" ,"MISSING", "ZERO" ),labels=c(4,3,2,1,NA,0))  
a$`Pre-Approved Personal Limit`<-ordered(a$`Pre-Approved Personal Limit`,  
 levels=c("MISSING ","ZERO","<1,500,000","<1,000,000","<500,000"),  
 labels=c(NA,0,3,2,1))  
a$Total\_Bank\_Products<-ordered(a$Total\_Bank\_Products,levels=c(">=6","5","2","1","3","0","4"),labels=c(6,5,2,1,3,0,4))  
a$Active\_Bank\_Products<-ordered(a$Active\_Bank\_Products,levels=c(">=5","2","1","3","0","4"),labels=c(5,2,1,3,0,4))  
a$Tenure\_with\_Bank\_Group<-ordered(a$Tenure\_with\_Bank\_Group,levels=c("<=5 YRS","<=10 YRS",">10 YRS","<=2 YRS", "<=1 YRS"),labels=c(2,4,5,1,0))  
a$Education<-ordered(a$Education,levels=c("University","Graduate and Higher","Intermediate","High School","Unknown","No Education","Primary School" ),labels=c(5,4,3,2,NA,0,1))  
  
colSums(is.na(a))

## Identifier Age   
## 0 0   
## Education Indutry\_Groups   
## 0 0   
## Industry\_Domain Marital\_Status   
## 0 0   
## Occupation Customer\_Segment   
## 0 0   
## Gender Metropolitan\_City   
## 0 0   
## Residence Insurance\_Product\_type   
## 0 0   
## Insurance\_Acquisition\_Channel Active\_Bank\_Products   
## 0 0   
## Total\_Bank\_Products Commercial\_Loan   
## 0 0   
## Consumer\_Auto\_Loan Mortgage\_Loan   
## 0 0   
## Personal\_Loan Credit\_Card   
## 0 0   
## Deposit Pre-Approved Auto Limit   
## 0 0   
## Pre-Approved Mortgage Limit Pre-Approved Personal Limit   
## 0 6084   
## Tenure\_with\_Bank\_Group Tenure\_of\_Insurance   
## 0 0   
## Salary\_Amount Credit\_Limit   
## 976 1681   
## Total\_Asset\_Under\_Mngmnt Avg\_Monthly\_Balance   
## 0 0   
## Application\_Score Behavioural\_Score   
## 0 0

a[,.(.N),.(`Pre-Approved Auto Limit`)]

## Pre-Approved Auto Limit N  
## 1: NA 6084  
## 2: 5 1944  
## 3: 3 328  
## 4: 4 354  
## 5: 2 344  
## 6: 1 151  
## 7: 0 26

a[,.(.N),.(`Pre-Approved Mortgage Limit`)]

## Pre-Approved Mortgage Limit N  
## 1: NA 6084  
## 2: 4 1434  
## 3: 3 612  
## 4: 2 505  
## 5: 1 335  
## 6: 0 261

a[,.(.N),.(`Pre-Approved Personal Limit`)]

## Pre-Approved Personal Limit N  
## 1: NA 6084  
## 2: 0 2068  
## 3: 3 782  
## 4: 2 250  
## 5: 1 47

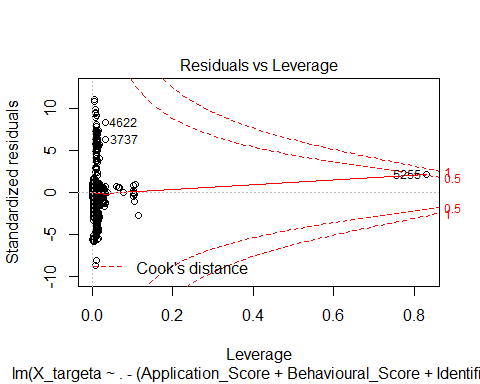
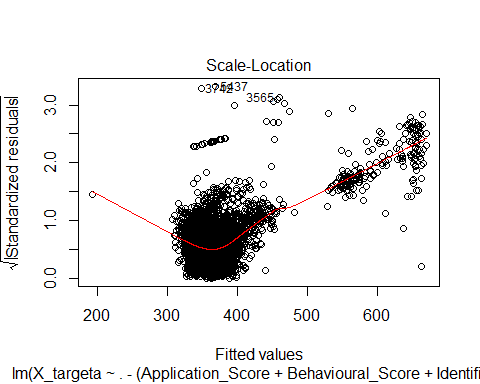
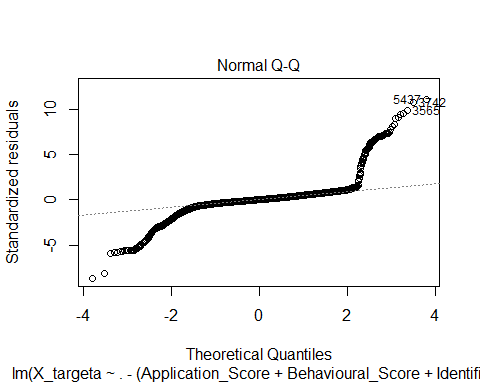
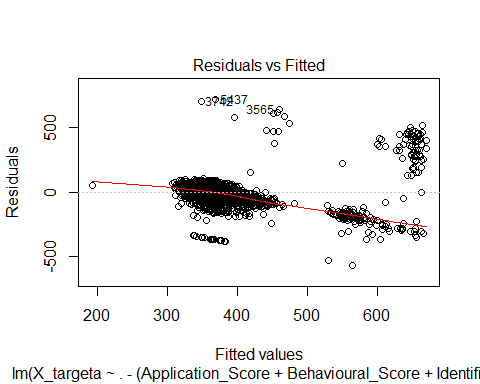
a\_train<-a[,":="(`Pre-Approved Mortgage Limit`=NULL,  
 `Pre-Approved Auto Limit`=NULL,  
 `Pre-Approved Personal Limit`=NULL)]  
  
a\_train <- a[,":="(Identifier =as.numeric(Identifier),  
 Age=as.numeric(Age),  
 Education=(Education),  
 Indutry\_Groups=as.numeric(as.factor(Indutry\_Groups)),  
 Industry\_Domain=as.numeric(as.factor(Industry\_Domain)),  
 Marital\_Status=as.numeric(as.factor(Marital\_Status)),  
 Occupation=as.numeric(as.factor(Occupation)),  
 Customer\_Segment=as.numeric(as.factor(Customer\_Segment)),  
 Gender=ifelse(Gender=="Male",1,0),  
 Metropolitan\_City=as.numeric(as.factor(Metropolitan\_City)),  
 Residence=as.numeric(as.factor(Residence)),  
 Insurance\_Product\_type=as.numeric(as.factor(Insurance\_Product\_type)),  
 Insurance\_Acquisition\_Channel=as.numeric(as.factor(Insurance\_Acquisition\_Channel)),  
 Active\_Bank\_Products=(Active\_Bank\_Products),  
 Total\_Bank\_Products=as.numeric(Total\_Bank\_Products),  
 Commercial\_Loan=as.numeric(as.factor(Commercial\_Loan)),  
 Consumer\_Auto\_Loan=as.numeric(as.factor(Consumer\_Auto\_Loan)),  
 Mortgage\_Loan=as.numeric(as.factor(Mortgage\_Loan)),  
 Personal\_Loan=as.numeric(as.factor(Personal\_Loan)),  
 Credit\_Card=as.numeric(as.factor(Credit\_Card)),  
 Deposit=as.numeric(as.factor(Deposit)),  
 Tenure\_with\_Bank\_Group=(Tenure\_with\_Bank\_Group),  
 Tenure\_of\_Insurance=as.numeric(Tenure\_of\_Insurance),  
 Salary\_Amount=as.numeric(Salary\_Amount),  
 Credit\_Limit=as.numeric(Credit\_Limit),  
 Total\_Asset\_Under\_Mngmnt=as.numeric(as.factor(Total\_Asset\_Under\_Mngmnt)),  
 Avg\_Monthly\_Balance=as.numeric(Avg\_Monthly\_Balance),  
 Application\_Score=as.numeric(Application\_Score),  
 Behavioural\_Score=as.numeric(Behavioural\_Score)  
)]  
  
  
Dat1 <- subset(a\_train, select=c(Salary\_Amount,Credit\_Limit,Avg\_Monthly\_Balance,Tenure\_of\_Insurance,  
 Active\_Bank\_Products,Total\_Bank\_Products,Age,Total\_Asset\_Under\_Mngmnt))   
imp <- mice(Dat1, m=3, maxit=10,method="pmm", seed=999)

##   
## iter imp variable  
## 1 1 Salary\_Amount Credit\_Limit  
## 1 2 Salary\_Amount Credit\_Limit  
## 1 3 Salary\_Amount Credit\_Limit  
## 2 1 Salary\_Amount Credit\_Limit.  
.  
.  
.  
  
