getwd()

tele<- read.csv("C:/Users/Rahul/Downloads/Jisaw/CAPSTONE PROJECT/sampletelecomfinal.csv")

options(scipen = 999)

library(dplyr)

names(tele)

str(tele)

summary(tele)

##---------Creating Data Quality Report(dqr)-----------##

#Extracting Variable names

Variables<-names(tele)

dqr<-as.data.frame(Variables)

rm(Variables)

#Recording Data Type for each Variable

dqr$DataType<-sapply(tele,class)

#No. of Records for each Variable

dqr$No.ofRecords<-nrow(tele)

#Counting No. of Unique Values for each variable

for(i in 1:ncol(tele))

{

dqr$UniqueRecords[i]<-length(unique(tele[,i]))

}

#No.of observations available for each variable and its percentage

dqr$DataAvailable<-colSums(!is.na(tele))

dqr$AvailablePercentage<-round(colMeans(!is.na(tele)),4)

#Total and Percentage of Missing Values for each Variable

dqr$Missing<-colSums(is.na(tele))

dqr$MissingPercentage<-round(colMeans(is.na(tele)),4)

#Minimum, Maximum, Mean, Quantile Values for each Variable

for(i in 1:ncol(tele))

{

dqr$Minimum[i]<-round(ifelse(class(tele[,i])=="integer"|class(tele[,i])=="numeric",min(tele[,i],na.rm=T),0),2)

dqr$Maximum[i]<-round(ifelse(class(tele[,i])=="integer"|class(tele[,i])=="numeric",max(tele[,i],na.rm=T),0),2)

dqr$Mean[i]<-round(ifelse(class(tele[,i])=="integer"|class(tele[,i])=="numeric",mean(tele[,i],na.rm=T),0),2)

dqr$fifthPercentile[i]<-round(ifelse(class(tele[,i])=="integer"|class(tele[,i])=="numeric",quantile(tele[,i],p=0.05,na.rm=T),0),2)

dqr$tenthPercentile[i]<-round(ifelse(class(tele[,i])=="integer"|class(tele[,i])=="numeric",quantile(tele[,i],p=0.10,na.rm=T),0),2)

dqr$twentyfifthPercentile[i]<-round(ifelse(class(tele[,i])=="integer"|class(tele[,i])=="numeric",quantile(tele[,i],p=0.25,na.rm=T),0),2)

dqr$fiftythPercentile[i]<-round(ifelse(class(tele[,i])=="integer"|class(tele[,i])=="numeric",quantile(tele[,i],p=0.50,na.rm=T),0),2)

dqr$seventyfifthPercentile[i]<-round(ifelse(class(tele[,i])=="integer"|class(tele[,i])=="numeric",quantile(tele[,i],p=0.75,na.rm=T),0),2)

dqr$ninetythPercentile[i]<-round(ifelse(class(tele[,i])=="integer"|class(tele[,i])=="numeric",quantile(tele[,i],p=0.90,na.rm=T),0),2)

dqr$ninetyfifthPercentile[i]<-round(ifelse(class(tele[,i])=="integer"|class(tele[,i])=="numeric",quantile(tele[,i],p=0.95,na.rm=T),0),2)

}

str(dqr)

#Exporting Data Quality Report

write.csv(dqr,"Data Quality Report.csv",row.names = T)

#Missing Value treatment of var, "retdays" and Creating Dummy Variable

summary(tele$retdays)

sort(unique(tele$retdays), na.last = F)

tele$retdays\_1<-ifelse(is.na(tele$retdays)==TRUE, 0, 1)

str(tele$retdays\_1)

summary(tele$retdays\_1)

#Ommitting variables with more than 15% missing values and creating a new data set

tele1<-tele[,colMeans(is.na(tele))<=0.15]

#Variable drop\_blk\_Mean is created by adding vars blck\_dat\_Mean + BLCK\_VCE\_MEAN + DROP\_DAT\_MEAN + DROP\_VCE\_MEAN

# So omitting Variable blck\_dat\_Mean

names(tele1)

tele1<-tele1[,-50]

##\*\*\*\*\*\*\*\*\*\*Data Exploration => Profiling (dat-Continuous Variables , datC-Categorical Variables)\*\*\*\*\*\*\*\*\*\*##

##-------- Deciling Continuous Variables Basis Target Variabe Churn---------##

names(tele1)

str(tele1)

# <1>Variable 'mou\_Mean'

summary(tele1$mou\_Mean)

tele1%>%mutate(dec=ntile(mou\_Mean,n=10))%>%count(churn,dec)%>%filter(churn==1)->dat1

dat1$N<-unclass(tele1%>%mutate(dec=ntile(mou\_Mean,n=10))%>%count(dec)%>%unname())[[2]]

dat1$churn\_perc<-round(dat1$n/dat1$N,2)

dat1$GreaterThan<-unclass(tele1%>%mutate(dec=ntile(mou\_Mean,n=10))%>%group\_by(dec)%>%summarise(min(mou\_Mean)))[[2]]

dat1$LessThan<-unclass(tele1%>%mutate(dec=ntile(mou\_Mean,n=10))%>%group\_by(dec)%>%summarise(max(mou\_Mean)))[[2]]

dat1$varname<-rep("mou\_Mean",nrow(dat1))

# <2> Variable "totmrc\_Mean"

summary(tele1$totmrc\_Mean)

tele1%>%mutate(dec=ntile(totmrc\_Mean,n=10))%>%count(churn,dec)%>%filter(churn==1)->dat2

dat2$N<-unclass(tele1%>%mutate(dec=ntile(totmrc\_Mean,n=10))%>%count(dec)%>%unname())[[2]]

dat2$churn\_perc<-dat2$n/dat2$N

dat2$GreaterThan<-unclass(tele1%>%mutate(dec=ntile(totmrc\_Mean,n=10))%>%group\_by(dec)%>%summarise(min(totmrc\_Mean)))[[2]]

dat2$LessThan<-unclass(tele1%>%mutate(dec=ntile(totmrc\_Mean,n=10))%>%group\_by(dec)%>%summarise(max(totmrc\_Mean)))[[2]]

dat2$varname<-rep("totmrc\_Mean",nrow(dat2))

# <3> Variable "rev\_Range"

summary(tele1$rev\_Range)

tele1%>%mutate(dec=ntile(rev\_Range,n=10))%>%count(churn,dec)%>%filter(churn==1)->dat3

dat3$N<-unclass(tele1%>%mutate(dec=ntile(rev\_Range,n=10))%>%count(dec)%>%unname())[[2]]

dat3$churn\_perc<-dat3$n/dat3$N

dat3$GreaterThan<-unclass(tele1%>%mutate(dec=ntile(rev\_Range,n=10))%>%group\_by(dec)%>%summarise(min(rev\_Range)))[[2]]

dat3$LessThan<-unclass(tele1%>%mutate(dec=ntile(rev\_Range,n=10))%>%group\_by(dec)%>%summarise(max(rev\_Range)))[[2]]

dat3$varname<-rep("rev\_Range",nrow(dat3))

# <4> Variable "mou\_Range"

summary(tele1$mou\_Range)

tele1%>%mutate(dec=ntile(mou\_Range,n=10))%>%count(churn,dec)%>%filter(churn==1)->dat4

dat4$N<-unclass(tele1%>%mutate(dec=ntile(mou\_Range,n=10))%>%count(dec)%>%unname())[[2]]

dat4$churn\_perc<-dat4$n/dat4$N

dat4$GreaterThan<-unclass(tele1%>%mutate(dec=ntile(mou\_Range,n=10))%>%group\_by(dec)%>%summarise(min(mou\_Range)))[[2]]

dat4$LessThan<-unclass(tele1%>%mutate(dec=ntile(mou\_Range,n=10))%>%group\_by(dec)%>%summarise(max(mou\_Range)))[[2]]

dat4$varname<-rep("mou\_Range",nrow(dat4))

# <5> Variable "change\_mou"

summary(tele1$change\_mou)

tele1%>%mutate(dec=ntile(change\_mou,n=10))%>%count(churn,dec)%>%filter(churn==1)->dat5

dat5$N<-unclass(tele1%>%mutate(dec=ntile(change\_mou,n=10))%>%count(dec)%>%unname())[[2]]

dat5$churn\_perc<-dat5$n/dat5$N

dat5$GreaterThan<-unclass(tele1%>%mutate(dec=ntile(change\_mou,n=10))%>%group\_by(dec)%>%summarise(min(change\_mou)))[[2]]

dat5$LessThan<-unclass(tele1%>%mutate(dec=ntile(change\_mou,n=10))%>%group\_by(dec)%>%summarise(max(change\_mou)))[[2]]

dat5$varname<-rep("change\_mou",nrow(dat5))

<6> Variable "drop\_blk\_Mean"

tele1%>%mutate(dec=ntile(drop\_blk\_Mean,n=10))%>%count(churn,dec)%>%filter(churn==1)->dat6

dat6$N<-unclass(tele1%>%mutate(dec=ntile(drop\_blk\_Mean,n=10))%>%count(dec)%>%unname())[[2]]

dat6$churn\_perc<-dat6$n/dat6$N

dat6$GreaterThan<-unclass(tele1%>%mutate(dec=ntile(drop\_blk\_Mean,n=10))%>%group\_by(dec)%>%summarise(min(drop\_blk\_Mean)))[[2]]

dat6$LessThan<-unclass(tele1%>%mutate(dec=ntile(drop\_blk\_Mean,n=10))%>%group\_by(dec)%>%summarise(max(drop\_blk\_Mean)))[[2]]

dat6$varname<-rep("drop\_blk\_Mean",nrow(dat6))

# <7> Variable "drop\_vce\_Range"

summary(tele1$drop\_vce\_Range)

tele1%>%mutate(dec=ntile(drop\_vce\_Range,n=10))%>%count(churn,dec)%>%filter(churn==1)->dat7

dat7$N<-unclass(tele1%>%mutate(dec=ntile(drop\_vce\_Range,n=10))%>%count(dec)%>%unname())[[2]]

dat7$churn\_perc<-dat7$n/dat7$N

dat7$GreaterThan<-unclass(tele1%>%mutate(dec=ntile(drop\_vce\_Range,n=10))%>%group\_by(dec)%>%summarise(min(drop\_vce\_Range)))[[2]]

dat7$LessThan<-unclass(tele1%>%mutate(dec=ntile(drop\_vce\_Range,n=10))%>%group\_by(dec)%>%summarise(max(drop\_vce\_Range)))[[2]]

dat7$varname<-rep("drop\_vce\_Range",nrow(dat7))

