PROJECT 2.1 – PROJECT CHURN PREDICTION

R - SCRIPT

```
library(readxl)
Churn <- read_excel("Churn Updated.xls")
View(Churn)
library(plyr)
library(corrplot)
library(ggplot2)
library(gridExtra)
library(ggthemes)
library(caret)
library(MASS)
library(randomForest)
library(party)
library(ggplot2)
library(reshape2)
library(corrplot)
library(e1071)
library(caret)
library(rpart)
library(C50)
library(party)
#library(partykit)
library(randomForest)
library(ROCR)
library(dplyr)
library(car)
str(Churn)
library(Amelia)
```

```
any(is.na(Churn))
# visualize the missing values using the missing map from the Amelia package
missmap(Churn,col=c("yellow","red"))
mydata2<-Churn[,-21] mydata<-mydata2[,-
19] sapply(mydata, function(x) sum(is.na(x)))
mydata <- mydata[complete.cases(mydata), ]



intrain<- createDataPartition(mydata$Churn,p=0.8,list=FALSE)
set.seed(2017)
training<- mydata[intrain,]
testing<- mydata[-intrain,]
dim(training); dim(testing)
library (data.table)
library (plyr)
library (stringr)
LogModel <- glm(Churn ~ .,family=binomial(link="logit"),data=training)
print(summary(LogModel))
anova(LogModel, test="Chisq")
testing$Churn <- as.character(testing$Churn)
testing$Churn[testing$Churn=="No"] <- "0"
testing$Churn[testing$Churn=="Yes"] <- "1"
fitted.results <- predict(LogModel,newdata=testing,type='response')
fitted.results
misClasificError <- mean(fitted.results != testing$Churn)
print(paste('Logistic Regression Accuracy',1-misClasificError))
# calculating the accuracy rate
accuracyRate <-1-misClasificError
print(accuracyRate)
print("Confusion Matrix for Logistic Regression"); table(testing$Churn, fitted.results > 0.5)
```

```
exp(cbind(OR=coef(LogModel), confint(LogModel)))

head(mydata)

summary(mydata)

View(mydata)

sapply(mydata, sd)

cormatrix <- round(cor(mydata), digits = 2 )

cormatrix

plot.new()

plot(mydata$Churn ~mydata$`Day Mins`)

title('Basic Scatterplot')

ggplot(mydata, aes(x=mydata$`Day Mins`)) + geom_histogram(binwidth = 1, fill = "white", color = "purple")

#Randomly split data into train and test set

#80% will be ssigned to train set, 20% will be assigned to tst set

barplot(table(mydata$Churn), col= c("green", "red"), main='bar plot of Churn')

text(barplot(table(mydata$Churn), col =c('green' , 'red'), main='bar plot of Churn'), 0,table(mydata$Churn), cex =2 , pos =3)

#proportion

round(prop.table(table(mydata$Churn))*100,digits = 2)

names(mydata)

normalize<-function(x){return((x-min(x))/(max(x)-min(x)))}

mydata_n<-as.data.frame(lapply(mydata[1:18],normalize))

str(mydata)

str(mydata_n)

mydata_train<-mydata_n[1:2666,]

mydata_test<-mydata_n[2667:3333,]

mydata_train_labels<-mydata_n[1:2666,7]

mydata_test_labels<-mydata_n[2667:3333,7]

str(mydata_train)

str(mydata_train_labels)

str(mydata_test)

str(mydata_test_labels)
```

```
library(class)

#Apply knn

mydata_test_pred<-knn(train = mydata_train,test = mydata_test, cl=mydata_train_labels,k=53)

summary(mydata_test_pred)

#Evalulalte model

library(gmodels)

CrossTable(x=mydata_test_labels, y=mydata_test_pred,prop.chisq = FALSE)


sapply(mydata_n, sd)

cormatrix <- round(cor(mydata_n), digits = 2 )

cormatrix

plot.new()

plot(mydata_n$Churn ~mydata_n$Day.Mins)

title('Basic Scatterplot')

ggplot(mydata_n, aes(x=mydata_n$Day.Mins)) + geom_histogram(binwidth = 1, fill = "yellow",
color = "black")

ggplot(mydata_n, aes(x=mydata_n$CustServ.Calls)) + geom_histogram(binwidth = 1, fill =
"green", color = "red")

names(mydata_n)

#Forward elimination

#Lower AIC indicates a better model

forward <- step(glm(Churn ~ 1, data = mydata_train), direction = 'forward', scope =
~Account.Length+VMail.Message+Day.Mins + Eve.Mins +

        Night.Mins + Intl.Mins + CustServ.Calls + Int.l.Plan + VMail.Plan +

        Day.Calls + Day.Charge + Eve.Calls + Eve.Charge + Night.Calls +

        Night.Charge + Intl.Calls + Intl.Charge)

logit<- glm(Churn ~Account.Length+Day.Mins+ Day.Charge +CustServ.Calls+VMail.Plan
+Eve.Mins+ Eve.Charge+VMail.Message+Day.Calls +Eve.Calls+ Intl.Mins + Night.Calls+Intl.Calls,
data = mydata_train, family = "binomial")

summary(logit)

#evaluate model's fit and performance

influenceIndexPlot(logit, vars = c('Cook', "hat"), id.n

=4) # Confidence interval using log-likelihood
```

```
confint(logit)
exp(logit$coefficients)

# logistic regression model:
fit <- glm(Churn~.,data =mydata_train ,family = binomial(link='logit'))
summary(fit)
library(MASS)
step_fit <- stepAIC(fit,method='backward')
summary(step_fit)
confint(step_fit)
#ANOVA on base model
anova(fit,test = 'Chisq')

#ANOVA from reduced model after applying the Step
AIC anova(step_fit,test = 'Chisq')

#plot the fitted model
plot(fit$fitted.values)

pred_link <- predict(fit,newdata = mydata_test,type = 'link')

#check for multicollinearity
library(car)
vif(fit)
vif(step_fit)
pred <- predict(fit,newdata = mydata_test,type ='response')
#check the AUC curve
library(pROC)
g <- roc( Churn~ pred, data = mydata_test)
g
plot(g)
```

```
library(caret)
#with default prob cut 0.50
mydata_test$pred_Churn <- ifelse(pred<0.8,'yes','no')

table(mydata_test$pred_Churn,mydata_test$Churn)

#training split of churn classes
round(table(mydata_train$Churn)/nrow(mydata_train),2)*100
# test split of churn classes
round(table(mydata_test$Churn)/nrow(mydata_test),2)*100
#predicted split of churn classes
round(table(mydata_test$pred_Churn)/nrow(mydata_test),2)*100
#create confusion matrix
#confusionMatrix(mydata_test$Churn,mydata_test$pred_Churn)
#how do we create a cross validation scheme
control <- trainControl(method = 'repeatedcv',
                        number = 10,
                        repeats = 3)
seed <-7
metric <- 'Accuracy'
set.seed(seed)
fit_default <- train(Churn~.,
                     data = mydata_train,
                     method = 'glm',
                     metric = NaN,
                     trControl = control)
print(fit_default)
library(caret)
varImp(step_fit)
varImp(fit_default)
library(devtools)
```

```
library(woe)


library(riv)
iv_df <- iv.mult(mydata_train, y="Churn", summary=TRUE, verbose=TRUE)
iv_df
iv <- iv.mult(mydata_train, y="Churn", summary=FALSE, verbose=TRUE)
# Plot information value summary


iv.plot.summary(iv_df)
library(Rserve)
Rserve()
```