## 9 2 Salary\_Amount Credit\_Limit  
## 9 3 Salary\_Amount Credit\_Limit  
## 10 1 Salary\_Amount Credit\_Limit  
## 10 2 Salary\_Amount Credit\_Limit  
## 10 3 Salary\_Amount Credit\_Limit

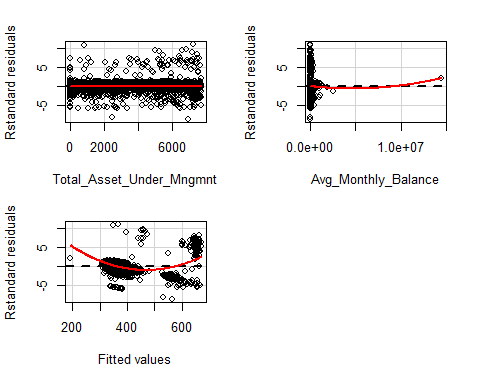
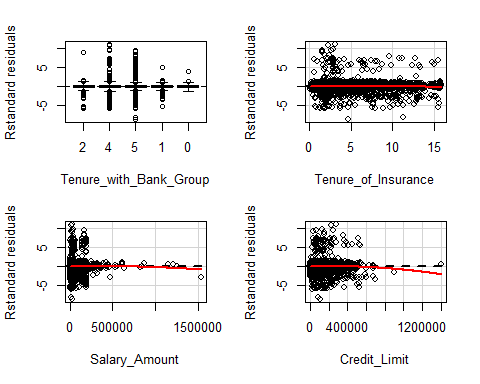
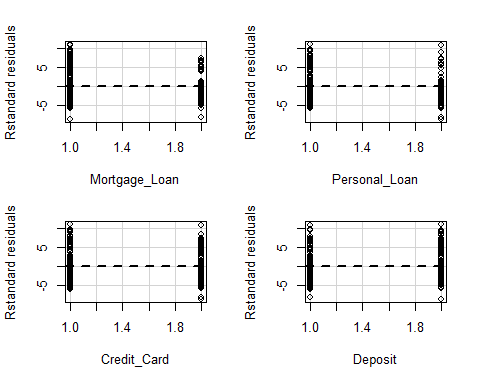
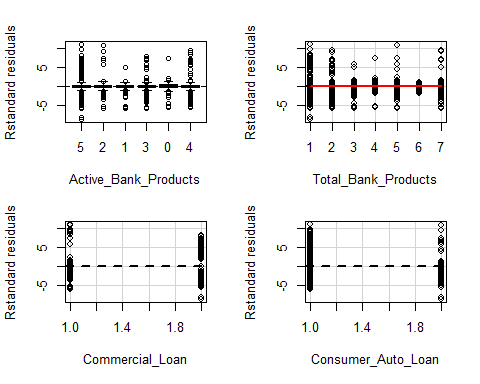
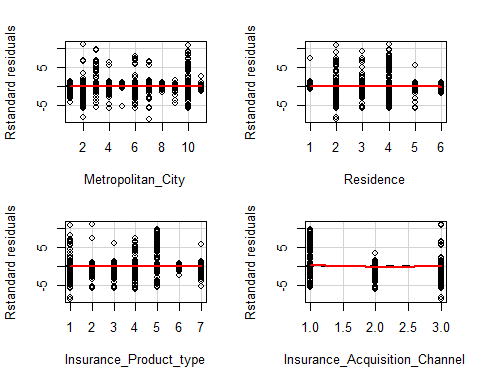
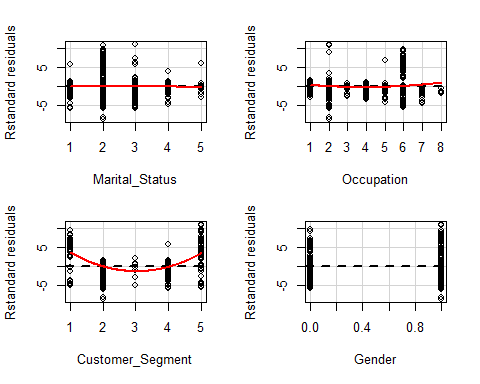
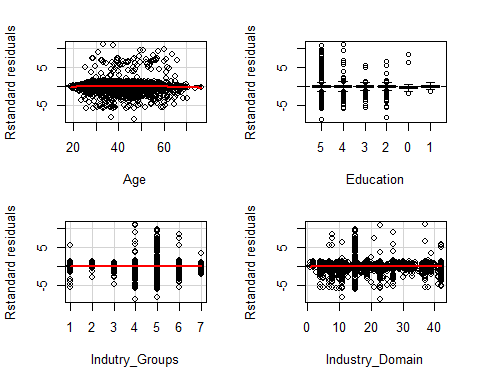
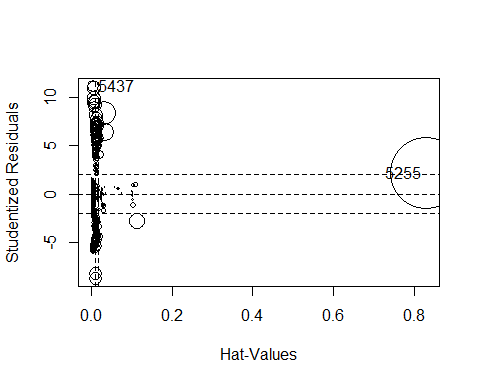
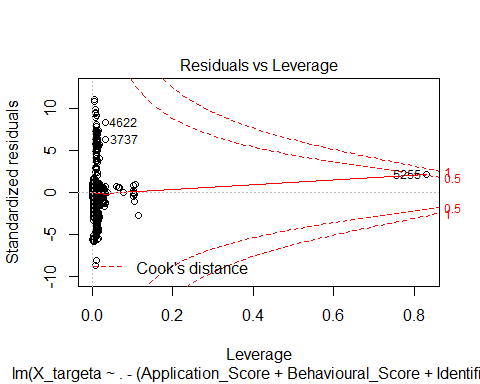
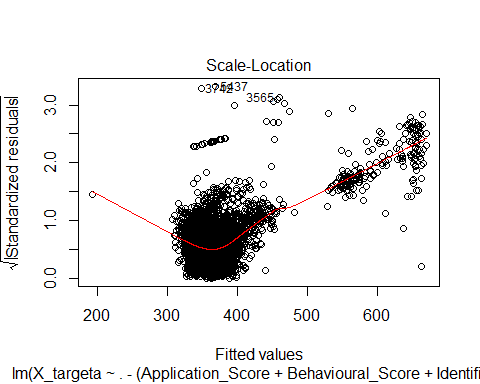
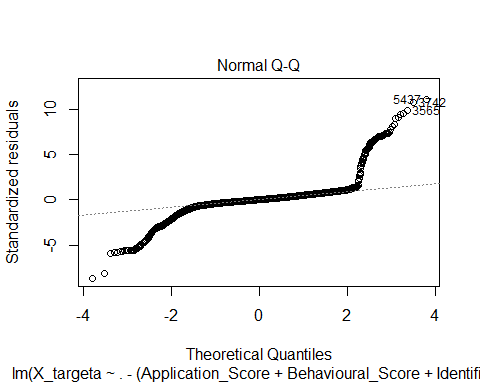
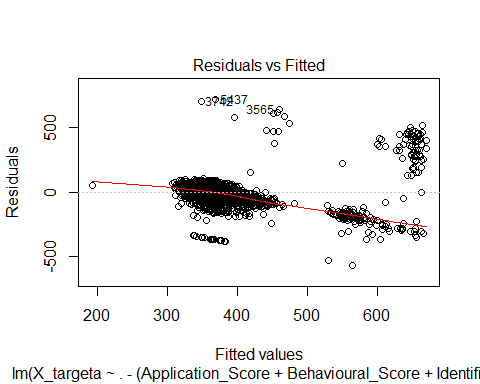
Data.imp<- complete(imp,3)  
  
X\_train <- a\_train[,":="(Salary\_Amount=Data.imp$Salary\_Amount,  
 Credit\_Limit=Data.imp$Credit\_Limit)][1:6924,]  
X\_test <- a\_train[,":="(Salary\_Amount=Data.imp$Salary\_Amount,  
 Credit\_Limit=Data.imp$Credit\_Limit)][6925:9231,]  
X\_targeta <- as.numeric(X\_train$Application\_Score)  
X\_targetb <- as.numeric(X\_train$Behavioural\_Score)  
  
ma<-lm(X\_targeta~.-(Application\_Score+Behavioural\_Score+Identifier),data = X\_train)  
summary(ma)

##   
## Call:  
## lm(formula = X\_targeta ~ . - (Application\_Score + Behavioural\_Score +   
## Identifier), data = X\_train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -564.32 -17.81 0.43 20.70 719.08   
## Residual standard error: 65.34 on 6885 degrees of freedom  
## Multiple R-squared: 0.3097, Adjusted R-squared: 0.3059   
## F-statistic: 81.3 on 38 and 6885 DF, p-value: < 2.2e-16

plot(ma)



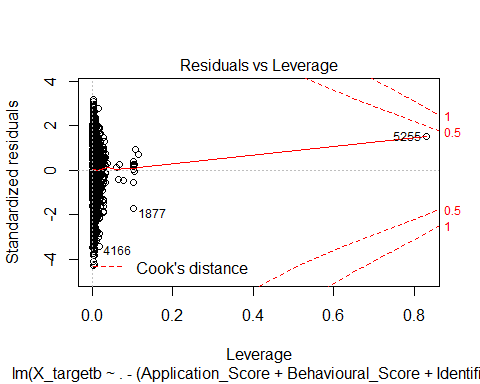
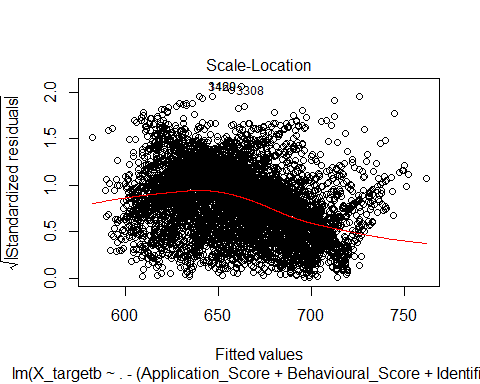
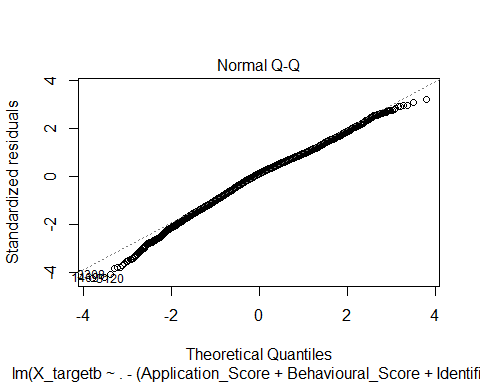
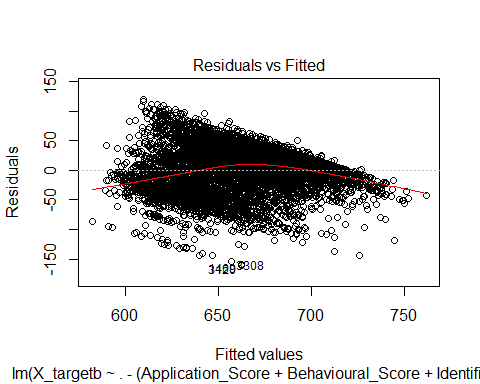
dia(ma)