# <8> Variable "owylis\_vce\_Range"

summary(tele1$owylis\_vce\_Range)

tele1%>%mutate(dec=ntile(owylis\_vce\_Range,n=10))%>%count(churn,dec)%>%filter(churn==1)->dat8

dat8$N<-unclass(tele1%>%mutate(dec=ntile(owylis\_vce\_Range,n=10))%>%count(dec)%>%unname())[[2]]

dat8$churn\_perc<-dat8$n/dat8$N

dat8$GreaterThan<-unclass(tele1%>%mutate(dec=ntile(owylis\_vce\_Range,n=10))%>%group\_by(dec)%>%summarise(min(owylis\_vce\_Range)))[[2]]

dat8$LessThan<-unclass(tele1%>%mutate(dec=ntile(owylis\_vce\_Range,n=10))%>%group\_by(dec)%>%summarise(max(owylis\_vce\_Range)))[[2]]

dat8$varname<-rep("owylis\_vce\_Range",nrow(dat8))

Variable "mou\_opkv\_Range"

summary(tele1$mou\_opkv\_Range)

tele1%>%mutate(dec=ntile(mou\_opkv\_Range,n=10))%>%count(churn,dec)%>%filter(churn==1)->dat9

dat9$N<-unclass(tele1%>%mutate(dec=ntile(mou\_opkv\_Range,n=10))%>%count(dec)%>%unname())[[2]]

dat9$churn\_perc<-dat9$n/dat9$N

dat9$GreaterThan<-unclass(tele1%>%mutate(dec=ntile(mou\_opkv\_Range,n=10))%>%group\_by(dec)%>%summarise(min(mou\_opkv\_Range)))[[2]]

dat9$LessThan<-unclass(tele1%>%mutate(dec=ntile(mou\_opkv\_Range,n=10))%>%group\_by(dec)%>%summarise(max(mou\_opkv\_Range)))[[2]]

dat9$varname<-rep("mou\_opkv\_Range",nrow(dat9))

# <10> Variable "months"

summary(tele1$months)

tele1%>%mutate(dec=ntile(months,n=10))%>%count(churn,dec)%>%filter(churn==1)->dat10

dat10$N<-unclass(tele1%>%mutate(dec=ntile(months,n=10))%>%count(dec)%>%unname())[[2]]

dat10$churn\_perc<-dat10$n/dat10$N

dat10$GreaterThan<-unclass(tele1%>%mutate(dec=ntile(months,n=10))%>%group\_by(dec)%>%summarise(min(months)))[[2]]

dat10$LessThan<-unclass(tele1%>%mutate(dec=ntile(months,n=10))%>%group\_by(dec)%>%summarise(max(months)))[[2]]

dat10$varname<-rep("months",nrow(dat10))

# <11> Variable "totcalls"

summary(tele1$totcalls)

tele1%>%mutate(dec=ntile(totcalls,n=10))%>%count(churn,dec)%>%filter(churn==1)->dat11

dat11$N<-unclass(tele1%>%mutate(dec=ntile(totcalls,n=10))%>%count(dec)%>%unname())[[2]]

dat11$churn\_perc<-dat11$n/dat11$N

dat11$GreaterThan<-unclass(tele1%>%mutate(dec=ntile(totcalls,n=10))%>%group\_by(dec)%>%summarise(min(totcalls)))[[2]]

dat11$LessThan<-unclass(tele1%>%mutate(dec=ntile(totcalls,n=10))%>%group\_by(dec)%>%summarise(max(totcalls)))[[2]]

dat11$varname<-rep("totcalls",nrow(dat11))

# <12> Variable "eqpdays"

summary(tele1$eqpdays)

#Missing Value Treatment - Since there is just 1 missing observation, will remove the same.

index<-which(is.na(tele1$eqpdays))

tele1<-tele1[-index,]

#Deciling basis Variable churn

tele1%>%mutate(dec=ntile(eqpdays,n=10))%>%count(churn,dec)%>%filter(churn==1)->dat12

dat12$N<-unclass(tele1%>%mutate(dec=ntile(eqpdays,n=10))%>%count(dec)%>%unname())[[2]]

dat12$churn\_perc<-dat12$n/dat12$N

dat12$GreaterThan<-unclass(tele1%>%mutate(dec=ntile(eqpdays,n=10))%>%group\_by(dec)%>%summarise(min(eqpdays)))[[2]]

dat12$LessThan<-unclass(tele1%>%mutate(dec=ntile(eqpdays,n=10))%>%group\_by(dec)%>%summarise(max(eqpdays)))[[2]]

dat12$varname<-rep("eqpdays",nrow(dat12))

#<13> Variable "custcare\_Mean"===>> \*\*\*Getting less than 4 deciles. Omit\*\*\*

summary(tele1$custcare\_Mean)

tele1%>%mutate(dec=ntile(custcare\_Mean,n=4))%>%count(churn,dec)%>%filter(churn==1)->dat13

dat13$varname<-rep("custcare\_Mean",nrow(dat13))

plot(tele1$churn,tele1$custcare\_Mean, col="red")

#<14> Variable "callwait\_Mean"

summary(tele1$callwait\_Mean)

tele1%>%mutate(dec=ntile(callwait\_Mean,n=4))%>%count(churn,dec)%>%filter(churn==1)->dat14

dat14$N<-unclass(tele1%>%mutate(dec=ntile(callwait\_Mean,n=4))%>%count(dec)%>%unname())[[2]]

dat14$churn\_perc<-dat14$n/dat14$N

dat14$GreaterThan<-unclass(tele1%>%mutate(dec=ntile(callwait\_Mean,n=4))%>%group\_by(dec)%>%summarise(min(callwait\_Mean)))[[2]]

dat14$LessThan<-unclass(tele1%>%mutate(dec=ntile(callwait\_Mean,n=4))%>%group\_by(dec)%>%summarise(max(callwait\_Mean)))[[2]]

dat14$varname<-rep("callwait\_Mean",nrow(dat14))

# <15> Variable "iwylis\_vce\_Mean"

summary(tele1$iwylis\_vce\_Mean)

tele1%>%mutate(dec=ntile(iwylis\_vce\_Mean,n=6))%>%count(churn,dec)%>%filter(churn==1)->dat15

dat15$N<-unclass(tele1%>%mutate(dec=ntile(iwylis\_vce\_Mean,n=6))%>%count(dec)%>%unname())[[2]]

dat15$churn\_perc<-dat15$n/dat15$N

dat15$GreaterThan<-unclass(tele1%>%mutate(dec=ntile(iwylis\_vce\_Mean,n=6))%>%group\_by(dec)%>%summarise(min(iwylis\_vce\_Mean)))[[2]]

dat15$LessThan<-unclass(tele1%>%mutate(dec=ntile(iwylis\_vce\_Mean,n=6))%>%group\_by(dec)%>%summarise(max(iwylis\_vce\_Mean)))[[2]]

dat15$varname<-rep("iwylis\_vce\_Mean",nrow(dat15))

# <16> Variable "callwait\_Range"===>> \*\*\*Getting less than 4 deciles. Omit\*\*\*

summary(tele1$callwait\_Range)

tele1%>%mutate(dec=ntile(callwait\_Range,n=4))%>%count(churn,dec)%>%filter(churn==1)->dat16

dat16$varname<-rep("callwait\_Range",nrow(dat16))

# <17> Variable "ccrndmou\_Range"===>> \*\*\*Getting less than 4 deciles. Omit\*\*\*

summary(tele1$ccrndmou\_Range)

tele1%>%mutate(dec=ntile(ccrndmou\_Range,n=4))%>%count(churn,dec)%>%filter(churn==1)->dat17

#<18> Variable "adjqty"

summary(tele1$adjqty)

tele1%>%mutate(dec=ntile(adjqty,n=10))%>%count(churn,dec)%>%filter(churn==1)->dat18

dat18$N<-unclass(tele1%>%mutate(dec=ntile(adjqty,n=10))%>%count(dec)%>%unname())[[2]]

dat18$churn\_perc<-dat18$n/dat18$N

dat18$GreaterThan<-unclass(tele1%>%mutate(dec=ntile(adjqty,n=10))%>%group\_by(dec)%>%summarise(min(adjqty)))[[2]]

dat18$LessThan<-unclass(tele1%>%mutate(dec=ntile(adjqty,n=10))%>%group\_by(dec)%>%summarise(max(adjqty)))[[2]]

dat18$varname<-rep("adjqty",nrow(dat18))

# <19> Variable "ovrrev\_Mean"

summary(tele1$ovrrev\_Mean)

tele1%>%mutate(dec=ntile(ovrrev\_Mean,n=4))%>%count(churn,dec)%>%filter(churn==1)->dat19

dat19$N<-unclass(tele1%>%mutate(dec=ntile(ovrrev\_Mean,n=4))%>%count(dec)%>%unname())[[2]]

dat19$churn\_perc<-dat19$n/dat19$N

dat19$GreaterThan<-unclass(tele1%>%mutate(dec=ntile(ovrrev\_Mean,n=4))%>%group\_by(dec)%>%summarise(min(ovrrev\_Mean)))[[2]]

dat19$LessThan<-unclass(tele1%>%mutate(dec=ntile(ovrrev\_Mean,n=4))%>%group\_by(dec)%>%summarise(max(ovrrev\_Mean)))[[2]]

dat19$varname<-rep("ovrrev\_Mean",nrow(dat19))

# <20> Variable "rev\_Mean"

summary(tele1$rev\_Mean)

tele1%>%mutate(dec=ntile(rev\_Mean,n=10))%>%count(churn,dec)%>%filter(churn==1)->dat20

dat20$N<-unclass(tele1%>%mutate(dec=ntile(rev\_Mean,n=10))%>%count(dec)%>%unname())[[2]]

dat20$churn\_perc<-dat20$n/dat20$N

dat20$GreaterThan<-unclass(tele1%>%mutate(dec=ntile(rev\_Mean,n=10))%>%group\_by(dec)%>%summarise(min(rev\_Mean)))[[2]]

dat20$LessThan<-unclass(tele1%>%mutate(dec=ntile(rev\_Mean,n=10))%>%group\_by(dec)%>%summarise(max(rev\_Mean)))[[2]]

dat20$varname<-rep("rev\_Mean",nrow(dat20))

#<21> Variable "ovrmou\_Mean"

summary(tele1$ovrmou\_Mean)

tele1%>%mutate(dec=ntile(ovrmou\_Mean,n=4))%>%count(churn,dec)%>%filter(churn==1)->dat21

dat21$N<-unclass(tele1%>%mutate(dec=ntile(ovrmou\_Mean,n=4))%>%count(dec)%>%unname())[[2]]

dat21$churn\_perc<-dat21$n/dat21$N

dat21$GreaterThan<-unclass(tele1%>%mutate(dec=ntile(ovrmou\_Mean,n=4))%>%group\_by(dec)%>%summarise(min(ovrmou\_Mean)))[[2]]

dat21$LessThan<-unclass(tele1%>%mutate(dec=ntile(ovrmou\_Mean,n=4))%>%group\_by(dec)%>%summarise(max(ovrmou\_Mean)))[[2]]

dat21$varname<-rep("ovrmou\_Mean",nrow(dat21))

