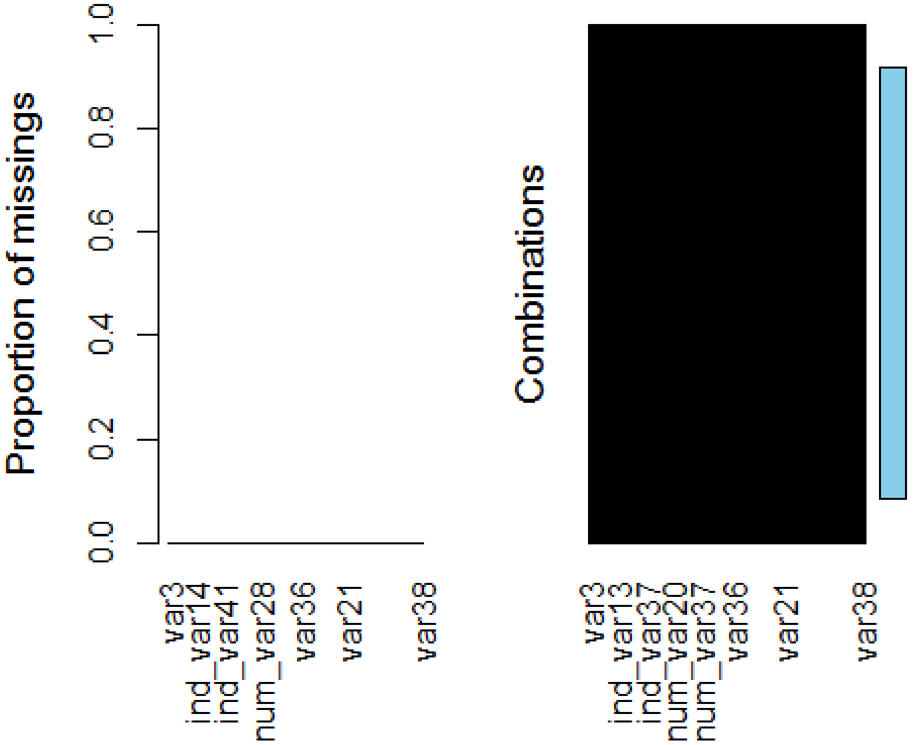
Rcode.R

vivek

Wed May 10 12:58:20 2017

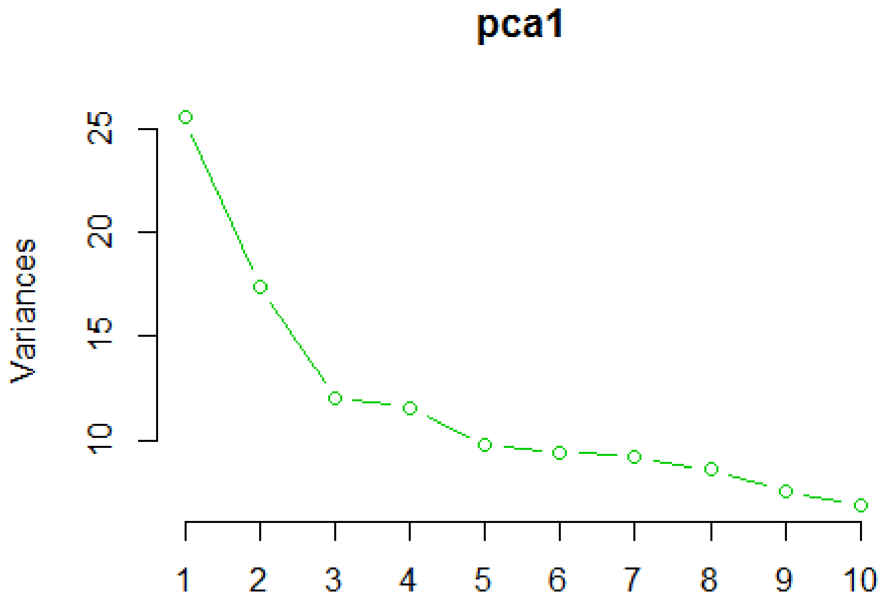
```
#########################################################
#Librries Used
#########################################################
library(xgboost)

## Warning: package 'xgboost' was built under R version 3.3.3

library(Matrix)
library(AUC)

library(ggplot2)
library(readr)

library(corrplot)

## Warning: package 'corrplot' was built under R version 3.3.3

library(glmnet)

set.seed(2908)

#########################################################
#Import Dataset
#########################################################

santander_traindataset <- read.csv("C:/Users/vivek/Desktop/Marketing
Project/train.csv")
santander_testdataset  <- read.csv("C:/Users/vivek/Desktop/Marketing
Project/test.csv")

#########################################################
#Clean the data and Count
#########################################################
santander_traindataset$ID <-NULL
##### Remove the test IDs
id <- santander_testdataset$ID
santander_testdataset$ID <-NULL
santander_traindataset$n0 <- apply(santander_traindataset, 1, FUN=function(x)
{return( sum(x == 0) )})
santander_testdataset$n0 <- apply(santander_testdataset, 1, FUN=function(x)
{return( sum(x == 0) )})

#########################################################
```

```
#Identify NULL vlues
#########################################################
aggr(santander_traindataset[1:1000,])
```

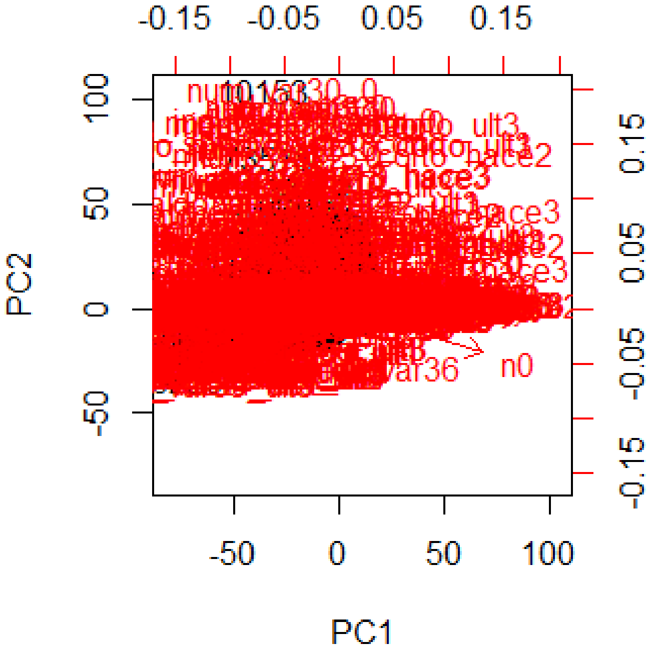


```
#########################################################