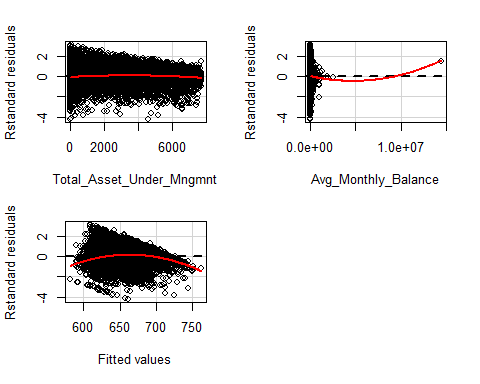
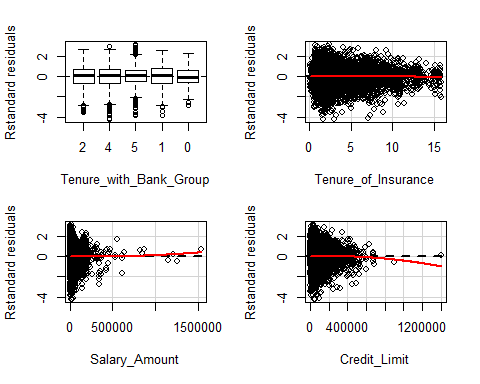
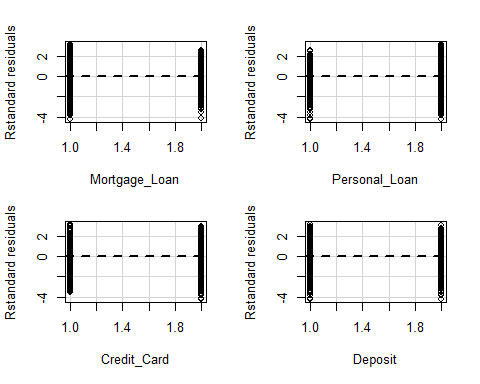
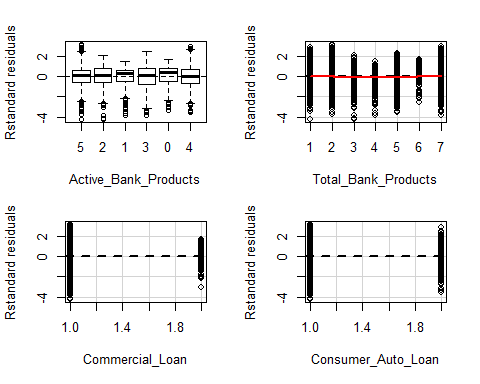
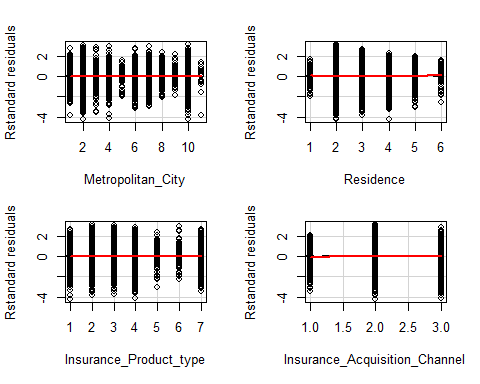
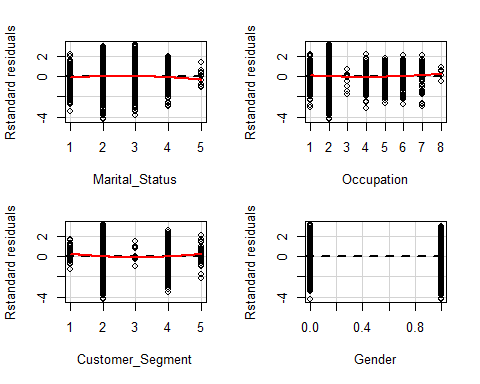
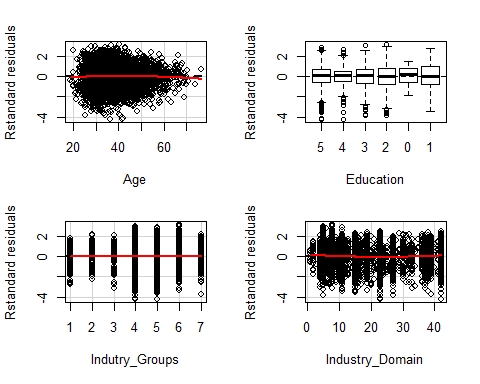
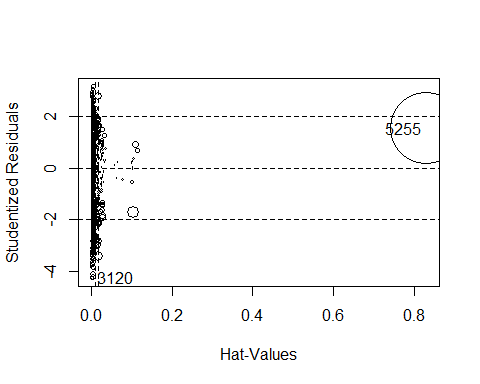
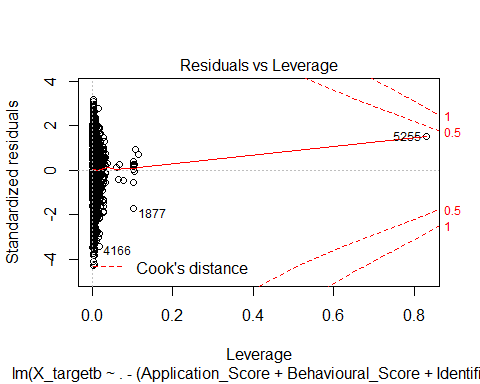
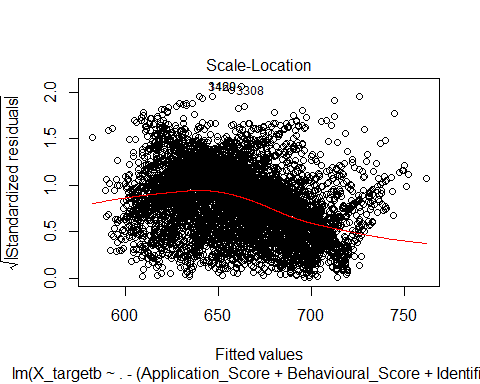
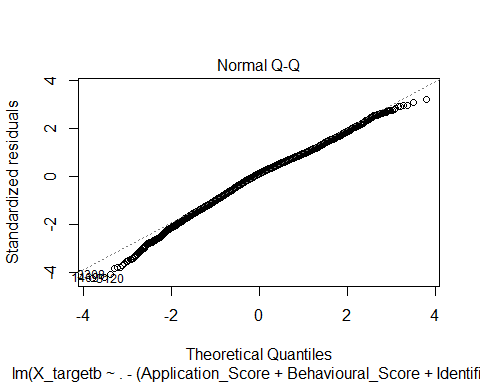
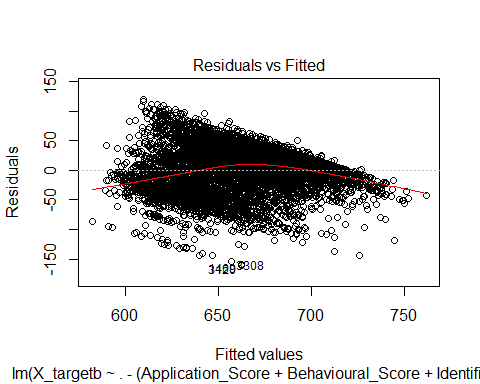
mb<-lm(X\_targetb~.-(Application\_Score+Behavioural\_Score+Identifier),data = X\_train)  
summary(mb)

##   
## Call:  
## lm(formula = X\_targetb ~ . - (Application\_Score + Behavioural\_Score +   
## Identifier), data = X\_train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -160.18 -24.21 3.80 25.32 119.57   
## Residual standard error: 37.61 on 6885 degrees of freedom  
## Multiple R-squared: 0.3922, Adjusted R-squared: 0.3889   
## F-statistic: 116.9 on 38 and 6885 DF, p-value: < 2.2e-16

plot(mb)



dia(mb)

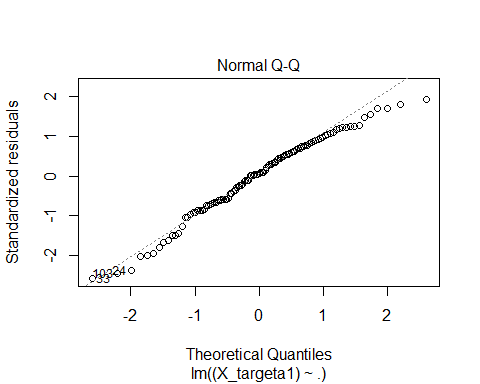
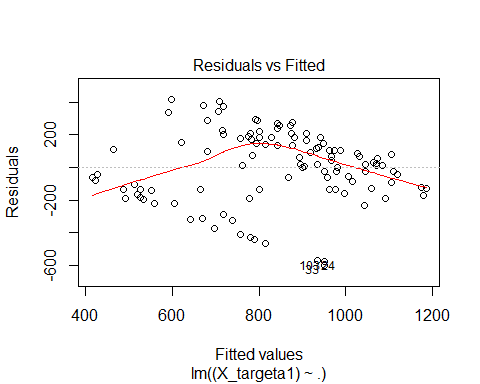


X\_train1<-X\_train[X\_train$Customer\_Segment %in% c(1,5)]  
X\_train2<-X\_train[X\_train$Customer\_Segment %in% c(2,3,4)]  
X\_targeta1 <- as.numeric(X\_train1$Application\_Score)  
X\_targetb1 <- as.numeric(X\_train1$Behavioural\_Score)  
X\_targeta2 <- as.numeric(X\_train2$Application\_Score)  
X\_targetb2 <- as.numeric(X\_train2$Behavioural\_Score)  
  