# <22> Variable "comp\_vce\_Mean" ===>> \*\*\*\* Data Transformation then Delete \*\*\*\*

summary(tele1$comp\_vce\_Mean)

tele1%>%mutate(dec=ntile(comp\_vce\_Mean,n=10))%>%count(churn,dec)%>%filter(churn==1)->dat22

dat22$N<-unclass(tele1%>%mutate(dec=ntile(comp\_vce\_Mean,n=10))%>%count(dec)%>%unname())[[2]]

dat22$churn\_perc<-dat22$n/dat22$N

dat22$GreaterThan<-unclass(tele1%>%mutate(dec=ntile(comp\_vce\_Mean,n=10))%>%group\_by(dec)%>%summarise(min(comp\_vce\_Mean)))[[2]]

dat22$LessThan<-unclass(tele1%>%mutate(dec=ntile(comp\_vce\_Mean,n=10))%>%group\_by(dec)%>%summarise(max(comp\_vce\_Mean)))[[2]]

dat22$varname<-rep("comp\_vce\_Mean",nrow(dat22))

# <23> Variable "plcd\_vce\_Mean" ===>> \*\*\*\* Data Transformation then Delete \*\*\*\*

summary(tele1$plcd\_vce\_Mean)

tele1%>%mutate(dec=ntile(plcd\_vce\_Mean,n=10))%>%count(churn,dec)%>%filter(churn==1)->dat23

dat23$N<-unclass(tele1%>%mutate(dec=ntile(plcd\_vce\_Mean,n=10))%>%count(dec)%>%unname())[[2]]

dat23$churn\_perc<-dat23$n/dat23$N

dat23$GreaterThan<-unclass(tele1%>%mutate(dec=ntile(plcd\_vce\_Mean,n=10))%>%group\_by(dec)%>%summarise(min(plcd\_vce\_Mean)))[[2]]

dat23$LessThan<-unclass(tele1%>%mutate(dec=ntile(plcd\_vce\_Mean,n=10))%>%group\_by(dec)%>%summarise(max(plcd\_vce\_Mean)))[[2]]

dat23$varname<-rep("plcd\_vce\_Mean",nrow(dat23))

# <24> Variable "avg3mou"

summary(tele1$avg3mou)

tele1%>%mutate(dec=ntile(avg3mou,n=10))%>%count(churn,dec)%>%filter(churn==1)->dat24

dat24$N<-unclass(tele1%>%mutate(dec=ntile(avg3mou,n=10))%>%count(dec)%>%unname())[[2]]

dat24$churn\_perc<-dat24$n/dat24$N

dat24$GreaterThan<-unclass(tele1%>%mutate(dec=ntile(avg3mou,n=10))%>%group\_by(dec)%>%summarise(min(avg3mou)))[[2]]

dat24$LessThan<-unclass(tele1%>%mutate(dec=ntile(avg3mou,n=10))%>%group\_by(dec)%>%summarise(max(avg3mou)))[[2]]

dat24$varname<-rep("avg3mou",nrow(dat24))

# <25> Variable "avgmou"

summary(tele1$avgmou)

tele1%>%mutate(dec=ntile(avgmou,n=10))%>%count(churn,dec)%>%filter(churn==1)->dat25

dat25$N<-unclass(tele1%>%mutate(dec=ntile(avgmou,n=10))%>%count(dec)%>%unname())[[2]]

dat25$churn\_perc<-dat25$n/dat25$N

dat25$GreaterThan<-unclass(tele1%>%mutate(dec=ntile(avgmou,n=10))%>%group\_by(dec)%>%summarise(min(avgmou)))[[2]]

dat25$LessThan<-unclass(tele1%>%mutate(dec=ntile(avgmou,n=10))%>%group\_by(dec)%>%summarise(max(avgmou)))[[2]]

dat25$varname<-rep("avgmou",nrow(dat25))

# <26> Variable "avg3qty"

summary(tele1$avg3qty)

tele1%>%mutate(dec=ntile(avg3qty,n=10))%>%count(churn,dec)%>%filter(churn==1)->dat26

dat26$N<-unclass(tele1%>%mutate(dec=ntile(avg3qty,n=10))%>%count(dec)%>%unname())[[2]]

dat26$churn\_perc<-dat26$n/dat26$N

dat26$GreaterThan<-unclass(tele1%>%mutate(dec=ntile(avg3qty,n=10))%>%group\_by(dec)%>%summarise(min(avg3qty)))[[2]]

dat26$LessThan<-unclass(tele1%>%mutate(dec=ntile(avg3qty,n=10))%>%group\_by(dec)%>%summarise(max(avg3qty)))[[2]]

dat26$varname<-rep("avg3qty",nrow(dat26))

# <27> Variable "avgqty"

summary(tele1$avgqty)

tele1%>%mutate(dec=ntile(avgqty,n=10))%>%count(churn,dec)%>%filter(churn==1)->dat27

dat27$N<-unclass(tele1%>%mutate(dec=ntile(avgqty,n=10))%>%count(dec)%>%unname())[[2]]

dat27$churn\_perc<-dat27$n/dat27$N

dat27$GreaterThan<-unclass(tele1%>%mutate(dec=ntile(avgqty,n=10))%>%group\_by(dec)%>%summarise(min(avgqty)))[[2]]

dat27$LessThan<-unclass(tele1%>%mutate(dec=ntile(avgqty,n=10))%>%group\_by(dec)%>%summarise(max(avgqty)))[[2]]

dat27$varname<-rep("avgqty",nrow(dat27))

# <28> Variable "avg6mou"

summary(tele1$avg6mou)

tele1%>%mutate(dec=ntile(avg6mou,n=10))%>%count(churn,dec)%>%filter(churn==1)->dat28

dat28$N<-unclass(tele1%>%mutate(dec=ntile(avg6mou,n=10))%>%count(dec)%>%unname())[[2]]

dat28$churn\_perc<-dat28$n/dat28$N

dat28$GreaterThan<-unclass(tele1%>%mutate(dec=ntile(avg6mou,n=10))%>%group\_by(dec)%>%summarise(min(avg6mou)))[[2]]

dat28$LessThan<-unclass(tele1%>%mutate(dec=ntile(avg6mou,n=10))%>%group\_by(dec)%>%summarise(max(avg6mou)))[[2]]

dat28$varname<-rep("avg6mou",nrow(dat28))

# <29> Variable "avg6qty"

summary(tele1$avg6qty)

tele1%>%mutate(dec=ntile(avg6qty,n=10))%>%count(churn,dec)%>%filter(churn==1)->dat29

dat29$N<-unclass(tele1%>%mutate(dec=ntile(avg6qty,n=10))%>%count(dec)%>%unname())[[2]]

dat29$churn\_perc<-dat29$n/dat29$N

dat29$GreaterThan<-unclass(tele1%>%mutate(dec=ntile(avg6qty,n=10))%>%group\_by(dec)%>%summarise(min(avg6qty)))[[2]]

dat29$LessThan<-unclass(tele1%>%mutate(dec=ntile(avg6qty,n=10))%>%group\_by(dec)%>%summarise(max(avg6qty)))[[2]]

dat29$varname<-rep("avg6qty",nrow(dat29))

# <30> Variable "age1" =====>>\*\*\*\*Use As Factor\*\*\*\*

summary(tele1$age1)

tele1%>%mutate(dec=ntile(age1,n=6))%>%count(churn,dec)%>%filter(churn==1)->dat30

dat30$N<-unclass(tele1%>%mutate(dec=ntile(age1,n=6))%>%count(dec)%>%unname())[[2]]

dat30$churn\_perc<-dat30$n/dat30$N

dat30$GreaterThan<-unclass(tele1%>%mutate(dec=ntile(age1,n=6))%>%group\_by(dec)%>%summarise(min(age1)))[[2]]

dat30$LessThan<-unclass(tele1%>%mutate(dec=ntile(age1,n=6))%>%group\_by(dec)%>%summarise(max(age1)))[[2]]

dat30$varname<-rep("age1",nrow(dat30))

# <31> Variable "age2"===>> \*\*\*Getting less than 4 deciles. Use As Factor\*\*\*

summary(tele1$age2)

tele1%>%mutate(dec=ntile(age2,n=4))%>%count(churn,dec)%>%filter(churn==1)->dat31

dat31$varname<-rep("age2",nrow(dat31))

# <32> Variable "models" ===>> \*\*\*Getting less than 4 deciles. Factor Variable\*\*\*

summary(tele1$models)

tele1%>%mutate(dec=ntile(models,n=4))%>%count(churn,dec)%>%filter(churn==1)->dat32

dat32$varname<-rep("models",nrow(dat32))

# <33> Variable "hnd\_price" =====>> \*\*\*\*\*\* Use as Factor Variable\*\*\*\*\*

summary(tele1$hnd\_price)

tele1%>%mutate(dec=ntile(hnd\_price,n=10))%>%count(churn,dec)%>%filter(churn==1)->dat33

dat33$N<-unclass(tele1%>%mutate(dec=ntile(hnd\_price,n=10))%>%count(dec)%>%unname())[[2]]

dat33$churn\_perc<-dat33$n/dat33$N

dat33$GreaterThan<-unclass(tele1%>%mutate(dec=ntile(hnd\_price,n=10))%>%group\_by(dec)%>%summarise(min(hnd\_price)))[[2]]

dat33$LessThan<-unclass(tele1%>%mutate(dec=ntile(hnd\_price,n=10))%>%group\_by(dec)%>%summarise(max(hnd\_price)))[[2]]

dat33$varname<-rep("hnd\_price",nrow(dat33))

# <34> Variable "actvsubs" ===>> \*\*\*Factor Variable\*\*\*

summary(tele1$actvsubs)

tele1%>%mutate(dec=ntile(actvsubs,n=4))%>%count(churn,dec)%>%filter(churn==1)->dat34

dat34$varname<-rep("actvsubs",nrow(dat34))

# <35> Variable "uniqsubs" ===>> \*\*\*Factor Variable\*\*\*

summary(tele1$uniqsubs)

tele1%>%mutate(dec=ntile(uniqsubs,n=4))%>%count(churn,dec)%>%filter(churn==1)->dat35

dat35$varname<-rep("uniqsubs",nrow(dat35))

