**Project - Churn Prediction (Logistic Regression)**

**R Code:**

* library(devtools)
* library(woe)
* data=read.csv(file.choose())
* iv.mult(data,"Churn",TRUE)
* str(data)
* table(data$Churn)
* index<-sample(1:nrow(data),0.7\*nrow(data))
* train\_data<-data[index,-21]
* test\_data<-data[-index,-c(8,21)]
* str(train\_data)
* str(test\_data)
* glm\_model<-glm(Churn~.,family=binomial,data=train\_data)
* glm\_model
* pred<-predict(glm\_model,test\_data,type = "response")
* pred
* hist(pred)
* plot(data$Churn[-index]~pred)
* gg=floor(pred+0.5)
* gg
* table(gg)
* ttt=table(data$Churn[-index],gg)
* ttt

**##########################Overcoming Imbalanced dataset###################**

* data\_1<-data[data$Churn==1,]
* ind\_1<-sample(rownames(data\_1),483)
* data\_0<-data[data$Churn==0,]
* ind\_0<-sample(rownames(data\_0),483)
* train\_data1<-data[c(ind\_1,ind\_0), -21 ]
* str(train\_data1)
* table(train\_data1$Churn)
* glm\_model1<-glm(Churn~.,family=binomial,train\_data1)
* glm\_model1
* pred1<-predict(glm\_model1,test\_data,type = "response")
* pred1
* hist(pred1)
* plot(data$Churn[-index]-pred1)
* gg1=floor(pred1+0.5)
* gg1
* ttt1=table(data$Churn[-index],gg1)
* ttt1
* exp(cbind(Odds\_and\_OR=coef(glm\_model1),confint(glm\_model1)))

**######################Accuracy measures##########################**

* acc<-(ttt1[1]+ttt1[4])/(ttt1[1]+ttt1[3]+ttt1[2]+ttt1[4]) #accuracy
* acc
* install.packages("ROCR")
* library(caret)
* confusionMatrix(ttt1)
* sens=ttt1[1]/(ttt1[1]+ttt1[3]) #sensitivity
* sens
* spec=ttt1[4]/(ttt1[4]+ttt1[2]) #specificity
* spec

**################### Visualizing Performance Tradeoffs ###################**

* library(ROCR)
* pred5<-predict(glm\_model1,train\_data1,type = "response")
* pred2<-prediction(predictions = pred5,labels = train\_data1$Churn )
* pred2
* eval<- performance(pred2,measure = "tpr", x.measure = "fpr")
* plot(eval,colorize=T, main="ROC curve",col="blue",lwd=5)
* auc<-performance(pred2,measure = "auc")
* str(auc)
* as.numeric(auc@y.values)

**######################## Optimum threshold & Accuracy ################**

* eval1=performance(pred2,measure = "acc")
* plot(eval1)
* max<- which.max(slot(eval1,"y.values")[[1]])
* acc<- slot(eval1,"y.values")[[1]][max]
* cut<- slot(eval1,"x.values")[[1]][max]
* print(c(Accuracy=acc, Cutoff=cut)

########**############# Connecting R to Tableau ###########################**

* install.packages("Rserve")
* library(Rserve)
* Rserve()