I<-X\_train$Identifier  
I1<-X\_train1$Identifier  
I2<-X\_train2$Identifier  
  
X\_train<-X\_train[,":="(Application\_Score=NULL,Behavioural\_Score=NULL,Identifier=NULL)]  
X\_train1<-X\_train1[,":="(Application\_Score=NULL,Behavioural\_Score=NULL,Identifier=NULL)]  
X\_train2<-X\_train2[,":="(Application\_Score=NULL,Behavioural\_Score=NULL,Identifier=NULL)]  
  
X\_test1<-X\_test[X\_test$Customer\_Segment %in% c(1,5)]  
X\_test2<-X\_test[X\_test$Customer\_Segment %in% c(2,3,4)]  
  
X\_test1 <- X\_test1[, .SDcols=names(X\_train)]  
X\_test2 <- X\_test2[, .SDcols=names(X\_train)]  
  
ma1<-lm((X\_targeta1)~.,data = X\_train1)  
summary(ma1)

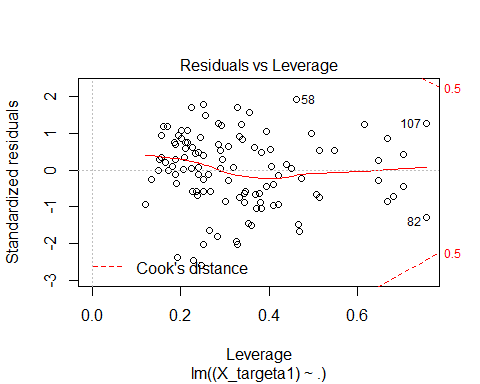
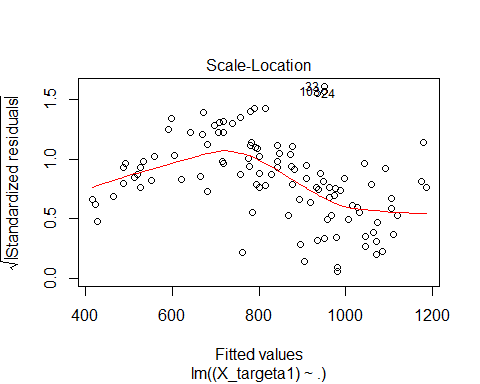
##   
## Call:  
## lm(formula = (X\_targeta1) ~ ., data = X\_train1)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -600.23 -135.24 13.37 165.27 416.63   
## Residual standard error: 267.5 on 73 degrees of freedom  
## Multiple R-squared: 0.4271, Adjusted R-squared: 0.1446   
## F-statistic: 1.512 on 36 and 73 DF, p-value: 0.06806

plot(ma1)

## Warning: not plotting observations with leverage one:  
## 59

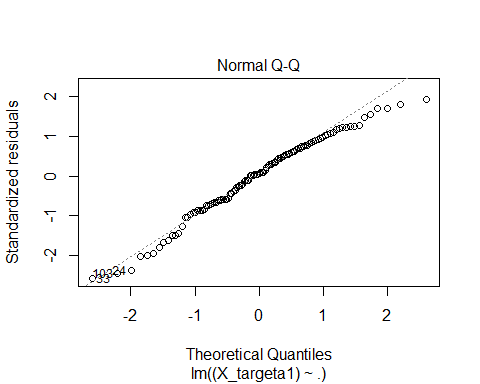
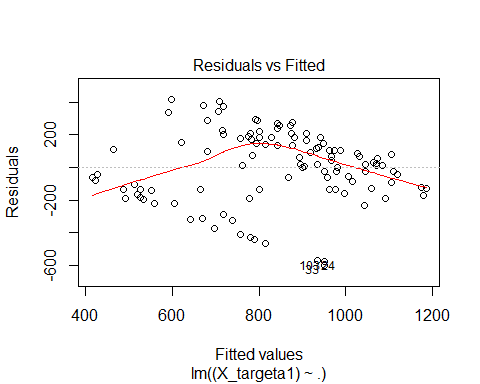


## Warning: not plotting observations with leverage one:  
## 59

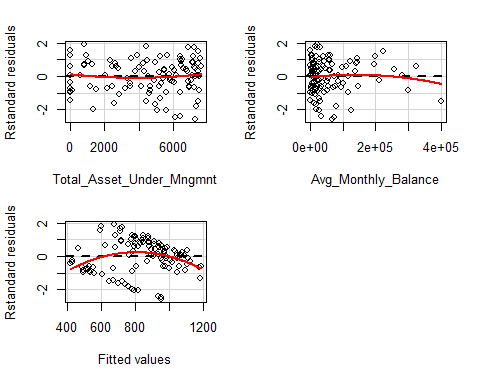
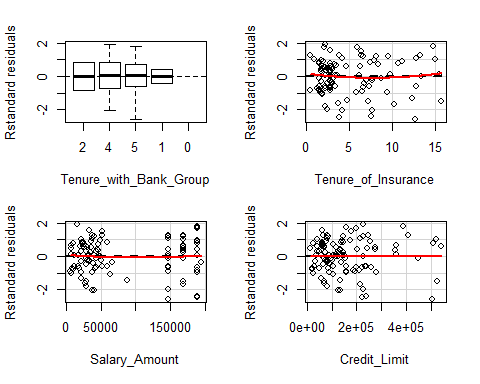
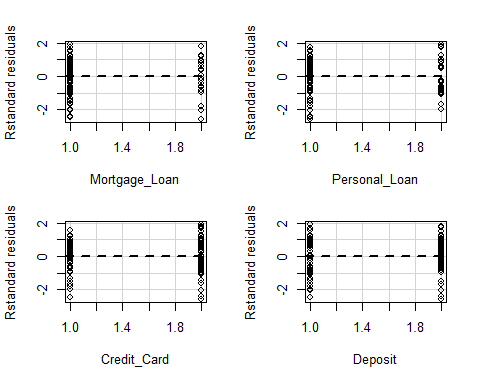
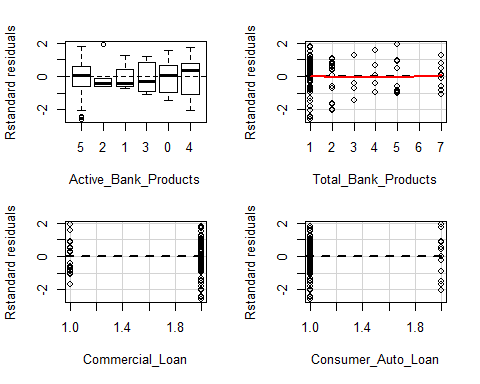
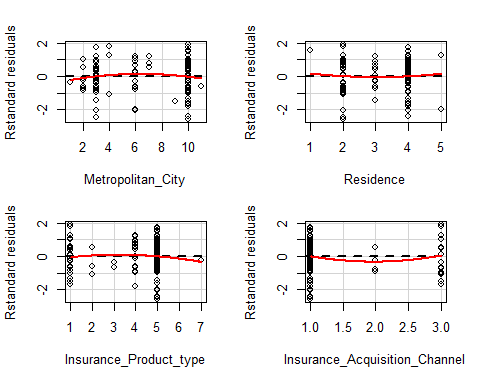
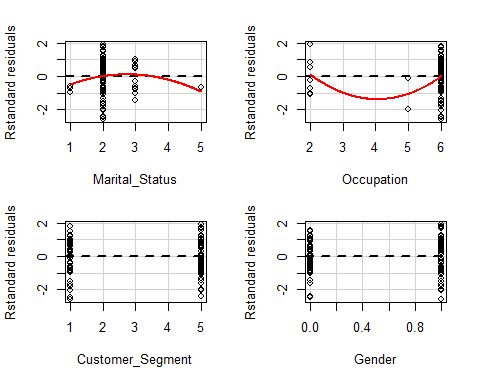
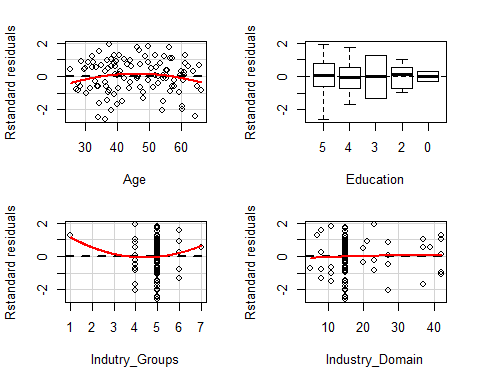
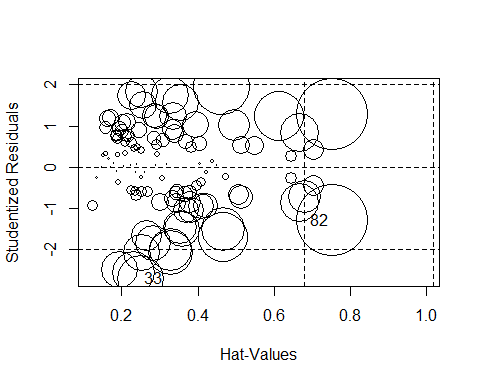
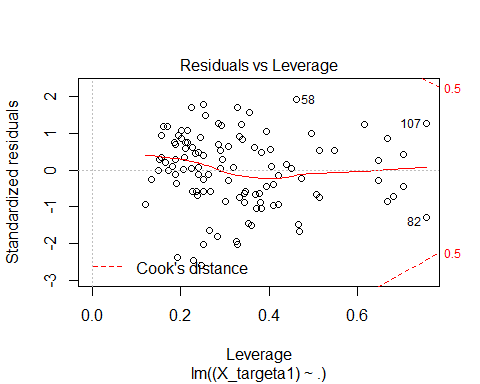
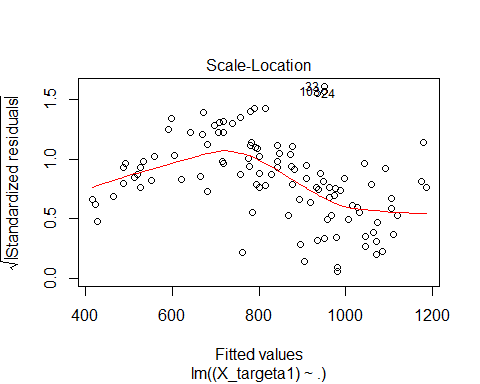


dia(ma1)