# <36> Variable "forgntvl" ===>> \*\*\*Factor Variable\*\*\*

summary(tele1$forgntvl)

tele1%>%mutate(dec=ntile(forgntvl,n=4))%>%count(churn,dec)%>%filter(churn==1)->dat36

dat36$varname<-rep("forgntvl",nrow(dat36))

# <37> Variable "opk\_dat\_Mean" ===>> \*\*\*Omit\*\*\*

summary(tele1$opk\_dat\_Mean)

tele1%>%mutate(dec=ntile(opk\_dat\_Mean,n=4))%>%count(churn,dec)%>%filter(churn==1)->dat37

dat37$varname<-rep("opk\_dat\_Mean",nrow(dat37))

# <38> Variable "mtrcycle" ===>> \*\*\*Factor variable\*\*\*

summary(tele1$mtrcycle)

tele1%>%mutate(dec=ntile(mtrcycle,n=4))%>%count(churn,dec)%>%filter(churn==1)->dat38

dat38$varname<-rep("mtrcycle",nrow(dat38))

# <39> Variable "truck" ===>> \*\*\*Factor variable\*\*\*

summary(tele1$truck)

tele1%>%mutate(dec=ntile(truck,n=4))%>%count(churn,dec)%>%filter(churn==1)->dat39

dat39$varname<-rep("truck",nrow(dat39))

# <40> Variable "roam\_Mean" ===>> \*\*\*Getting less than 4 deciles. So Omit\*\*\*

summary(tele1$roam\_Mean)

tele1%>%mutate(dec=ntile(roam\_Mean,n=4))%>%count(churn,dec)%>%filter(churn==1)->dat40

dat40$varname<-rep("roam\_Mean",nrow(dat40))

# <41> Variable "recv\_sms\_Mean" ===>> \*\*\*Getting less than 4 deciles. So Omit\*\*\*

summary(tele1$recv\_sms\_Mean)

tele1%>%mutate(dec=ntile(recv\_sms\_Mean,n=4))%>%count(churn,dec)%>%filter(churn==1)->dat41

dat41$varname<-rep("recv\_sms\_Mean",nrow(dat41))

# <42> Variable "mou\_pead\_Mean" ===>> \*\*\*Getting less than 4 deciles. So Omit\*\*\*

summary(tele1$mou\_pead\_Mean)

tele1%>%mutate(dec=ntile(mou\_pead\_Mean,n=4))%>%count(churn,dec)%>%filter(churn==1)->dat42

dat42$varname<-rep("mou\_pead\_Mean",nrow(dat42))

# <43> Variable "da\_Mean"

summary(tele1$da\_Mean)

tele1%>%mutate(dec=ntile(da\_Mean,n=4))%>%count(churn,dec)%>%filter(churn==1)->dat43

dat43$N<-unclass(tele1%>%mutate(dec=ntile(da\_Mean,n=4))%>%count(dec)%>%unname())[[2]]

dat43$churn\_perc<-dat43$n/dat43$N

dat43$GreaterThan<-unclass(tele1%>%mutate(dec=ntile(da\_Mean,n=4))%>%group\_by(dec)%>%summarise(min(da\_Mean)))[[2]]

dat43$LessThan<-unclass(tele1%>%mutate(dec=ntile(da\_Mean,n=4))%>%group\_by(dec)%>%summarise(max(da\_Mean)))[[2]]

dat43$varname<-rep("da\_Mean",nrow(dat43))

# <44> Variable "da\_Range"

summary(tele1$da\_Range)

tele1%>%mutate(dec=ntile(da\_Range,n=4))%>%count(churn,dec)%>%filter(churn==1)->dat44

dat44$N<-unclass(tele1%>%mutate(dec=ntile(da\_Range,n=4))%>%count(dec)%>%unname())[[2]]

dat44$churn\_perc<-dat44$n/dat44$N

dat44$GreaterThan<-unclass(tele1%>%mutate(dec=ntile(da\_Range,n=4))%>%group\_by(dec)%>%summarise(min(da\_Range)))[[2]]

dat44$LessThan<-unclass(tele1%>%mutate(dec=ntile(da\_Range,n=4))%>%group\_by(dec)%>%summarise(max(da\_Range)))[[2]]

dat44$varname<-rep("da\_Range",nrow(dat44))

# <45> Variable "datovr\_Mean" ===>> \*\*\*Getting less than 4 deciles. So Omit\*\*\*

summary(tele1$datovr\_Mean)

tele1%>%mutate(dec=ntile(datovr\_Mean,n=4))%>%count(churn,dec)%>%filter(churn==1)->dat45

dat45$varname<-rep("datovr\_Mean",nrow(dat45))

# <46> Variable "datovr\_Range" ===>> \*\*\*Getting less than 4 deciles. So Omit\*\*\*

summary(tele1$datovr\_Range)

tele1%>%mutate(dec=ntile(datovr\_Range,n=4))%>%count(churn,dec)%>%filter(churn==1)->dat46

dat46$varname<-rep("datovr\_Range",nrow(dat46))

# <47> Variable "drop\_dat\_Mean" ===>> \*\*\*Getting less than 4 deciles.

#And almost 95% data has values 0. So Omit\*\*\*

summary(tele1$drop\_dat\_Mean)

tele1%>%mutate(dec=ntile(drop\_dat\_Mean,n=4))%>%count(churn,dec)%>%filter(churn==1)->dat47

dat47$varname<-rep("drop\_dat\_Mean",nrow(dat47))

# <48> Variable "drop\_vce\_Mean"

summary(tele1$drop\_vce\_Mean)

tele1%>%mutate(dec=ntile(drop\_vce\_Mean,n=10))%>%count(churn,dec)%>%filter(churn==1)->dat48

dat48$N<-unclass(tele1%>%mutate(dec=ntile(drop\_vce\_Mean,n=10))%>%count(dec)%>%unname())[[2]]

dat48$churn\_perc<-dat48$n/dat48$N

dat48$GreaterThan<-unclass(tele1%>%mutate(dec=ntile(drop\_vce\_Mean,n=10))%>%group\_by(dec)%>%summarise(min(drop\_vce\_Mean)))[[2]]

dat48$LessThan<-unclass(tele1%>%mutate(dec=ntile(drop\_vce\_Mean,n=10))%>%group\_by(dec)%>%summarise(max(drop\_vce\_Mean)))[[2]]

dat48$varname<-rep("drop\_vce\_Mean",nrow(dat48))

# <49> Variable "adjmou"

summary(tele1$adjmou)

tele1%>%mutate(dec=ntile(adjmou,n=10))%>%count(churn,dec)%>%filter(churn==1)->dat49

dat49$N<-unclass(tele1%>%mutate(dec=ntile(adjmou,n=10))%>%count(dec)%>%unname())[[2]]

dat49$churn\_perc<-dat49$n/dat49$N

dat49$GreaterThan<-unclass(tele1%>%mutate(dec=ntile(adjmou,n=10))%>%group\_by(dec)%>%summarise(min(adjmou)))[[2]]

dat49$LessThan<-unclass(tele1%>%mutate(dec=ntile(adjmou,n=10))%>%group\_by(dec)%>%summarise(max(adjmou)))[[2]]

dat49$varname<-rep("adjmou",nrow(dat49))

# <50> Variable "totrev"

summary(tele1$totrev)

tele1%>%mutate(dec=ntile(totrev,n=10))%>%count(churn,dec)%>%filter(churn==1)->dat50

dat50$N<-unclass(tele1%>%mutate(dec=ntile(totrev,n=10))%>%count(dec)%>%unname())[[2]]

dat50$churn\_perc<-dat50$n/dat50$N

dat50$GreaterThan<-unclass(tele1%>%mutate(dec=ntile(totrev,n=10))%>%group\_by(dec)%>%summarise(min(totrev)))[[2]]

dat50$LessThan<-unclass(tele1%>%mutate(dec=ntile(totrev,n=10))%>%group\_by(dec)%>%summarise(max(totrev)))[[2]]

dat50$varname<-rep("totrev",nrow(dat50))

# <51> Variable "adjrev"

summary(tele1$adjrev)

tele1%>%mutate(dec=ntile(adjrev,n=10))%>%count(churn,dec)%>%filter(churn==1)->dat51

dat51$N<-unclass(tele1%>%mutate(dec=ntile(adjrev,n=10))%>%count(dec)%>%unname())[[2]]

dat51$churn\_perc<-dat51$n/dat51$N

dat51$GreaterThan<-unclass(tele1%>%mutate(dec=ntile(adjrev,n=10))%>%group\_by(dec)%>%summarise(min(adjrev)))[[2]]

dat51$LessThan<-unclass(tele1%>%mutate(dec=ntile(adjrev,n=10))%>%group\_by(dec)%>%summarise(max(adjrev)))[[2]]

dat51$varname<-rep("adjrev",nrow(dat51))

# <52> Variable "avgrev"

summary(tele1$avgrev)

tele1%>%mutate(dec=ntile(avgrev,n=10))%>%count(churn,dec)%>%filter(churn==1)->dat52

dat52$N<-unclass(tele1%>%mutate(dec=ntile(avgrev,n=10))%>%count(dec)%>%unname())[[2]]

dat52$churn\_perc<-dat52$n/dat52$N

dat52$GreaterThan<-unclass(tele1%>%mutate(dec=ntile(avgrev,n=10))%>%group\_by(dec)%>%summarise(min(avgrev)))[[2]]

dat52$LessThan<-unclass(tele1%>%mutate(dec=ntile(avgrev,n=10))%>%group\_by(dec)%>%summarise(max(avgrev)))[[2]]

dat52$varname<-rep("avgrev",nrow(dat52))

## \*\*\*\* Data Transformation. Creating Dummy Variables \*\*\*\* ##

#<55> Create Dummy Variable plcd\_Atempt\_Mean and Deciling

tele1$plcd\_attempt\_Mean<-tele1$plcd\_vce\_Mean+tele1$plcd\_dat\_Mean

summary(tele1$plcd\_attempt\_Mean)

tele1%>%mutate(dec=ntile(plcd\_attempt\_Mean,n=10))%>%count(churn,dec)%>%filter(churn==1)->dat55

dat55$N<-unclass(tele1%>%mutate(dec=ntile(plcd\_attempt\_Mean,n=10))%>%count(dec)%>%unname())[[2]]

dat55$churn\_perc<-dat55$n/dat55$N

dat55$GreaterThan<-unclass(tele1%>%mutate(dec=ntile(plcd\_attempt\_Mean,n=10))%>%group\_by(dec)%>%summarise(min(plcd\_attempt\_Mean)))[[2]]

dat55$LessThan<-unclass(tele1%>%mutate(dec=ntile(plcd\_attempt\_Mean,n=10))%>%group\_by(dec)%>%summarise(max(plcd\_attempt\_Mean)))[[2]]

dat55$varname<-rep("plcd\_attempt\_Mean",nrow(dat55))

