**CODE**

**Data Cleaning - Python**

import pandas as pd

import numpy as np

#functions

def mis\_na(x):

return sum(x.isnull())

#outliers function

def get\_median\_filtered(signal, threshold=3):

# signal = signal.copy()

difference = np.abs(signal - np.median(signal))

median\_difference = np.median(difference)

if median\_difference == 0:

s = 0

else:

s = difference / float(median\_difference)

mask = s > threshold

signal[mask] = np.median(signal)

return signal

##July data

July\_data = pd.read\_csv("E:/Churn Data/churn data/Data\_msk/July\_data1.csv")

July\_data = pd.DataFrame(July\_data)

list(July\_data)

mis\_na(July\_data)

#July\_data.apply(mis\_na,axis=0)

July\_data.describe()

July\_data.head()

July\_data.dtypes

#Outliers

July\_data.describe()

data=July\_data.ix[:,1:5]

data.describe()

mis\_na(data)

g=get\_median\_filtered(data)

g.describe()

g.head()

mis\_na(g)

type(g)

b=pd.concat([July\_data['id'],g], axis=1 ,ignore\_index=False)

b

b.head()

mis\_na(b)

#July pay

July\_pay = pd.read\_csv("E:/Churn Data/churn data/Data\_msk/July\_pay1.csv")

July\_pay = pd.DataFrame(July\_pay)

July\_pay

3

mis\_na(July\_pay)

July\_pay.describe()

July\_pay.head()

July\_pay.dtypes

#Outliers

July\_pay.describe()

pay=July\_pay.ix[:,6:11]

pay.describe()

h=get\_median\_filtered(pay)

h.describe()

c=pd.concat([July\_pay[["id", "mob\_actv1"]],h], axis=1 ,ignore\_index=False)

c.head()

c

#July usage

July\_usage = pd.read\_csv("E:/Churn Data/churn data/Data\_msk/Jul\_usg1.csv")

July\_usage = pd.DataFrame(July\_usage)

July\_usage

list(July\_usage)

mis\_na(July\_usage)

July\_usage.describe()

July\_usage.head(10)

July\_usage.dtypes

sum(July\_usage.isnull().values.ravel())

July\_usage.median()

July\_usage.fillna('NAN')

A=July\_usage.fillna(July\_usage.median())

A

mis\_na(A)

sum(A.isnull().values.ravel())

#Outliers

A.describe()

usage=A.ix[:,1:17]

usage.describe()

f= get\_median\_filtered(usage)

f.describe()

f.head()

d=pd.concat([A['id'],f], axis=1 ,ignore\_index=False)

d

d.head()

#merging of data

result = pd.merge(b, c, how='outer',on=['id'])

result

mis\_na(result)

type(result)

result1 = pd.merge(result, d ,how='outer', on=['id'])

result1

mis\_na(result1)

len(mis\_na(result1))

sum(result1.isnull().values.ravel())

type(result1)

result1.describe()

result1.head()

result1.dtypes

#replace na with o

July\_final = result1.replace(np.nan,'0', regex=True)

mis\_na(July\_final)

July\_final

July\_final.describe()

July\_final.head()

#find unique values

z=July\_final.loc[~July\_final['id'].duplicated()]

z

len(z)

#write csv

z.to\_csv('E:\churn data new\July\_final.csv', index = False)

z.head()

We applied the above code on all the 5 months. And merged all the 5 months after cleaning into one for a Single View.

**Code for Single View – Python**

import pandas as pd

import numpy as np

#functions

def mis\_na(x):

return sum(x.isnull())

July\_data = pd.read\_csv("E:/churn data new/July\_final.csv")

July\_data = pd.DataFrame(July\_data)

July\_data.head()

August\_data = pd.read\_csv("E:/churn data new/Aug\_final.csv")

August\_data = pd.DataFrame(August\_data)

August\_data.head()

September\_data = pd.read\_csv("E:/churn data new/sep\_final.csv")

September\_data = pd.DataFrame(September\_data)

September\_data

October\_data = pd.read\_csv("E:/churn data new/Oct\_final.csv")

October\_data = pd.DataFrame(October\_data)

October\_data.head()

November\_data = pd.read\_csv("E:/churn data new/November\_final.csv")

November\_data = pd.DataFrame(November\_data)

November\_data.head()

list(November\_data)

#merge\_df = pd.merge(August\_data,July\_data , on='id', how='left').fillna(0)