## Warning: not plotting observations with leverage one:  
## 59



## Warning: not plotting observations with leverage one:  
## 59

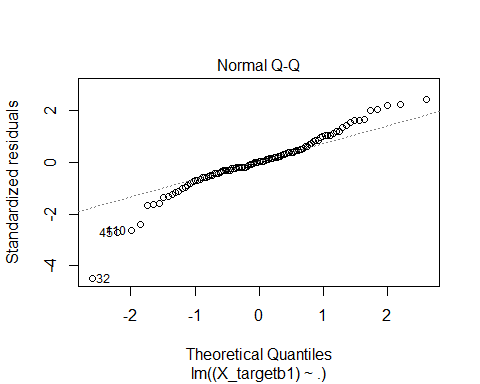
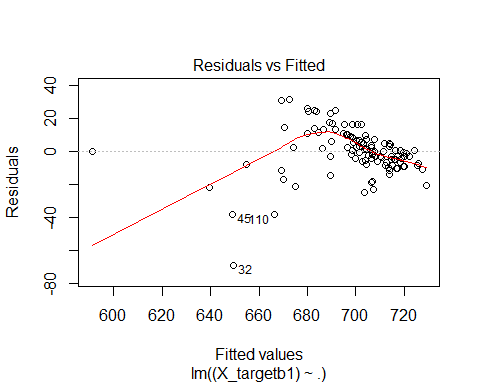


mb1<-lm((X\_targetb1)~.,data = X\_train1)  
summary(mb1)

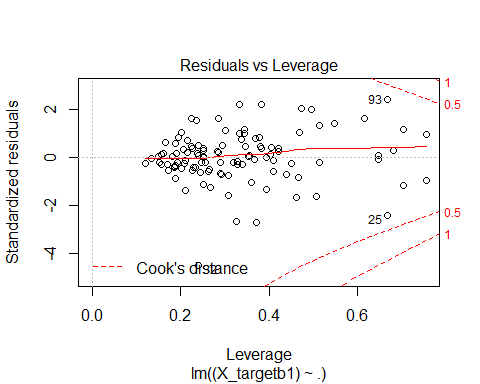
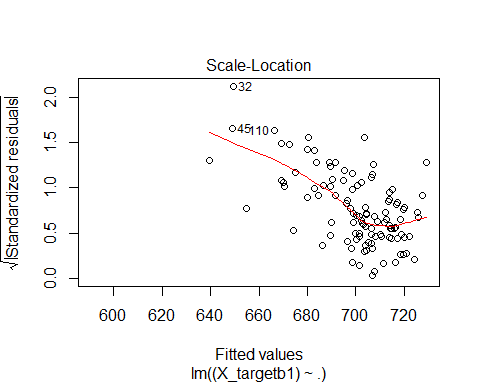
##   
## Call:  
## lm(formula = (X\_targetb1) ~ ., data = X\_train1)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -68.652 -6.588 0.260 7.377 31.522   
## Residual standard error: 17.56 on 73 degrees of freedom  
## Multiple R-squared: 0.6681, Adjusted R-squared: 0.5045   
## F-statistic: 4.082 on 36 and 73 DF, p-value: 1.731e-07

plot(mb1)

## Warning: not plotting observations with leverage one:  
## 59

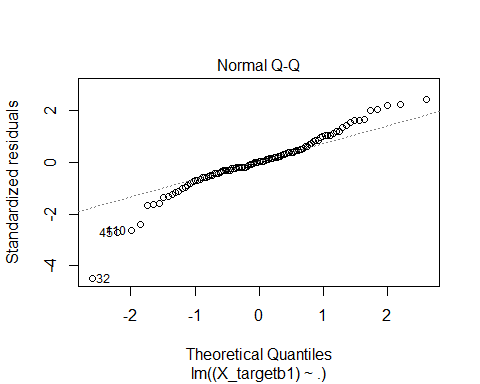
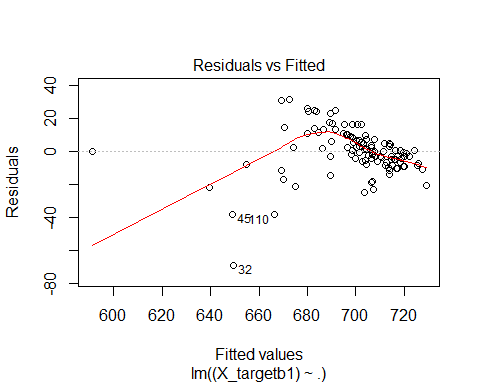


## Warning: not plotting observations with leverage one:  
## 59

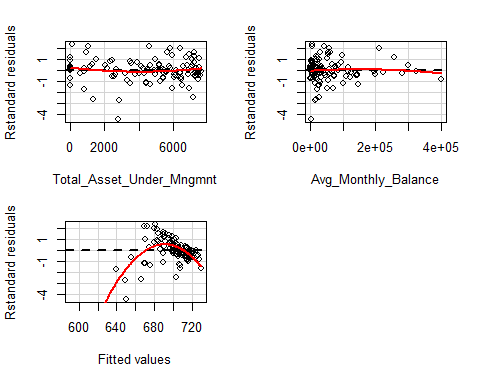
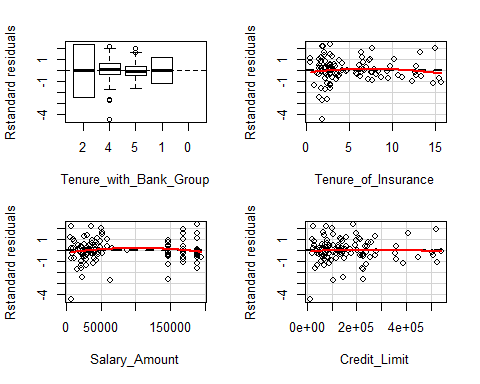
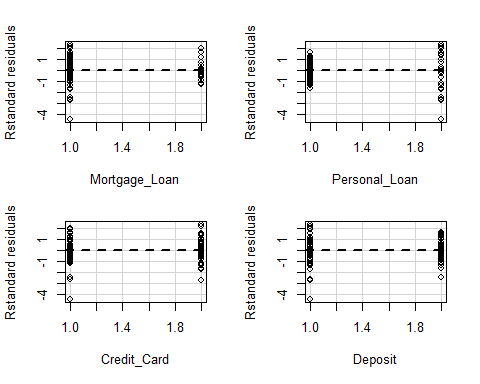
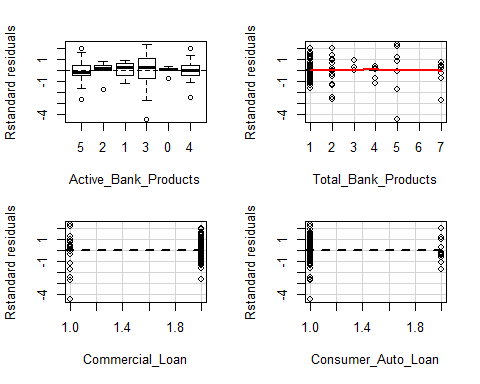
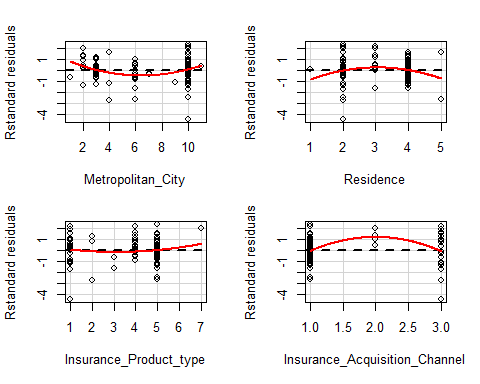
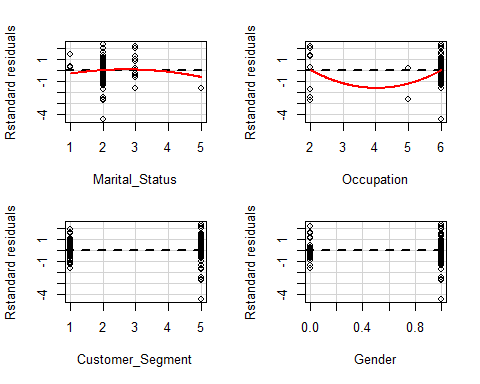
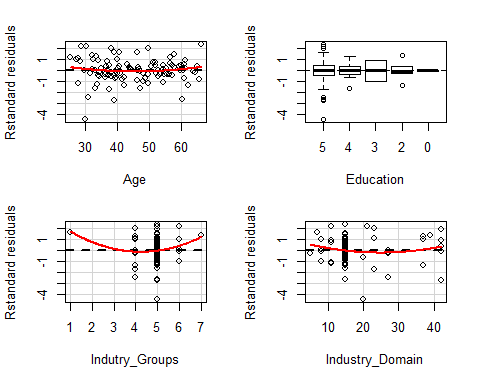
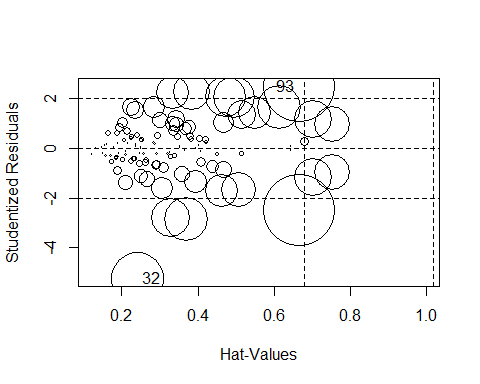
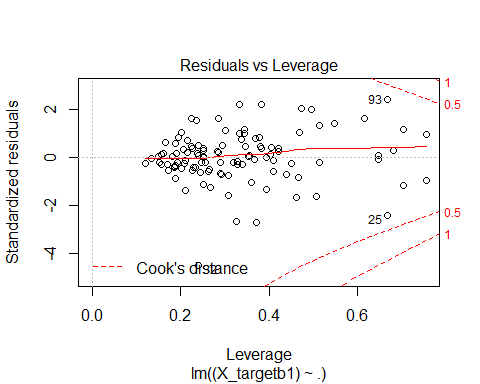
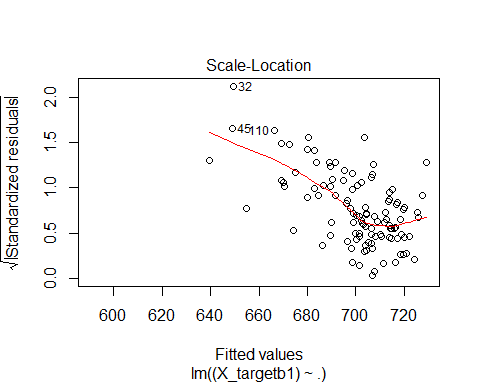


dia(mb1)