#Remove useless variables
#########################################################
for (f in names(santander_traindataset)) {
  if (length(unique(santander_traindataset[[f]])) == 1) {
    santander_traindataset[[f]] <- NULL
    santander_testdataset[[f]] <- NULL
  }}
combo <- combn(names(santander_traindataset), 2, simplify = F)
eli <- c()
for(i in combo) {
  feature1 <- i[1]
  feature2 <- i[2]
  if (!(feature1 %in% eli) & !(feature2 %in% eli)) {
    if (all(santander_traindataset[[feature1]] ==
santander_traindataset[[feature2]])) {
      eli <- c(eli, feature2)
    }}}
feature <- setdiff(names(santander_traindataset), eli)
santander_traindataset <- santander_traindataset[, feature]
feature<-feature[-307]
santander_testdataset <- santander_testdataset[, feature]
```

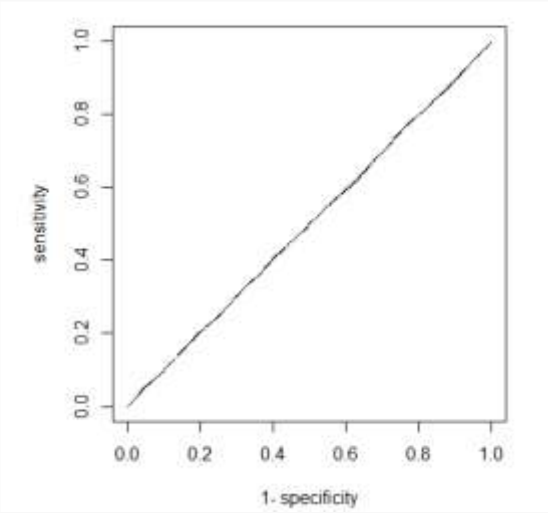
```r
#Reduce variables from santander_testdataset
for(f in colnames(santander_traindataset)[-307]){
  lim <- min(santander_traindataset[,f!="TARGET"])
  santander_testdataset[santander_testdataset[,f]<lim,f] <- lim
  lim <- max(santander_traindataset[,f!="TARGET"])
  santander_testdataset[santander_testdataset[,f]>lim,f] <- lim
}
#########################################################
#Convert to matrix
#########################################################
train<-as.matrix(santander_traindataset[,-307])
test<-as.matrix(santander_testdataset)
#######################################################
#PCA and Logistic Regression
######################################################
pca1 <- prcomp(santander_traindataset[,sapply(santander_traindataset,