#<56> Create Dummy Variable complete\_Mean and Deciling

tele1$complete\_Mean<-tele1$comp\_vce\_Mean+tele1$comp\_dat\_Mean

summary(tele1$complete\_Mean)

tele1%>%mutate(dec=ntile(complete\_Mean,n=10))%>%count(churn,dec)%>%filter(churn==1)->dat56

dat56$N<-unclass(tele1%>%mutate(dec=ntile(complete\_Mean,n=10))%>%count(dec)%>%unname())[[2]]

dat56$churn\_perc<-dat56$n/dat56$N

dat56$GreaterThan<-unclass(tele1%>%mutate(dec=ntile(complete\_Mean,n=10))%>%group\_by(dec)%>%summarise(min(complete\_Mean)))[[2]]

dat56$LessThan<-unclass(tele1%>%mutate(dec=ntile(complete\_Mean,n=10))%>%group\_by(dec)%>%summarise(max(complete\_Mean)))[[2]]

dat56$varname<-rep("complete\_Mean",nrow(dat56))

#Adding all appropriate dat1 to dat54 objects to create a dat object

dat<-rbind(dat1,dat2,dat4,dat5,dat6,dat7,dat8,dat9,dat10,dat11,dat12,dat14,dat15,dat18,dat19,

dat20,dat21,dat22,dat23,dat24,dat25,dat26,dat27,dat28,dat29,dat30,dat33,dat43,dat44,

dat48,dat49,dat50,dat51,dat52,dat55,dat56)

#Exporting Deciled variables

write.csv(dat,"Deciled Usable Continuous variables.csv",row.names = F)

#Removing Variables that could not be deciled as will come insignificant in the model

#Also omitting transformed Vars "comp\_vce\_Mean" ,"comp\_dat\_Mean", "plcd\_vce\_Mean", plcd\_dat\_Mean

names(tele1)

tele1<-tele1[,-c(13,16,17,22,23,45,48:50,56:58,65,66)]

names(tele1)

##-------- Categorical Variables---------##

##-----Event rate for each level in a categorical variable-----##

# <19> Variable "crclscod" =====>>> \*\*\*\* Some Levels show less than 5% churn rate. So Omit as will come insignificant \*\*\*\*

summary(tele1$crclscod)

tele1%>%count(churn,levels=crclscod)%>%filter(churn==1)->datC19

datC19$N<-unclass(tele1%>%filter(crclscod%in%datC19$levels)%>%count(crclscod))[[2]]

datC19$ChurnPerc<-datC19$n/datC19$N

datC19$Var.Name<-rep("crclscod",nrow(datC19))

# <20> Variable "asl\_flag"

summary(tele1$asl\_flag)

tele1%>%count(churn,levels=asl\_flag)%>%filter(churn==1)->datC20

datC20$N<-unclass(tele1%>%filter(asl\_flag%in%datC20$levels)%>%count(asl\_flag))[[2]]

datC20$ChurnPerc<-datC20$n/datC20$N

datC20$Var.Name<-rep("asl\_flag",nrow(datC20))

# <21> Variable "prizm\_social\_one"

summary(tele1$prizm\_social\_one)

tele1%>%count(churn,levels=prizm\_social\_one)%>%filter(churn==1)->datC21

datC21$N<-unclass(tele1%>%filter(prizm\_social\_one%in%datC21$levels)%>%count(prizm\_social\_one))[[2]]

datC21$ChurnPerc<-datC21$n/datC21$N

datC21$Var.Name<-rep("prizm\_social\_one",nrow(datC21))

# <22> Variable "area"

summary(tele1$area)

tele1%>%count(churn,levels=area)%>%filter(churn==1)->datC22

datC22$N<-unclass(tele1%>%filter(area%in%datC22$levels)%>%count(area))[[2]]

datC22$ChurnPerc<-datC22$n/datC22$N

datC22$Var.Name<-rep("area",nrow(datC22))

# <23> Variable "refurb\_new"

summary(tele1$refurb\_new)

tele1%>%count(churn,levels=refurb\_new)%>%filter(churn==1)->datC23

datC23$N<-unclass(tele1%>%filter(refurb\_new%in%datC23$levels)%>%count(refurb\_new))[[2]]

datC23$ChurnPerc<-datC23$n/datC23$N

datC23$Var.Name<-rep("refurb\_new",nrow(datC23))

# <24> Variable "hnd\_webcap"

summary(tele1$hnd\_webcap)

tele1%>%count(churn,levels=hnd\_webcap)%>%filter(churn==1)->datC24

datC24$N<-unclass(tele1%>%filter(hnd\_webcap%in%datC24$levels)%>%count(hnd\_webcap))[[2]]

datC24$ChurnPerc<-datC24$n/datC24$N

datC24$Var.Name<-rep("hnd\_webcap",nrow(datC24))

# <25> Variable "marital"

summary(tele1$marital)

tele1%>%count(churn,levels=marital)%>%filter(churn==1)->datC25

datC25$N<-unclass(tele1%>%filter(marital%in%datC25$levels)%>%count(marital))[[2]]

datC25$ChurnPerc<-datC25$n/datC25$N

datC25$Var.Name<-rep("marital",nrow(datC25))

# <26> Variable "ethnic"

summary(tele1$ethnic)

tele1%>%count(churn,levels=ethnic)%>%filter(churn==1)->datC26

datC26$N<-unclass(tele1%>%filter(ethnic%in%datC26$levels)%>%count(ethnic))[[2]]

datC26$ChurnPerc<-datC26$n/datC26$N

datC26$Var.Name<-rep("ethnic",nrow(datC26))

# <27> Variable "car\_buy"

summary(tele1$car\_buy)

tele1%>%count(churn,levels=car\_buy)%>%filter(churn==1)->datC27

datC27$N<-unclass(tele1%>%filter(car\_buy%in%datC27$levels)%>%count(car\_buy))[[2]]

datC27$ChurnPerc<-datC27$n/datC27$N

datC27$Var.Name<-rep("car\_buy",nrow(datC27))

# <28> Variable "csa" ===>>> \*\*\*\* Some Levels show less than 5% churn rate. So Omit as will come insignificant \*\*\*\*

summary(tele1$csa)

tele1%>%count(churn,levels=csa)%>%filter(churn==1)->datC28

datC28$N<-unclass(tele1%>%filter(csa%in%datC28$levels)%>%count(csa))[[2]]

datC28$ChurnPerc<-datC28$n/datC28$N

datC28$Var.Name<-rep("csa",nrow(datC28))

# <29> Variable "retdays\_1"

summary(tele1$retdays\_1)

tele1$retdays\_1<-as.factor(tele1$retdays\_1)

tele1%>%count(churn,levels=retdays\_1)%>%filter(churn==1)->datC29

datC29$N<-unclass(tele1%>%filter(retdays\_1%in%datC29$levels)%>%count(retdays\_1))[[2]]

datC29$ChurnPerc<-datC29$n/datC29$N

datC29$Var.Name<-rep("retdays\_1",nrow(datC29))

# Use VArs as Factor => age2, models, actvsubs, uniqsubs, forgntvl, mtrcycle, truck,

# <31> Variable "age2"

summary(tele1$age2)

tele1%>%count(churn,levels=age2)%>%filter(churn==1)->datC31

datC31$N<-unclass(tele1%>%filter(age2%in%datC31$levels)%>%count(age2))[[2]]

datC31$ChurnPerc<-datC31$n/datC31$N

datC31$Var.Name<-rep("age2",nrow(datC31))

# <32> Variable "models"

summary(tele1$models)

tele1%>%count(churn,levels=models)%>%filter(churn==1)->datC32

datC32$N<-unclass(tele1%>%filter(models%in%datC32$levels)%>%count(models))[[2]]

datC32$ChurnPerc<-datC32$n/datC32$N

datC32$Var.Name<-rep("models",nrow(datC32))

# <34> Variable "actvsubs"

summary(tele1$actvsubs)

tele1%>%count(churn,levels=actvsubs)%>%filter(churn==1)->datC34

datC34$N<-unclass(tele1%>%filter(actvsubs%in%datC34$levels)%>%count(actvsubs))[[2]]

datC34$ChurnPerc<-datC34$n/datC34$N

datC34$Var.Name<-rep("actvsubs",nrow(datC34))

# <35> Variable "uniqsubs"

summary(tele1$uniqsubs)

tele1%>%count(churn,levels=uniqsubs)%>%filter(churn==1)->datC35

datC35$N<-unclass(tele1%>%filter(uniqsubs%in%datC35$levels)%>%count(uniqsubs))[[2]]

datC35$ChurnPerc<-datC35$n/datC35$N

datC35$Var.Name<-rep("uniqsubs",nrow(datC35))

# <36> Variable "forgntvl"

summary(tele1$forgntvl)

tele1%>%count(churn,levels=forgntvl)%>%filter(churn==1)->datC36

datC36$N<-unclass(tele1%>%filter(forgntvl%in%datC36$levels)%>%count(forgntvl))[[2]]

datC36$ChurnPerc<-datC36$n/datC36$N

datC36$Var.Name<-rep("forgntvl",nrow(datC36))

# <37> Variable "mtrcycle"

summary(tele1$mtrcycle)

tele1%>%count(churn,levels=mtrcycle)%>%filter(churn==1)->datC37

datC37$N<-unclass(tele1%>%filter(mtrcycle%in%datC37$levels)%>%count(mtrcycle))[[2]]

datC37$ChurnPerc<-datC37$n/datC37$N

datC37$Var.Name<-rep("mtrcycle",nrow(datC37))

# <38> Variable "Truck"

summary(tele1$truck)

tele1%>%count(churn,levels=truck)%>%filter(churn==1)->datC38

datC38$N<-unclass(tele1%>%filter(truck%in%datC38$levels)%>%count(truck))[[2]]

datC38$ChurnPerc<-datC38$n/datC38$N

datC38$Var.Name<-rep("Truck",nrow(datC38))

# <39> Variable "hnd\_price"

summary(tele1$hnd\_price)

tele1%>%count(churn,levels=hnd\_price)%>%filter(churn==1)->datC39

datC39$N<-unclass(tele1%>%filter(hnd\_price%in%datC39$levels)%>%count(hnd\_price))[[2]]

datC39$ChurnPerc<-datC39$n/datC39$N

datC39$Var.Name<-rep("hnd\_price",nrow(datC39))

datC39

#Adding datC19 to datC38 objects to create a datC object

datC\_1<-rbind(datC19,datC20,datC21,datC22,datC23,datC24,datC25,datC26,datC27,

datC28,datC29)

datC\_2<-rbind(datC31,datC32,datC34,datC35,datC36,datC37,datC38)