## Warning: not plotting observations with leverage one:  
## 59



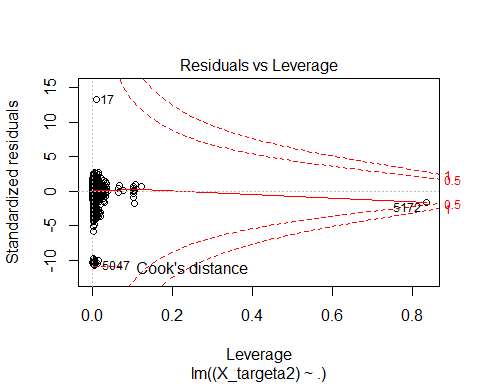
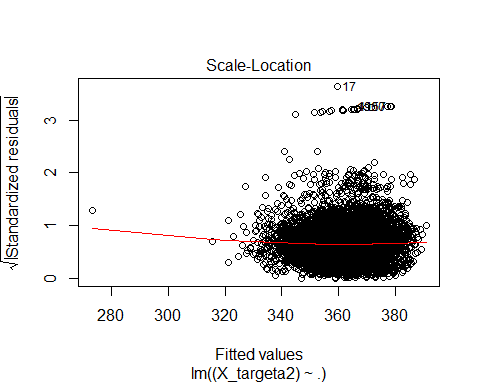
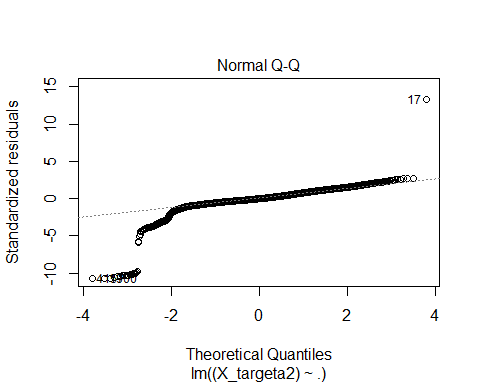
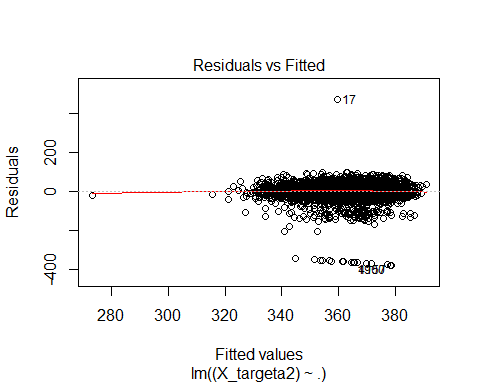
## Warning: not plotting observations with leverage one:  
## 59



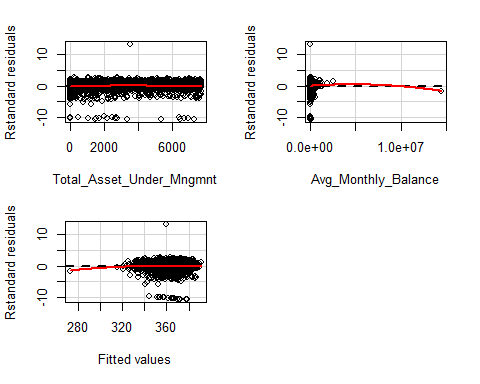
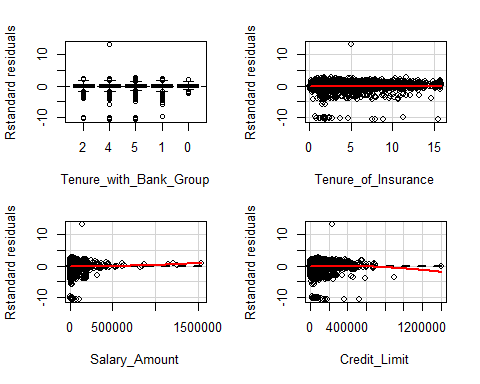
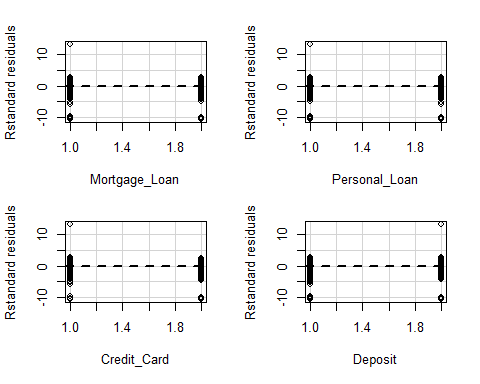
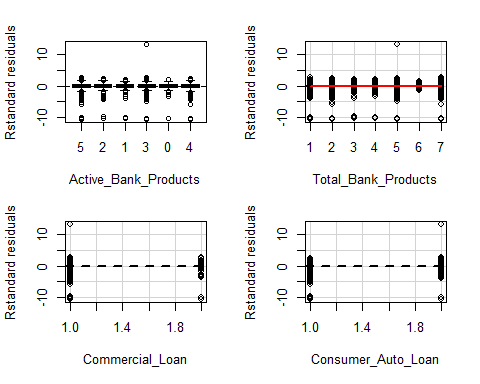
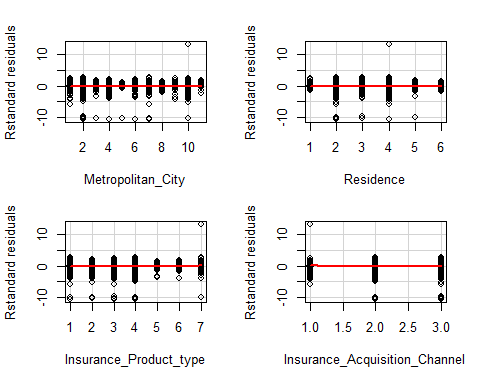
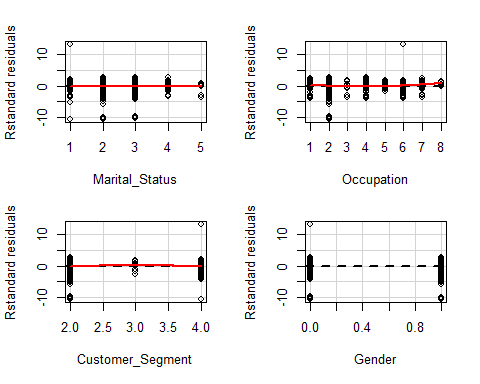
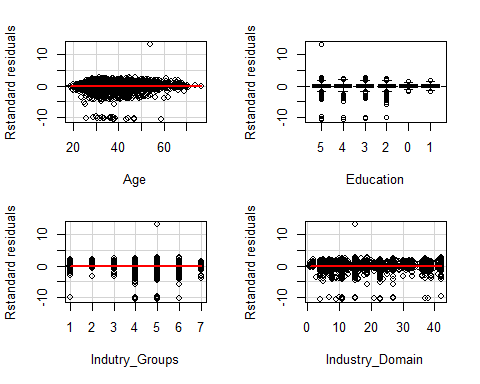
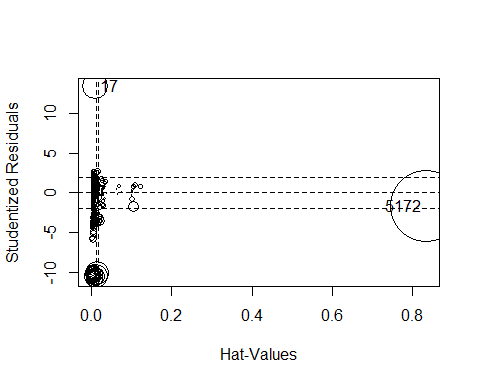
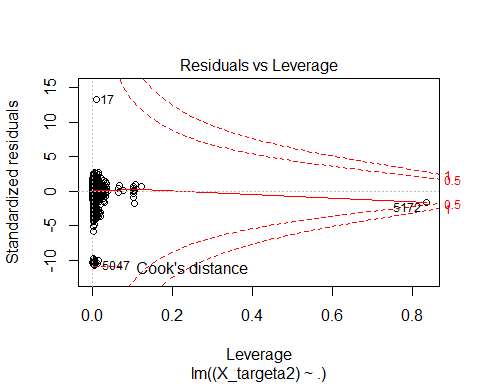
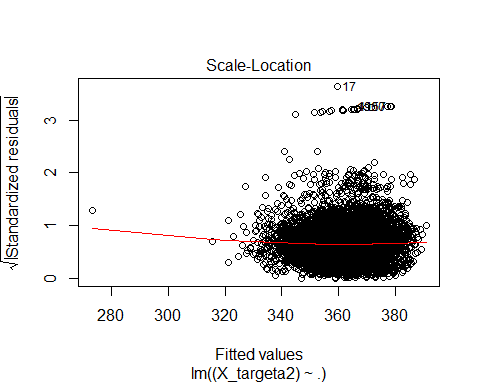
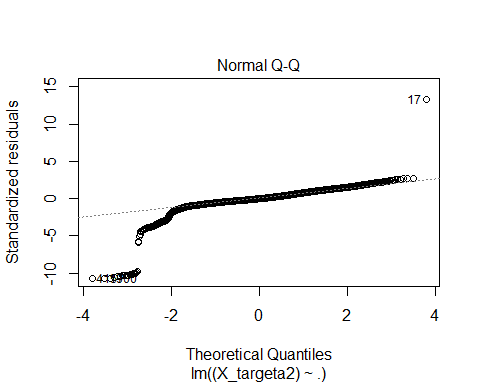
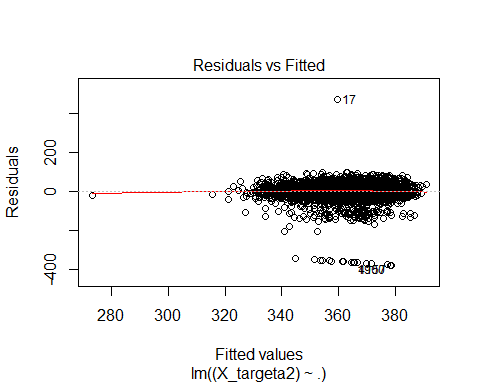
ma2<-lm((X\_targeta2)~.,data = X\_train2)  
summary(ma2)