       is.numeric)][-307], center = TRUE, scale. = TRUE)
screeplot(pca1, type="lines",col=3)
```



```r
biplot(pca1, scale = 0)
```



```
pcacomb<-cbind(santander_traindataset$TARGET,pca1$x)
pcacomb<- as.data.frame(pcacomb)
pca_pred <- predict(pca1, test)
pca_pred <- as.data.frame(pca_pred)
logreg<- glm(V1~PC1+PC2+PC3+PC4+PC5,data=pcacomb,family="binomial")
pred_test <- predict(logreg,pca_pred[,1:5])

pcacomb$V1<-as.factor(pcacomb$V1)
#ROC CURVE
plot(roc(pred_test,pcacomb$V1))
```

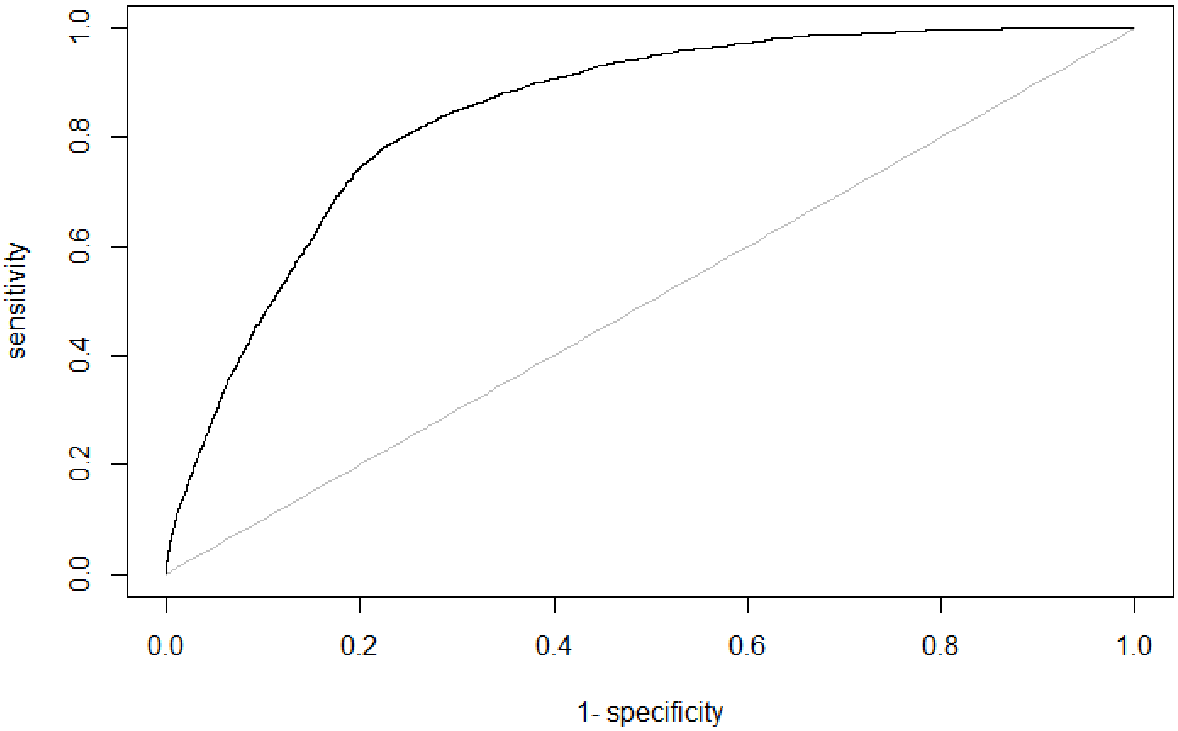


```
########################################################
#XGboost Model
########################################################
h <- sample(nrow(train),1000)
dval<-xgb.DMatrix(train[h,],
label=santander_traindataset$TARGET[h],missing=0)
dtrain <- xgb.DMatrix(train[-h,],label=santander_traindataset$TARGET[-
h],missing=0)
dtest <- xgb.DMatrix(test, missing=0)

watchlist <- list(val=dval, train=dtrain)

parameter <- list(  objective           = "binary:logistic",
                    booster             = "gbtree",
                    eval_metric         = "auc",
                    eta                 = 0.25,
                    max_depth           = 7,
                    subsample           = 0.80,
                    colsample_bytree    = 0.95
)

c <- xgb.train(   params              = parameter,
                  data                = dtrain,
                  nrounds             = 100,
                  verbose             = 1,
                  watchlist           = watchlist,
                  maximize            = TRUE

)