#Exporting Deciled variables

write.csv(datC\_1,"Event Rate - Categorical variables1.csv",row.names = F)

write.csv(datC\_2,"Event Rate - Categorical variables2.csv",row.names = F)

#Removing Variables with levels less than 5% churn rate as will come insignificant

names(tele1)

tele1<-tele1[,-c(25,44)]

names(tele1)

#\*\*\*\*\*\*\*\*\*Data Preparation\*\*\*\*\*\*\*\*\*#

#-----Outlier Treatment----#

#-----Continuous Variables------#

#Method <II> ----- Box Plot Method==>> Used

names(tele1)

summary(tele1)

str(tele1)

#Factor Variables=> asl\_flag, prizm\_social\_one, area, refurb\_new, hnd\_webcap,

#marital, ethnic, age1, age2, models, hnd\_price, actvsubs, "uniqsubs",

#"forgntvl", "mtrcycle", "truck", churn, car\_buy, Customer\_ID, retdays\_1

str(tele1)

list<-names(tele1)

list

# Removing Categorical Variables

list<-list[-c(25:42,50,51)]

list

# Outlier Plots

par(mfrow=c(3,11))

for(i in 1:length(list))

{

boxplot(tele1[,list[i]],main=list[i])

}

for(i in 1:length(list))

{

plot(tele1[,list[i]],main=list[i])

}

# Outlier Treatment

for(i in 1:length(list))

{

x<-boxplot(tele1[,list[i]],main=list[i])

out<-x$out

index<-which(tele1[,list[i]]%in% x$out)

tele1[index,list[i]]<-mean(tele1[,list[i]],na.rm = T)

rm(x)

rm(out)

}

# Checking After Treatment

for(i in 1:length(list))

{

boxplot(tele1[,list[i]],main=list[i])

}

for(i in 1:length(list))

{

plot(tele1[,list[i]],main=list[i])

}

dev.off()

#-----Missing Value Treatment -------#

summary(tele1)

names(tele1)

# Factor Variables=> crclscod, asl\_flag, prizm\_social\_one, area, refurb\_new, hnd\_webcap, marital, ethnic, "age1",

# "age2", "models", "hnd\_price","actvsubs", "uniqsubs", "forgntvl", "mtrcycle", "truck", car\_buy, csa retdays\_1

# Deleting Missing Values

index1<-which(is.na(tele1[,c(1:5)]))

tele1<-tele1[-index1,]

summary(tele1)

index2<-which(is.na(tele1$change\_mou))

tele1<-tele1[-index2,]

index4<-which(is.na(tele1$area))

tele1<-tele1[-index4,]

index5<-which(is.na(tele1$marital))

tele1<-tele1[-index5,]

summary(tele1)

# Mean Imputation

tele1$avg6mou[is.na(tele1$avg6mou)]<-mean(tele1$avg6mou,na.rm = T)

tele1$avg6qty[is.na(tele1$avg6qty)]<-mean(tele1$avg6qty,na.rm = T)

tele1$hnd\_price[is.na(tele1$hnd\_price)]<-mean(tele1$hnd\_price,na.rm = T)

summary(tele1)

# Creating seperate category "Missing" for Factor Variables

#Variable prizm\_social\_one

tele1$prizm\_social\_one\_1<-ifelse(is.na(tele1$prizm\_social\_one),"Missing",as.factor(tele1$prizm\_social\_one))

str(tele1$prizm\_social\_one\_1)

tele1$prizm\_social\_one\_1<-as.factor(tele1$prizm\_social\_one\_1)

summary(tele1$prizm\_social\_one)

summary(tele1$prizm\_social\_one\_1)

tele1$prizm\_social\_one\_1<-factor(tele1$prizm\_social\_one\_1,labels =c("C","R","S","T","U","Missing"))

summary(tele1$prizm\_social\_one\_1)

names(tele1)

tele1<-tele1[,-26]

summary(tele1)

#Variable hnd\_webcap

tele1$hnd\_webcap\_1<-ifelse(is.na(tele1$hnd\_webcap),"Missing",as.factor(tele1$hnd\_webcap))

str(tele1$hnd\_webcap\_1)

tele1$hnd\_webcap\_1<-as.factor(tele1$hnd\_webcap\_1)

summary(tele1$hnd\_webcap)

summary(tele1$hnd\_webcap\_1)

tele1$hnd\_webcap\_1<-factor(tele1$hnd\_webcap\_1,labels =c("UNKW","WC","WCMB","Missing"))

summary(tele1$hnd\_webcap\_1)

names(tele1)

tele1<-tele1[,-28]

summary(tele1)

#Checking Churn Rate in the data after Imputations

table(tele$churn)/nrow(tele)

table(tele1$churn)/nrow(tele1)

# Convert to Factor and Create Dummy Variables =>

#age1, age2, models, hnd\_price, actvsubs,uniqsubs, forgntvl, mtrcycle, truck, Customer ID, Churn

str(tele1$age1)

tele1$age1\_1<-ifelse(tele1$age1==0,"Default",ifelse(tele1$age1<=30,"Young",

ifelse(tele1$age1>30 & tele1$age1<=55,"Mid Age","Old")))

str(tele1$age1\_1)

tele1$age1\_1<-as.factor(tele1$age1\_1)

summary(tele1$age1\_1)

names(tele1)

tele1<-tele1[,-30]

summary(tele1)

str(tele1$age2)

tele1$age2\_1<-ifelse(tele1$age2==0,"Default",ifelse(tele1$age2<=30,"Young",

ifelse(tele1$age2>30 & tele1$age2<=55,"Mid Age","Old")))

str(tele1$age2\_1)

tele1$age2\_1<-as.factor(tele1$age2\_1)

summary(tele1$age2\_1)

names(tele1)

tele1<-tele1[,-30]

summary(tele1)

str(tele1$models)

summary(tele1$models)

tele1$models<-as.factor(tele1$models)

summary(tele1$models)

str(tele1$hnd\_price)

summary(tele1$hnd\_price)

tele1$hnd\_price<-as.factor(tele1$hnd\_price)

summary(tele1$hnd\_price)

str(tele1$actvsubs)

summary(tele1$actvsubs)

tele1$actvsubs<-as.factor(tele1$actvsubs)

summary(tele1$actvsubs)

str(tele1$uniqsubs)

summary(tele1$uniqsubs)

tele1$uniqsubs<-as.factor(tele1$uniqsubs)

summary(tele1$uniqsubs)

str(tele1$forgntvl)

summary(tele1$forgntvl)

tele1$forgntvl<-as.factor(tele1$forgntvl)

summary(tele1$forgntvl)

str(tele1$mtrcycle)

summary(tele1$mtrcycle)

tele1$mtrcycle<-as.factor(tele1$mtrcycle)

summary(tele1$mtrcycle)

str(tele1$truck)

summary(tele1$truck)

tele1$truck<-as.factor(tele1$truck)

summary(tele1$truck)

### \*\*\*\*\*\*\*\*\*\* Logistic Regression Model Building \*\*\*\*\*\*\*\*\*\* ###

# Splitting into Test and Training Samples

set.seed(200)

index<-sample(nrow(tele1),0.70\*nrow(tele1),replace=F)

train<-tele1[index,]

test<-tele1[-index,]

#Checking Churn Rate

table(train$churn)/nrow(train)

table(test$churn)/nrow(test)

names(tele1)

# Building Logistic Regression Model after excluding var "Customer\_ID"

#debug\_contr\_error

mod<-glm(churn~.,data=train[,-40],family="binomial")

summary(mod)

# Step wise Regression Model ===>> Each Step Takes Atleat 1/2 Hour and the system shuts down in 3 hrs,

# before the process completes...Also showing extensive memory issues.

# So cannot use the method. Doing Manually..

step(mod,direction = "both")

## \*\*\*\*\* Creating Dummy Vars for Factor Vars with significant levels \*\*\*\*\* ##

summary(tele1$asl\_flag)

train$asl\_flag\_Y<-ifelse(train$asl\_flag == "Y", 1, 0)

test$asl\_flag\_Y<-ifelse(test$asl\_flag == "Y", 1, 0)

summary(train$area)

train$area\_Cal\_Nrth<-ifelse(train$area == "CALIFORNIA NORTH AREA", 1, 0)

test$area\_Cal\_Nrth<-ifelse(test$area == "CALIFORNIA NORTH AREA", 1, 0)

train$area\_texas<-ifelse(train$area == "CENTRAL/SOUTH TEXAS AREA", 1, 0)

test$area\_texas<-ifelse(test$area == "CENTRAL/SOUTH TEXAS AREA", 1, 0)

train$area\_nrthflrda<-ifelse(train$area == "NORTH FLORIDA AREA", 1, 0)

test$area\_nrthflrda<-ifelse(test$area == "NORTH FLORIDA AREA", 1, 0)

train$area\_nrthwst<-ifelse(train$area == "NORTHWEST/ROCKY MOUNTAIN AREA", 1, 0)

test$area\_nrthwst<-ifelse(test$area == "NORTHWEST/ROCKY MOUNTAIN AREA", 1, 0)

train$area\_southflrda<-ifelse(train$area == "SOUTH FLORIDA AREA", 1, 0)

test$area\_southflrda<-ifelse(test$area == "SOUTH FLORIDA AREA", 1, 0)

train$area\_southwst<-ifelse(train$area == "SOUTHWEST AREA", 1, 0)

test$area\_southwst<-ifelse(test$area == "SOUTHWEST AREA", 1, 0)

train$area\_tenese<-ifelse(train$area == "TENNESSEE AREA", 1, 0)

test$area\_tenese<-ifelse(test$area == "TENNESSEE AREA", 1, 0)

summary(train$refurb\_new)

train$refurb\_R<-ifelse(train$refurb\_new == "R", 1, 0)

test$refurb\_R<-ifelse(test$refurb\_new == "R", 1, 0)

summary(train$ethnic)

train$ethnic\_C<-ifelse(train$ethnic == "C", 1, 0)

test$ethnic\_C<-ifelse(test$ethnic == "C", 1, 0)

train$ethnic\_N<-ifelse(train$ethnic == "N", 1, 0)

test$ethnic\_N<-ifelse(test$ethnic == "N", 1, 0)

train$ethnic\_O<-ifelse(train$ethnic == "O", 1, 0)

test$ethnic\_O<-ifelse(test$ethnic == "O", 1, 0)

train$ethnic\_S<-ifelse(train$ethnic == "S", 1, 0)

test$ethnic\_S<-ifelse(test$ethnic == "S", 1, 0)

train$ethnic\_U<-ifelse(train$ethnic == "U", 1, 0)