##   
## Call:  
## lm(formula = (X\_targeta2) ~ ., data = X\_train2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -378.83 -13.05 -0.32 17.37 469.32   
## Residual standard error: 35.54 on 6775 degrees of freedom  
## Multiple R-squared: 0.07601, Adjusted R-squared: 0.07083   
## F-statistic: 14.67 on 38 and 6775 DF, p-value: < 2.2e-16

plot(ma2)



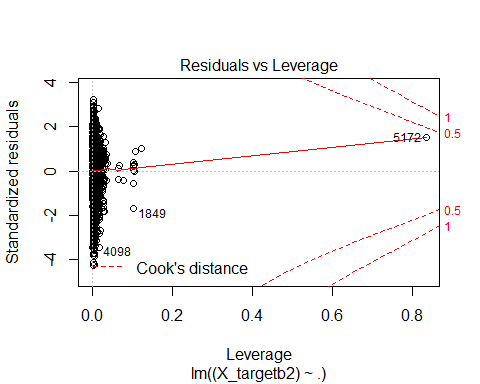
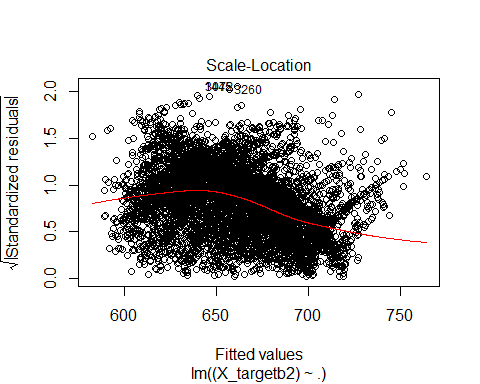
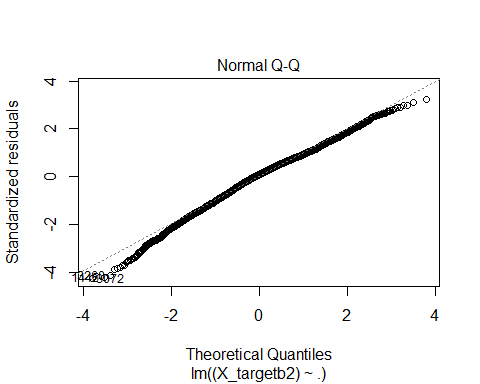
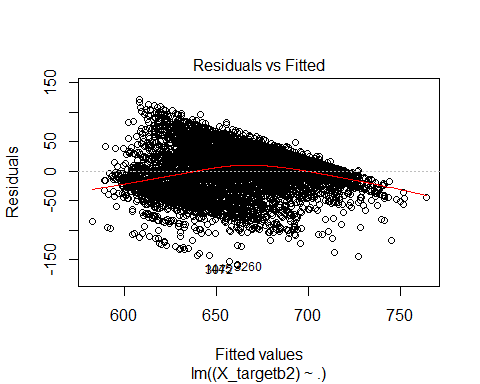
dia(ma2)



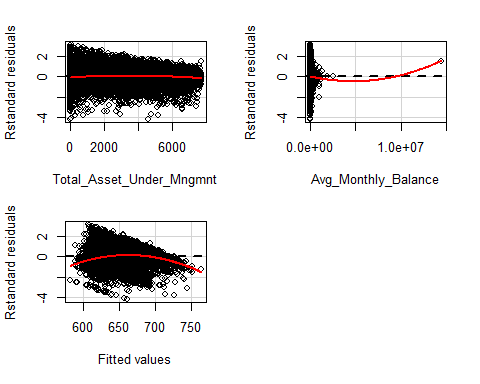
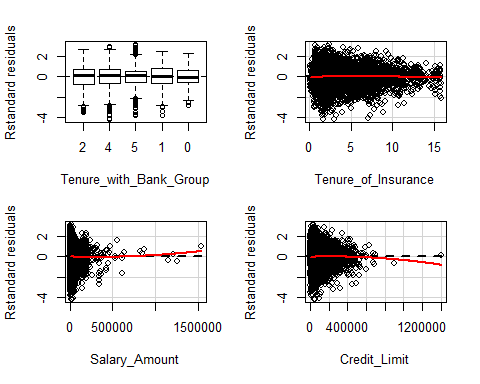
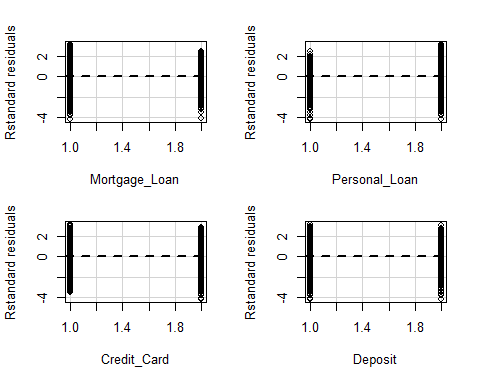
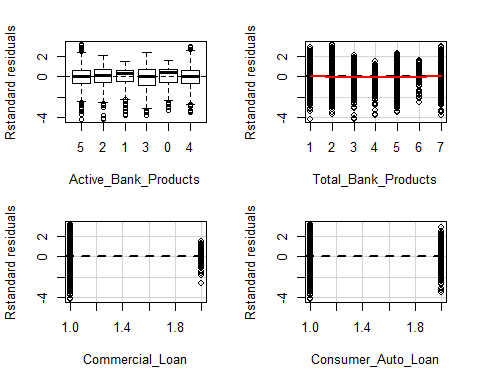
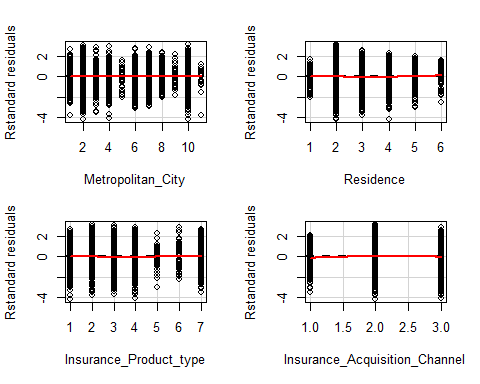
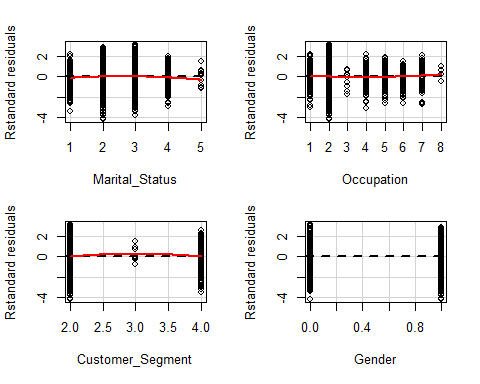
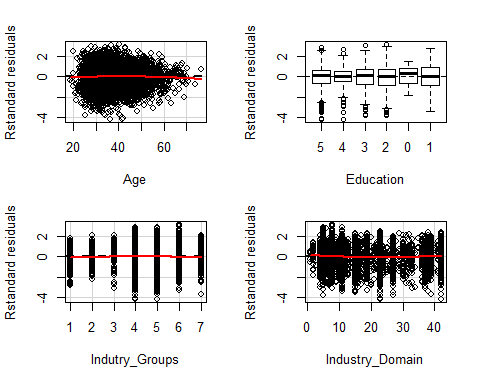
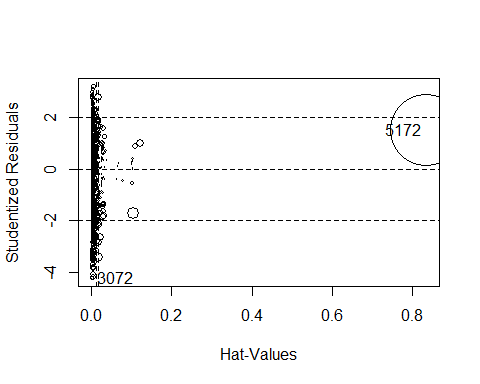
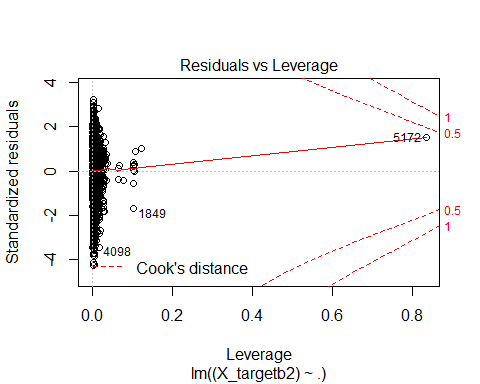
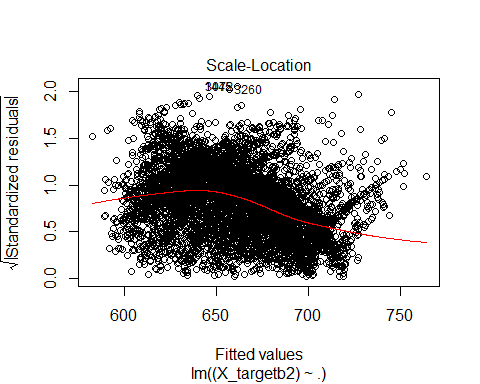
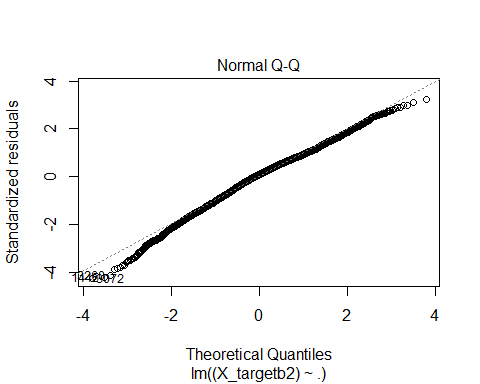
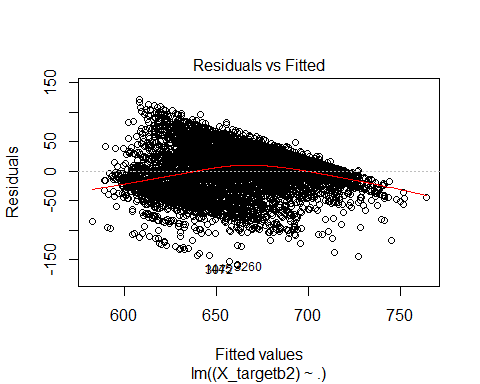
mb2<-lm((X\_targetb2)~.,data = X\_train2)  
summary(mb2)