## [1]  val-auc:0.903202    train-auc:0.927655
## [2]  val-auc:0.935309    train-auc:0.940558
## [3]  val-auc:0.936362    train-auc:0.943090
## [4]  val-auc:0.946100    train-auc:0.951789
## [5]  val-auc:0.943645    train-auc:0.952128
## [6]  val-auc:0.940642    train-auc:0.953611
## [7]  val-auc:0.938186    train-auc:0.954621
## [8]  val-auc:0.940038    train-auc:0.957089
## [9]  val-auc:0.940431    train-auc:0.962295
## [10] val-auc:0.941554    train-auc:0.963467
## [11] val-auc:0.946437    train-auc:0.965138
## [12] val-auc:0.947602    train-auc:0.966518
## [13] val-auc:0.944248    train-auc:0.967775
## [14] val-auc:0.941596    train-auc:0.969203
## [15] val-auc:0.944346    train-auc:0.970893
## [16] val-auc:0.942564    train-auc:0.972007
## [17] val-auc:0.940375    train-auc:0.973075
## [18] val-auc:0.937779    train-auc:0.973417
## [19] val-auc:0.938074    train-auc:0.974991
## [20] val-auc:0.937484    train-auc:0.975830
.....

## [86] val-auc:0.950100    train-auc:0.998322
## [87] val-auc:0.949931    train-auc:0.998385
## [88] val-auc:0.949819    train-auc:0.998409
## [89] val-auc:0.950296    train-auc:0.998478
## [90] val-auc:0.950633    train-auc:0.998550
## [91] val-auc:0.952289    train-auc:0.998602
## [92] val-auc:0.951278    train-auc:0.998679
## [93] val-auc:0.949173    train-auc:0.998827
## [94] val-auc:0.949791    train-auc:0.998886
## [95] val-auc:0.951727    train-auc:0.998966
## [96] val-auc:0.952176    train-auc:0.999004
## [97] val-auc:0.951391    train-auc:0.999046
## [98] val-auc:0.951475    train-auc:0.999058
## [99] val-auc:0.952485    train-auc:0.999149
## [100]    val-auc:0.951643    train-auc:0.999255
```