#merge\_df.to\_csv('merged.csv', index=False)

merge = pd.merge(July\_data, August\_data, on='id', how='outer')

merge.head()

type(merge)

mis\_na(merge)

list(merge)

merge1 = pd.merge(merge, September\_data, on='id', how='outer')

type(merge1)

merge1.head()

list(merge1)

merge2 = pd.merge(merge1, October\_data, on='id', how='outer')

merge2.head()

list(merge2)

merge3 = pd.merge(merge2, November\_data, on='id', how='outer')

merge3.head()

final\_data = pd.DataFrame(merge3)

final\_data.head()

list(merge3)

#write csv

final\_data.to\_csv('E:\churn data new\New\_final1.csv', index = False)

final\_data.head()

final\_data.tail()

mis\_na(final\_data)

**Data Processing on R and RHadoop**

Sys.setenv("HADOOP\_PREFIX"="/opt/hadoop")

Sys.setenv("HADOOP\_CMD"="/opt/hadoop/bin/hadoop")

Sys.setenv("HADOOP\_STREAMING"="/opt/hadoop/contrib/streaming/hadoop-streaming-1.2.1.jar")

################# Install packages ######################

library(dplyr)

library(rmr2)

library(rhdfs)

hdfs.init()

library(data.table)

library(lubridate)

library(pROC)

library(caret)

################# Read the files ########################

sd1 =fread(paste("/opt/hadoop/bin/hadoop", "fs -text /tmp/nikhilproject/new\_final1.csv"))

sd2 =fread(paste("/opt/hadoop/bin/hadoop", "fs -text /tmp/nikhilproject/Discon1.csv"))

#View(sd1)

#View(sd2)

#names(sd1)

#Names(sd2)

sapply(sd1, function(x) sum(is.na(x)))

################# Change the date format #################

#for july month

sd1$mob\_actv1\_x <-format(dmy(sd1$mob\_actv1\_x),"%d/%m/%Y")

sum(is.na(sd1$mob\_actv1\_x))

sd1$mob\_actv1\_x <-replace(sd1$mob\_actv1\_x,is.na(sd1$mob\_actv1\_x),"01/07/2013")

#sum(is.na(sd1$mob\_actv1))

#for august month

sd1$mob\_actv1\_y <-format(dmy(sd1$mob\_actv1\_y),"%d/%m/%Y")

sd1$mob\_actv1\_y <-replace(sd1$mob\_actv1\_y,is.na(sd1$mob\_actv1\_y),"01/08/2013")

#sum(is.na(sd1$mob\_actv1\_y))

###############change the coloumn names ############

names(sd1)[58]

names(sd1)[58]<-"mob\_act\_x2"

names(sd1)[84]

names(sd1)[84]<- "mob\_act\_y2"

names(sd1)[110]

names(sd1)[6]

names(sd1)[32]

names(sd1)[58]

names(sd1)[84]

names(sd1)[110]

#for sep data

sd1$mob\_act\_x2 <-format(dmy(sd1$mob\_act\_x2),"%d/%m/%Y")

sd1$mob\_act\_x2 <-replace(sd1$mob\_act\_x2,is.na(sd1$mob\_act\_x2),"01/09/2013")

sum(is.na(sd1$mob\_act\_x2))

#for oct data

sd1$mob\_act\_y2 <-format(dmy(sd1$mob\_act\_y2),"%d/%m/%Y")

sd1$mob\_act\_y2 <-replace(sd1$mob\_act\_y2,is.na(sd1$mob\_act\_y2),"01/10/2013")

sum(is.na(sd1$mob\_act\_y2))

#for Nov data

sd1$mob\_actv1 <-format(dmy(sd1$mob\_actv1),"%d/%m/%Y")

sd1$mob\_actv1 <-replace(sd1$mob\_actv1,is.na(sd1$mob\_actv1),"01/11/2013")

sum(is.na(sd1$mob\_actv1))

##################### creating new variables ####################

by\_id = group\_by(sd1,id)

newdata <- by\_id %>%

mutate(Total\_Bill = var15a\_x\_x+var15a\_x\_y+var15a\_x\_x+var15a\_x\_y+var15a\_x) %>%

mutate(RentalCharge = var16a\_x\_x+var16a\_x\_y+var16a\_x\_x+var16a\_x\_y+var16a\_x) %>%

mutate(NonRentalCharge = var17a\_x\_x+var17a\_x\_y+var17a\_x\_x+var17a\_x\_y+var17a\_x) %>%

mutate(Adj = var18a\_x\_x+var18a\_x\_y+var18a\_x\_x+var18a\_x\_y+var18a\_x) %>%

mutate(Usage = var19a\_x\_x+var19a\_x\_y+var19a\_x\_x+var19a\_x\_y+var19a\_x) %>%