##   
## Call:  
## lm(formula = (X\_targetb2) ~ ., data = X\_train2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -159.587 -24.574 3.659 25.535 121.137   
## Residual standard error: 37.74 on 6775 degrees of freedom  
## Multiple R-squared: 0.3896, Adjusted R-squared: 0.3862   
## F-statistic: 113.8 on 38 and 6775 DF, p-value: < 2.2e-16

plot(mb2)



dia(mb2)



xgb\_cva1 <- xgb.cv(data=data.matrix(X\_train1), nfold=10,  
 label=as.matrix(X\_targeta1), booster = "gblinear",   
 objective = "reg:linear",   
 max.depth = 20,  
 nrounds = 35,  
 lambda = 0,   
 lambda\_bias = 0,   
 alpha = 0)

## [0] train-rmse:270.014584+5.292522 test-rmse:283.015747+55.483159  
## [1] train-rmse:261.173338+6.819910 test-rmse:286.086842+60.378928  
## [2] train-rmse:255.470799+7.744083 test-rmse:283.995288+63.383184  
.  
.  
.  
## [32] train-rmse:232.526514+9.797615 test-rmse:304.345451+75.477713  
## [33] train-rmse:232.393762+9.822536 test-rmse:304.871492+75.225469  
## [34] train-rmse:232.277098+9.826984 test-rmse:304.599525+75.742134

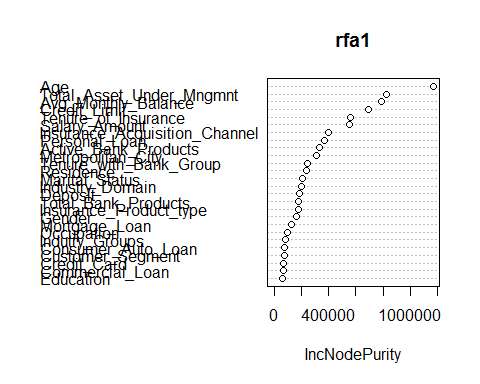
#one-hot-encoding categorical features  
ohe\_feats = c('Insurance\_Acquisition\_Channel','Insurance\_Product\_type','Residence',  
 'Metropolitan\_City','Customer\_Segment')  
a0<-rbindlist(c)  
dummies <- dummyVars(~ (Insurance\_Acquisition\_Channel+Insurance\_Product\_type  
 +Residence+Metropolitan\_City+Customer\_Segment+Identifier),sep=':',  
 fullRank=T,sparse=T,data = a0)  
a\_ohe <- as.data.table(predict(dummies, newdata =a0))  
#a\_cbd <- merge(a[,-c(which(colnames(a) %in% ohe\_feats))],a\_ohe,  
# by='Identifier',all=TRUE)  
a\_cbd <- cbind(a[,-c(which(colnames(a0) %in% ohe\_feats)),with=F],a\_ohe)  
a\_cbd <-(a\_cbd[,":="(Application\_Score=NULL,  
 Behavioural\_Score=NULL)])  
X\_train0 = a\_cbd[a\_cbd$Identifier %in% I,2:49,with=FALSE]  
X\_train01 = a\_cbd[a\_cbd$Identifier %in% I1,2:49,with=FALSE]  
X\_train02 = a\_cbd[a\_cbd$Identifier %in% I2,2:49,with=FALSE]  
ma01<-lm((X\_targeta1)~.,data = X\_train01)  
summary(ma01)

##   
## Call:  
## lm(formula = (X\_targeta1) ~ ., data = X\_train01)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -558.92 -90.53 3.84 125.64 428.33   
## Residual standard error: 265 on 58 degrees of freedom  
## Multiple R-squared: 0.5534, Adjusted R-squared: 0.1606   
## F-statistic: 1.409 on 51 and 58 DF, p-value: 0.103

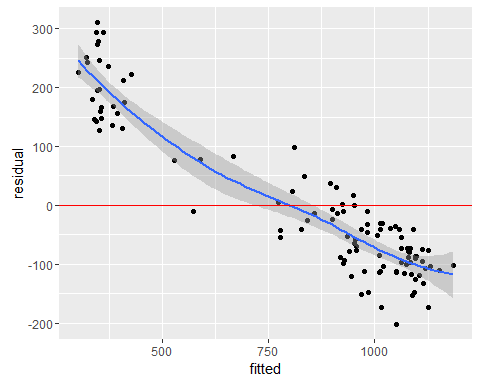
rfa1<-randomForest((X\_targeta1)~.,data = X\_train1)  
p\_rf1<-predict(rfa1,X\_train1)  
RMS(p\_rf1,X\_targeta1,36)[1]

## [1] 156.901

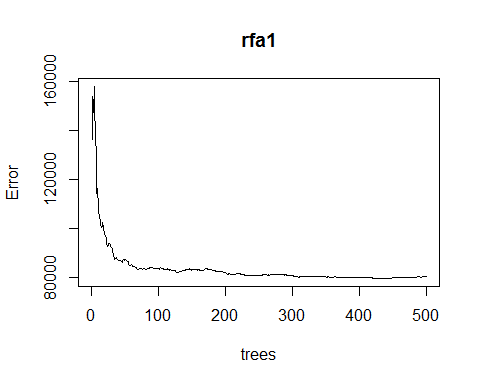
varImpPlot(rfa1)



res\_plot2(X\_targeta1,p\_rf1)



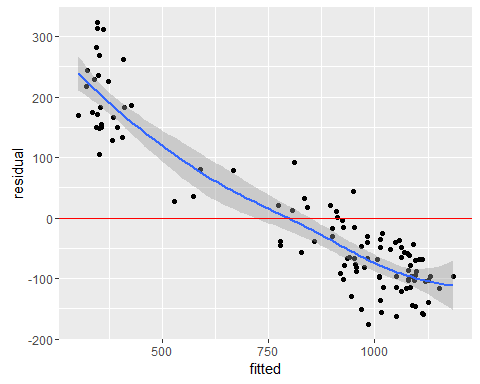
plot(rfa1)



rfa1<-randomForest((X\_targeta1)~.-(Gender+Occupation+Indutry\_Groups+  
 Consumer\_Auto\_Loan+Commercial\_Loan+Credit\_Card+  
 Customer\_Segment+Education+Mortgage\_Loan),  
 importance=F,ntree =100,data = X\_train1)  
p\_rf1<-predict(rfa1,X\_train1)  
RMS(p\_rf1,X\_targeta1,36)[1]

## [1] 156.8346

res\_plot2(X\_targeta1,p\_rf1)



pa1<-predict(rfa1,X\_test1)  
pa2<-predict(ma2,X\_test2)  
pb1<-predict(mb1,X\_test1)  
pb2<-predict(mb2,X\_test2)  
  
da1<-as.data.table(cbind(X\_test1$Identifier,pa1),check.names=T)[,.(P\_Application\_Score=pa1,Identifier=V1)]  
sol<-merge(x = sol, y = da1, by = "Identifier", all.x = TRUE)  
da2<-as.data.table(cbind(X\_test2$Identifier,pa2),check.names=T)[,.(P\_Application\_Score=pa2,Identifier=V1)]  
sol<-merge(x = sol, y = da2, by = "Identifier", all.x = TRUE)  
sol$P\_Application\_Score<- rowSums(sol[,c("P\_Application\_Score.y","P\_Application\_Score"),with=F], na.rm=T)  
  
da1<-as.data.table(cbind(X\_test1$Identifier,pb1),check.names=T)[,.(P\_Behavioural\_Score=pb1,Identifier=V1)]  
sol<-merge(x = sol, y = da1, by = "Identifier", all.x = TRUE)  
da2<-as.data.table(cbind(X\_test2$Identifier,pb2),check.names=T)[,.(P\_Behavioural\_Score=pb2,Identifier=V1)]  
sol<-merge(x = sol, y = da2, by = "Identifier", all.x = TRUE)  
sol$P\_Behavioural\_Score<- rowSums(sol[,c("P\_Behavioural\_Score.y","P\_Behavioural\_Score"),with=F], na.rm=T)  
  
p1<-sol$P\_Application\_Score  
p2<-sol$P\_Behavioural\_Score  
sol[,c("P\_Application\_Score","P\_Behavioural\_Score"):=list(p1,p2)]  
write.xlsx(sol,"BFSI - Solution submission template.xlsx",row.names=FALSE)