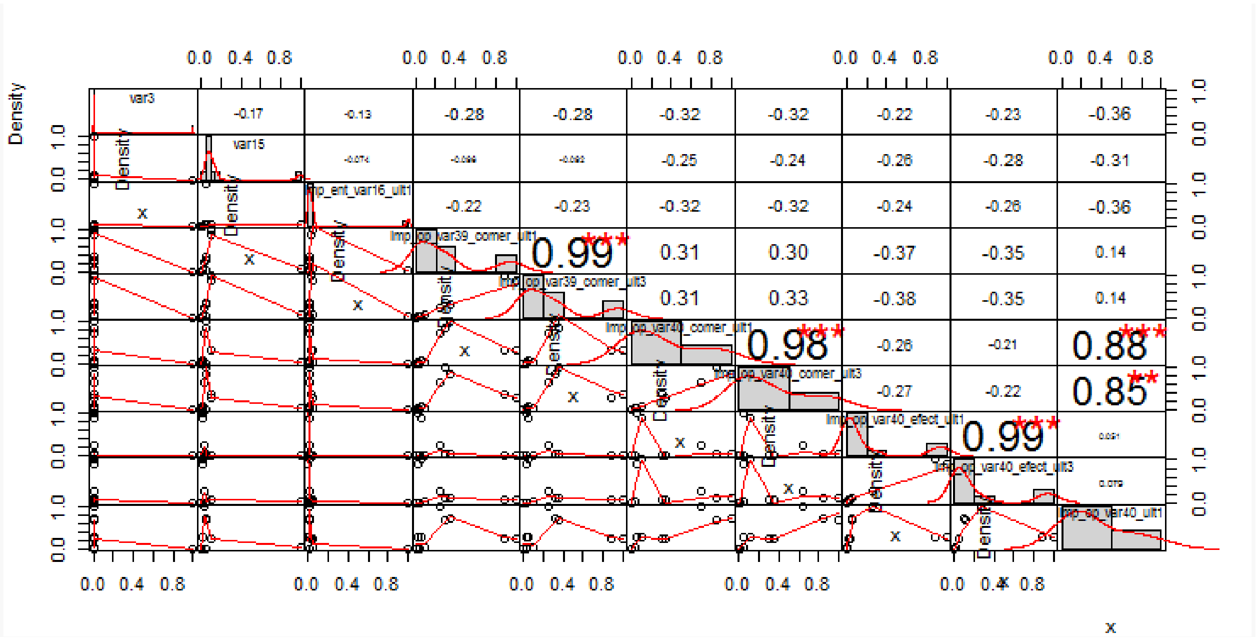
```
summary(c)
```

```
##                 Length Class              Mode
## handle               1 xgb.Booster.handle externalptr
## raw             390684 -none-             raw
## niter                1 -none-             numeric
## evaluation_log       3 data.table         list
## call                 7 -none-             call
## params               8 -none-             list
## callbacks            2 -none-             list
```

```
trainpreds <- predict(c, train)
santander_traindataset$TARGET<-as.factor(santander_traindataset$TARGET)
#ROC CURVE
plot(roc(trainpreds,santander_traindataset$TARGET))
```



```
##########################################################
#Librries Used
##########################################################
importance_matrix <- xgb.importance(feature, model = c)
#Plot Important matrix
xgb.plot.importance(importance_matrix)
#variable importance
Cor_Matrix<-cor(train[,1:10])
chart.Correlation(Cor_Matrix)
corrplot(MatrizCor_Stder,type = "upper")
```


var3
var15
Imp_ent_var16_ult1
Imp_op_var39_comer_ult1
Imp_op_var39_comer_ult3
Imp_op_var40_comer_ult1
Imp_op_var40_comer_ult3
Imp_op_var40_efect_ult1
Imp_op_var40_efect_ult3
Imp_op_var40_ult1
Density
x
0.0
0.4
0.8
1.0
-0.17
-0.13
-0.28
-0.28
-0.32
-0.32
-0.22
-0.23
-0.36
-0.074
-0.099
-0.092
-0.25
-0.24
-0.26
-0.28
-0.31
-0.22
-0.23
-0.32
-0.32
-0.24
-0.26
-0.36
0.99***
0.31
0.30
-0.37
-0.35
0.14
0.31
0.33
-0.38
-0.35
0.14
0.98***
-0.26
-0.21
0.88***
-0.27
-0.22
0.85**
0.99***
0.091
0.079