mutate(LOC\_OG\_XYZ2XYZ\_MOU = var2a\_x+var2a\_y+var2a\_x+var2a\_y+var2a) %>%

mutate(STD\_OG\_XYZ2XYZ\_MOU = var3a\_x+var3a\_y+var3a\_x+var3a\_y+var3a) %>%

mutate(LOC\_OG\_XYZ2M\_MOU = var4a\_x+var4a\_y+var4a\_x+var4a\_y+var4a) %>%

mutate(STD\_OG\_XYZ2M\_MOU = var5a\_x+var5a\_y+var5a\_x+var5a\_y+var5a) %>%

mutate(STD\_OG\_MOU = var6a\_x+var6a\_y+var6a\_x+var6a\_y+var6a) %>%

mutate(ISD\_OG\_MOU = var9a\_x+var9a\_y+var9a\_x+var9a\_y+var9a) %>%

mutate(TOTAL\_OG\_MOU = var10a\_x+var10a\_y+var10a\_x+var10a\_y+var10a) %>%

mutate(LOC\_IC\_XYZ2XYZ\_MOU = var11a\_x+var11a\_y+var11a\_x+var11a\_y+var11a) %>%

mutate(STD\_IC\_XYZ2XYZ\_MOU = var12a\_x+var12a\_y+var12a\_x+var12a\_y+var12a) %>%

mutate(LOC\_IC\_XYZ2M\_MOU = var13a\_x+var13a\_y+var13a\_x+var13a\_y+var13a) %>%

mutate(STD\_IC\_XYZ2M\_MOU = var14a\_x+var14a\_y+var14a\_x+var14a\_y+var14a) %>%

mutate(STD\_IC\_MOU = var15a\_y\_x+var15a\_y\_y+var15a\_y\_x+var15a\_y\_y+var15a\_y) %>%

mutate(ISD\_IC\_MOU = var16a\_y\_x+var16a\_y\_y+var16a\_y\_x+var16a\_y\_y+var16a\_y) %>%

mutate(ROAM\_OG\_MOU = var17a\_y\_x+var17a\_y\_y+var17a\_y\_x+var17a\_y\_y+var17a\_y) %>%

mutate(ROAM\_IC\_MOU = var18a\_y\_x+var18a\_y\_y+var18a\_y\_x+var18a\_y\_y+var18a\_y) %>%

mutate(TOTAL\_IC\_MOU = var19a\_y\_x+var19a\_y\_y+var19a\_y\_x+var19a\_y\_y+var19a\_y)

newdata <- as.data.frame(newdata)

sum(is.na(newdata))

names(newdata)

View(newdata)

################## separating the required variables ###################

newf1=as.data.frame(newdata)[,c(1,2,3,4,5,6,28,29,30,31,32,54,55,56,57,58,80,81,82,83,84,106,107,108,109,110,132:152)]

names(newf1)

View(newf1)

class(sd1)

############### Change the date format of disconnection data ############

sd2$Date2 <-format(dmy(sd2$Date2),"%d/%m/%Y")

View(sd2)

names(sd2)[1]<-'id'

################# merging the files ####################

finalmerge = merge(newf1,sd2,by='id',all = TRUE)

finalmerge$Date2 <-ifelse(is.na(finalmerge$Date2),"30/11/2014",finalmerge$Date2)

sum(is.na(finalmerge$Date2))

finalmerge$churn\_status<-ifelse(finalmerge$Date2<"30/11/2014",1,0)

View(finalmerge$churn\_status)

table(finalmerge$churn\_status)

finalmerge[is.na(finalmerge)] <- 0

View(finalmerge)

names(finalmerge)

sum(is.na(finalmerge))

########## Write the merged file ##############

write.csv(finalmerge,"/home/hduser/finalmerge.csv")

##########read the merged file ################

sd3 =fread(paste("/home/hduser/finalmerge.csv"))

View(sd3)

sapply(sd3, function(x) sum(is.na(x)))

############# Change the data type of date ##################

sd3$jul\_act\_date=as.Date(sd3$mob\_actv1\_x)

class(sd3$jul\_act\_date)

sd3$aug\_act\_date=as.Date(sd3$mob\_actv1\_y)

class(sd3$aug\_act\_date)

sd3$sep\_act\_date=as.Date(sd3$mob\_act\_x2)

class(sd3$sep\_act\_date)

sd3$oct\_act\_date=as.Date(sd3$mob\_act\_y2)

class(sd3$oct\_act\_date)

sd3$nov\_act\_date=as.Date(sd3$mob\_actv1)

class(sd3$nov\_act\_date)

sd3$mob\_actv1\_x<-NULL

sd3$mob\_actv1\_y<-NULL

sd3$mob\_act\_x2<-NULL

sd3$mob\_act\_y2<-NULL

sd3$mob\_actv1<-NULL

View(sd3)