test$ethnic\_U<-ifelse(test$ethnic == "U", 1, 0)

train$ethnic\_Z<-ifelse(train$ethnic == "Z", 1, 0)

test$ethnic\_Z<-ifelse(test$ethnic == "Z", 1, 0)

summary(train$hnd\_price)

train$hnd\_price\_79.98<-ifelse(train$hnd\_price == "79.98999023", 1, 0)

test$hnd\_price\_79.98<-ifelse(test$hnd\_price == "79.98999023", 1, 0)

train$hnd\_price\_105.08<-ifelse(train$hnd\_price == "105.083038078331", 1, 0)

test$hnd\_price\_105.08<-ifelse(test$hnd\_price == "105.083038078331", 1, 0)

train$hnd\_price\_129.98<-ifelse(train$hnd\_price == "129.9899902", 1, 0)

test$hnd\_price\_129.98<-ifelse(test$hnd\_price == "129.9899902", 1, 0)

train$hnd\_price\_149.98<-ifelse(train$hnd\_price == "149.9899902", 1, 0)

test$hnd\_price\_149.98<-ifelse(test$hnd\_price == "149.9899902", 1, 0)

train$hnd\_price\_199.98<-ifelse(train$hnd\_price == "199.9899902", 1, 0)

test$hnd\_price\_199.98<-ifelse(test$hnd\_price == "199.9899902", 1, 0)

train$hnd\_price\_249.98<-ifelse(train$hnd\_price == "249.9899902", 1, 0)

test$hnd\_price\_249.98<-ifelse(test$hnd\_price == "249.9899902", 1, 0)

summary(train$uniqsubs)

train$unq\_2<-ifelse(train$uniqsubs == "2", 1, 0)

test$unq\_2<-ifelse(test$uniqsubs == "2", 1, 0)

train$unq\_3<-ifelse(train$uniqsubs == "3", 1, 0)

test$unq\_3<-ifelse(test$uniqsubs == "3", 1, 0)

train$unq\_4<-ifelse(train$uniqsubs == "4", 1, 0)

test$unq\_4<-ifelse(test$uniqsubs == "4", 1, 0)

train$unq\_5<-ifelse(train$uniqsubs == "5", 1, 0)

test$unq\_5<-ifelse(test$uniqsubs == "5", 1, 0)

train$unq\_6<-ifelse(train$uniqsubs == "6", 1, 0)

test$unq\_6<-ifelse(test$uniqsubs == "6", 1, 0)

train$unq\_7<-ifelse(train$uniqsubs == "7", 1, 0)

test$unq\_7<-ifelse(test$uniqsubs == "7", 1, 0)

train$unq\_9<-ifelse(train$uniqsubs == "9", 1, 0)

test$unq\_9<-ifelse(test$uniqsubs == "9", 1, 0)

summary(train$truck)

summary(train$prizm\_social\_one\_1)

train$przm\_social\_R<-ifelse(train$prizm\_social\_one\_1 == "R", 1, 0)

test$przm\_social\_R<-ifelse(test$prizm\_social\_one\_1 == "R", 1, 0)

train$przm\_social\_T<-ifelse(train$prizm\_social\_one\_1 == "T", 1, 0)

test$przm\_social\_T<-ifelse(test$prizm\_social\_one\_1 == "T", 1, 0)

summary(train$age1\_1)

train$age1\_Mid\_Age<-ifelse(train$age1\_1 == "Mid Age", 1, 0)

test$age1\_Mid\_Age<-ifelse(test$age1\_1 == "Mid Age", 1, 0)

train$age1\_Old<-ifelse(train$age1\_1 == "Old", 1, 0)

test$age1\_Old<-ifelse(test$age1\_1 == "Old", 1, 0)

train$age1\_Young<-ifelse(train$age1\_1 == "Young", 1, 0) # Not Required

test$age1\_Young<-ifelse(test$age1\_1 == "Young", 1, 0)

summary(train$age2\_1)

train$age2\_Old<-ifelse(train$age2\_1 == "Old", 1, 0)

test$age2\_Old<-ifelse(test$age2\_1 == "Old", 1, 0)

## \*\*\*\* Rerunning Model with Significant Variables \*\*\*\*\* ##

names(train)

mod1<-glm(churn ~ mou\_Mean + totmrc\_Mean + rev\_Range + mou\_Range + change\_mou + drop\_blk\_Mean + drop\_vce\_Range +

mou\_opkv\_Range + months + eqpdays + iwylis\_vce\_Mean + ovrrev\_Mean + avgmou + avg3qty + avgqty +

avg6mou + asl\_flag\_Y + area\_Cal\_Nrth + area\_texas + area\_nrthflrda + area\_nrthwst + area\_southflrda +

area\_southwst + area\_tenese + refurb\_R + ethnic\_C + ethnic\_N + ethnic\_O + ethnic\_S + ethnic\_U +

ethnic\_Z + hnd\_price\_79.98 + hnd\_price\_105.08 + hnd\_price\_129.98 + hnd\_price\_149.98 +

hnd\_price\_199.98 + hnd\_price\_249.98 + unq\_2 + unq\_3 + unq\_4 + unq\_5 + unq\_6 + unq\_7 + unq\_9 +

truck + adjmou + totrev + retdays\_1 + complete\_Mean + przm\_social\_R + przm\_social\_T + age1\_Mid\_Age +

age1\_Old + age1\_Young + age2\_Old,data=train,family="binomial")

summary(mod1)

## \*\*\*\* Further Rerunning Model with Significant Variables \*\*\*\*\* ##

mod2<-glm(churn ~ mou\_Mean + totmrc\_Mean + rev\_Range + mou\_Range + change\_mou + drop\_blk\_Mean + drop\_vce\_Range +

mou\_opkv\_Range + months + eqpdays + iwylis\_vce\_Mean + ovrrev\_Mean + avgmou + avg3qty + avgqty +

avg6mou + asl\_flag\_Y + area\_Cal\_Nrth + area\_texas + area\_nrthflrda + area\_nrthwst + area\_southflrda +

area\_southwst + area\_tenese + refurb\_R + ethnic\_C + ethnic\_N + ethnic\_O + ethnic\_S + ethnic\_U +

ethnic\_Z + hnd\_price\_79.98 + hnd\_price\_105.08 + hnd\_price\_129.98 + hnd\_price\_149.98 + hnd\_price\_199.98 +

hnd\_price\_249.98 + unq\_2 + unq\_3 + unq\_4 + unq\_7 + adjmou + totrev + retdays\_1 + complete\_Mean +

przm\_social\_R + przm\_social\_T + age1\_Mid\_Age + age1\_Old + age1\_Young + age2\_Old, data=train, family="binomial")

summary(mod2)

# All the variables have come significant. Also all the signs of the beta coefficients are in line

# with probablity values less than 5%. So this model can be finalised after checking for absence of Multicollinearity.

## \*\*\*\*\* Model Diagnostics \*\*\*\*\* ##

# Checking For Multicollinearity

library(car)

vif(mod2)

# Variables => Ideally vif values should be < 5. Choosing vif cut-off value of 5,

# 4 of the variables have vif of > 5 , showing Multicollinearity and should be removed from the model.

# Vars to remove from model are mou\_Mean, avgmou, avg3qty, avg6mou

# Re-running Model with above vars omited to remove problem of Multicollinearity.

mod3<-glm(churn ~ totmrc\_Mean + rev\_Range + mou\_Range + change\_mou + drop\_blk\_Mean + drop\_vce\_Range +

mou\_opkv\_Range + months + eqpdays + iwylis\_vce\_Mean + ovrrev\_Mean + avgqty + asl\_flag\_Y +

area\_Cal\_Nrth + area\_texas + area\_nrthflrda + area\_nrthwst + area\_southflrda + area\_southwst +

area\_tenese + refurb\_R + ethnic\_C + ethnic\_N + ethnic\_O + ethnic\_S + ethnic\_U + ethnic\_Z +

hnd\_price\_79.98 + hnd\_price\_105.08 + hnd\_price\_129.98 + hnd\_price\_149.98 + hnd\_price\_199.98 +

hnd\_price\_249.98 + unq\_2 + unq\_3 + unq\_4 + unq\_7 + adjmou + totrev + retdays\_1 + complete\_Mean +

przm\_social\_R + przm\_social\_T + age1\_Mid\_Age + age1\_Old + age1\_Young + age2\_Old, data=train, family="binomial")

summary(mod3)

# var adjmou is coming insignificant. SO rerunning the model after removing 1 var at a time.

# Removing var mou\_Mean

mod4<-glm(churn ~ totmrc\_Mean + rev\_Range + mou\_Range + change\_mou + drop\_blk\_Mean + drop\_vce\_Range +

mou\_opkv\_Range + months + eqpdays + iwylis\_vce\_Mean + ovrrev\_Mean + avgmou + avg3qty + avgqty +

avg6mou + asl\_flag\_Y + area\_Cal\_Nrth + area\_texas + area\_nrthflrda + area\_nrthwst + area\_southflrda +

area\_southwst + area\_tenese + refurb\_R + ethnic\_C + ethnic\_N + ethnic\_O + ethnic\_S + ethnic\_U +

ethnic\_Z + hnd\_price\_79.98 + hnd\_price\_105.08 + hnd\_price\_129.98 + hnd\_price\_149.98 + hnd\_price\_199.98 +

hnd\_price\_249.98 + unq\_2 + unq\_3 + unq\_4 + unq\_7 + adjmou + totrev + retdays\_1 + complete\_Mean +

przm\_social\_R + przm\_social\_T + age1\_Mid\_Age + age1\_Old + age1\_Young + age2\_Old, data=train, family="binomial")

summary(mod4)

# All variables are coming significant.

## \*\*\*\*\* Model Diagnostics \*\*\*\*\* ##

# Checking For Multicollinearity

vif(mod4)

# variables avgmou, avg3qty, have vif of > 5 , showing Multicollinearity and should be removed from the model.

# Rerunning model after removing var avgmou

mod5<-glm(churn ~ totmrc\_Mean + rev\_Range + mou\_Range + change\_mou + drop\_blk\_Mean + drop\_vce\_Range +

mou\_opkv\_Range + months + eqpdays + iwylis\_vce\_Mean + ovrrev\_Mean + avg3qty + avgqty +

avg6mou + asl\_flag\_Y + area\_Cal\_Nrth + area\_texas + area\_nrthflrda + area\_nrthwst + area\_southflrda +

area\_southwst + area\_tenese + refurb\_R + ethnic\_C + ethnic\_N + ethnic\_O + ethnic\_S + ethnic\_U +

ethnic\_Z + hnd\_price\_79.98 + hnd\_price\_105.08 + hnd\_price\_129.98 + hnd\_price\_149.98 + hnd\_price\_199.98 +

hnd\_price\_249.98 + unq\_2 + unq\_3 + unq\_4 + unq\_7 + adjmou + totrev + retdays\_1 + complete\_Mean +

przm\_social\_R + przm\_social\_T + age1\_Mid\_Age + age1\_Old + age1\_Young + age2\_Old, data=train, family="binomial")

summary(mod5)