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

class1<-subset(sd3,churn\_status=="1")

nrow(class1)

class0<-subset(sd3,churn\_status=="0")

nrow(class0)

s1<-sample(nrow(class1))

s0<-sample(nrow(class0))

train1<-class1[s1[1:round(nrow(class1)\*0.70)],]

test1 <-class1[s1[(round(nrow(class1)\*0.70)+1):nrow(class1)],]

train0<-class0[s0[1:round(nrow(class0)\*0.70)],]

test0 <-class0[s0[(round(nrow(class0)\*0.70)+1):nrow(class0)],]

train <-rbind(train1,train0)

test <-rbind(test1,test0)

dim(train)

dim(test)

View(train)

################## Write the train and test files ##################

write.csv(train,"/home/hduser/train.csv")

write.csv(test,"/home/hduser/test.csv")

################### For train data #######################

sd4 =fread(paste("/home/hduser/train.csv"))

sum(is.na(sd4))

View(sd4)

sd4$V1<- NULL

sd4$V1<- NULL

sd4$id<- NULL

View(sd4)

sd5=na.omit(sd4)

sum(is.na(sd5))

sd5= as.data.frame(sd5)

#View(sd5)

sd5$Date2<-format(dmy(sd5$Date2),"%Y/%m/%d")

sd5$jul\_act\_date<-format(dmy(sd5$jul\_act\_date),"%Y/%m/%d")

sd5$aug\_act\_date<-format(dmy(sd5$aug\_act\_date),"%Y/%m/%d")

sd5$sep\_act\_date<-format(dmy(sd5$sep\_act\_date),"%Y/%m/%d")

sd5$oct\_act\_date<-format(dmy(sd5$oct\_act\_date),"%Y/%m/%d")

sd5$nov\_act\_date<-format(dmy(sd5$nov\_act\_date),"%Y/%m/%d")

sd5$Date2=as.Date(sd5$Date2)

sd5$jul\_act\_date=as.Date(sd5$jul\_act\_date)

sd5$aug\_act\_date=as.Date(sd5$aug\_act\_date)

sd5$sep\_act\_date=as.Date(sd5$sep\_act\_date)

sd5$oct\_act\_date=as.Date(sd5$oct\_act\_date)

sd5$nov\_act\_date=as.Date(sd5$nov\_act\_date)

fit=glm(churn\_status ~.,family = binomial(link="logit"), data = sd5)

summary(fit)

#out=step(fit)

#summary(out)

################### ForTest data #####################

sd6 =fread(paste("/home/hduser/test.csv"))

sum(is.na(sd6))

View(sd6)

sd6$V1<- NULL

sd6$V1<- NULL

sd6$id<- NULL

View(sd6)

sd7=na.omit(sd6)

sd7= as.data.frame(sd7)

sum(is.na(sd7))

sd7$Date2<-format(dmy(sd7$Date2),"%Y/%m/%d")

sd7$jul\_act\_date<-format(dmy(sd7$jul\_act\_date),"%Y/%m/%d")

sd7$aug\_act\_date<-format(dmy(sd7$aug\_act\_date),"%Y/%m/%d")

sd7$sep\_act\_date<-format(dmy(sd7$sep\_act\_date),"%Y/%m/%d")

sd7$oct\_act\_date<-format(dmy(sd7$oct\_act\_date),"%Y/%m/%d")

sd7$nov\_act\_date<-format(dmy(sd7$nov\_act\_date),"%Y/%m/%d")

sd7$Date2=as.Date(sd7$Date2)

sd7$jul\_act\_date=as.Date(sd7$jul\_act\_date)

sd7$aug\_act\_date=as.Date(sd7$aug\_act\_date)

sd7$sep\_act\_date=as.Date(sd7$sep\_act\_date)

sd7$oct\_act\_date=as.Date(sd7$oct\_act\_date)

sd7$nov\_act\_date=as.Date(sd7$nov\_act\_date)

str(sd7)

View(sd7)

######################## predict the churn ########################

n=predict(fit, sd7, type = “response”)

summary(n)

sd7$churn=ifelse(n>=0.5,1,0)

sd7$churn

View(sd7$churn)

confusionMatrix(sd7$churn\_status,sd7$churn,positive="1")

roc(sd7$churn\_status,sd7$churn,plot=TRUE)