# All the variables have come significant.

## \*\*\*\*\* Model Diagnostics \*\*\*\*\* ##

# Checking For Multicollinearity

vif(mod5)

# variables avg3qty, have vif of > 5 , showing Multicollinearity and should be removed from the model.

# Rerunning model after removing var avg3qty

mod6<-glm(churn ~ totmrc\_Mean + rev\_Range + mou\_Range + change\_mou + drop\_blk\_Mean + drop\_vce\_Range +

mou\_opkv\_Range + months + eqpdays + iwylis\_vce\_Mean + ovrrev\_Mean + avgqty + avg6mou +

asl\_flag\_Y + area\_Cal\_Nrth + area\_texas + area\_nrthflrda + area\_nrthwst + area\_southflrda +

area\_southwst + area\_tenese + refurb\_R + ethnic\_C + ethnic\_N + ethnic\_O + ethnic\_S + ethnic\_U +

ethnic\_Z + hnd\_price\_79.98 + hnd\_price\_105.08 + hnd\_price\_129.98 + hnd\_price\_149.98 + hnd\_price\_199.98 +

hnd\_price\_249.98 + unq\_2 + unq\_3 + unq\_4 + unq\_7 + adjmou + totrev + retdays\_1 + complete\_Mean +

przm\_social\_R + przm\_social\_T + age1\_Mid\_Age + age1\_Old + age2\_Old, data=train, family="binomial")

summary(mod6)

# All the variables have come significant.

## \*\*\*\*\* Model Diagnostics \*\*\*\*\* ##

# Checking For Multicollinearity

library(car)

vif(mod6)

# All the vif values are well below 5. Thus there is no Multicollinearity. So this model is finalised.

# Checking Confidence Interval

confint(mod6)

## \*\*\*\*\* Model Testing \*\*\*\*\* ##

#Predicted Values ==> Predicting the probability of a customer churning.

pred<-predict(mod6, type="response", newdata=test)

head(pred)

#Assuming cut-off probablity as per the churn rate in data set

table(tele1$churn)/nrow(tele1)

#choosing cutoff value according to kappa value

s<-seq(0.2,0.5,0.01)

n<-1

a<-as.vector(length(s))

for (i in s ) {

print(i)

test$result<-ifelse(test$pred>i,1,0)

a[n]<-confusionMatrix(test$result,test$churn,positive = "1")$overall[2]

print(n)

n=n+1

}

max(a)

#As maximum kappa is related to cutoff 0.23 we would go with this cutoff value

pred1<-ifelse(pred>=0.2380871,1,0)

table(pred1)

# After several itteration in cut-off values, the model is predicting the best at the above Cut-off level.

## \*\*\*\* Checking Prediction Quality \*\*\*\* ##

#Kappa Matrix

library(irr)

kappa2(data.frame(test$churn,pred1))

#Confusion Matrix

library(caret)

confusionMatrix(pred1,test$churn,positive = "1")

table(test$churn)

# The confusion MAtrix shows 2721 correct events and 1881 incorrect events.

# And also shows 8602 correct Non-Events and 6212 incorrect Non-Events

# The model is doing an ok job.

#ROCR Curve

library(ROCR)

pred2<-prediction(pred1,test$churn)

pref<-performance(pred2,"tpr","fpr")

plot(pref,col="red")

abline(0,1,lty=8,col="grey")

auc<-performance(pred2,"auc")

auc

auc<-unlist(slot(auc,"y.values"))

auc

# The auc is 0.5859658 which is more than 0.50.

# Also the curve seems to be well above the grey line.

# So the model seems to be ok and is acceptable.

#Gains Chart

library(gains)

gains(test$churn,predict(mod6,type="response",newdata=test),groups = 10)

#the Gains Chart shows that the top 30% of the probabilities contain 42.2% customers that are likely to churn.

test$prob<-predict(mod6,type="response",newdata=test)

quantile(test$prob,prob=c(0.10,0.20,0.30,0.40,0.50,0.60,0.70,0.80,0.90,1))

#Top 40% of the probability scores lie between 0.2508397 and 0.2747686

#We can use this probablity to extract the data of customers who are highly likely to churn.

### \*\*\*\*\*\*\*\*\*\* Answering Business Questions \*\*\*\*\*\*\*\*\*\* ###

### Top Line Questions of Interest to Senior Management:

# 1. What are the top five factors driving likelihood of churn at Mobicom?

library(lm.beta)

?lm.beta

lm.beta(mod6)

# The above library is not available to me on LMS so I am manually looking at the beta coefficients

head(sort(abs(mod6$coefficients),decreasing = T),10)

summary(mod6)

# from the summary of my final model, "Mod6".

## The model results show that the top 5 factors affecting churn are:

### a. unq\_7 with beta coefficient of 0.735625538

### b. retdays\_1 with beta coefficient of 0.670774312

### c. ethnic\_O with beta coefficient of 0.313047896

### d. area\_nrthwst with beta coefficient of 0.283039470

### e. area\_southflrda with beta coefficient of 0.272490954

# The 1st factor explains, with a unit increase in level 7 of variable uniqsubs, there is 0.735625538 unit increase

# in churn.

# The 2nd Factor explains, with a unit increase in variable retdays, there is 0.670774312 unit increase in churn.

# Same explaination applies to the next 3 variables.

# var ethnic\_O represents var ethnic with level o, var area\_nrthwst is NORTHWEST/ROCKY MOUNTAIN AREA, and

# var area\_southflrda represents var SOUTH FLORIDA AREA, Var retdays\_1 represents valid values for var retdays,i.e.

# values more than "0"

# Thus family bundles should be rolled out for families with 7 unique subscribers. Special offers should be given

# to customers who makes retention calls, at the earliest as per their grieviances. Special plans should be rolled out for

# people with Asian Ethnicity. Special special plans should be rolled out for customers located in NORTHWEST/ROCKY

# MOUNTAIN AREA and SOUTH FLORIDA AREA.

# 2. Validation of survey findings.

# a) Whether "cost and billing" and "network and service quality" are important factors influencing churn behaviour.

# The following variables explain "cost and billing" and "network and service quality"

# Variables totmrc\_Mean i.e. 'base plan charge' representing cost to customer,

# var rev\_Range i.e. 'Range of Revenue(charge amount)' representing billing amount,

# var ovrrev\_Mean = DATOVR\_MEAN + VCEOVR\_MEAN i.e. 'Mean overage revenue' (It is the sum of data and voice

# overage revenues) representing the overage revenue earned from customers after billing the same to them.

# and var totrev i.e. 'Total revenue' representing total revenue earned from customers.

# var totmrc\_Mean has beta coefficient value of -0.005294251 meaning a unit increase in this variable is causing

# decrease in churn by 0.005294251/unit.

# var rev\_Range has beta coefficient value of 0.002095208 meaning a unit increase in this variable is causing

# increase in churn by 0.002095208/unit

# var ovrrev\_Mean has beta coefficient value of 0.007265908 meaning a unit increase in this variable is causing

# increase in churn by 0.007265908/unit

# var totrev has beta coefficient value of 0.000197018 meaning a unit increase in this variable is causing

# increase in churn by 0.000197018/unit

# Having said that, if we notice above mentioned beta values, a unit increase in them is having almost 0% impact

# on churn. SO it seems cost and billing is not very important factors here influencing churn behaviour at Mobicom.

# The following variables explain "network and service quality"

# VARIABLE BETA COEFFICIENT

# mou\_Range 0.000300765

# change\_mou -0.000653979

# drop\_blk\_Mean 0.007668757

# drop\_vce\_Range 0.018691566

# mou\_opkv\_Range -0.001117168

# iwylis\_vce\_Mean -0.015130015

# avgqty 0.001032554

# avg6mou -0.000327649

# adjmou 0.000014846

# retdays\_1 0.670774312

# complete\_Mean -0.001719650

# From the above statistics, data explains the following:

# Variables mou\_Range 1.e. with a unit increase in 'Range of number of minutes of use',

# there is increase in Churn by 0.000300765 units.

# var change\_mou i.e. with a unit increase in 'Percentage change in monthly minutes of

# use vs previous three month average, there is decrease in Churn by -0.000653979 units.

# var drop\_blk\_Mean i.e. with unit increase in 'Mean number of dropped or blocked calls',

# there is an increase in churn by 0.007668757 units

# var drop\_vce\_Range i.e. with a unit increase in 'Range of number of dropped (failed) voice calls',

# there is an increase in Churn by 0.018691566 units.

# var mou\_opkv\_Range i.e. with a unit increase in 'Range of unrounded minutes of use of

# off-peak voice calls, there is a decrease in Churn by -0.001117168 units.

# var iwylis\_vce\_Mean i.e. with a unit increase in 'Mean number of inbound wireless to wireless voice calls',

# there is a decrease in churn by -0.015130015 units.

# var avgqty i.e. with a unit increase in 'Average monthly number of calls over the life of the customer',

# there is an increase in Churn by 0.001032554 units.

# var avg6mou i.e. with unit increase in 'Average monthly minutes of use over the previous six months',

# there is a decrease in Churn by -0.000327649 units.

# var adjmou i.e. with unit increase in 'Billing adjusted total minutes of use over the life of the customer',

# there is an increase in Churn by 0.000014846 units.

# var retdays\_1 representing values captured in the variable retdays i.e. with a unit increase in

# 'Number of days since last retention call', there is an increase in Churn by 0.000014846 units.

# This variable is probably explaining the service quality of the company.

# var complete\_Mean i.e. with unit increase in 'Mean number of completed voice and data calls'

# there is a decrease in Churn by -0.001719650 units

# Of the above variables, the beta coefficient of variable retdays\_1 is expressing a very important

# factor influencing Churn behaviour. That is with the increase in the number of days since a customer

# makes a retention call, the customer's chances of churning is very high. This could probably be because

# their grieviances are not being catered to properly. These customers should be paid more attention to and

# special offers should be made to them depending upon their grieviances.

# 2b) Are data usage connectivity issues turning out to be costly? In other words, is it leading to churn?

# comp\_dat\_Mean - Mean no. of completed data calls.

# plcd\_dat\_Mean - Mean number of attempted data calls placed

# opk\_dat\_Mean - Mean number of off-peak data calls

# blck\_dat\_Mean - Mean no. of blocked / failed data calls

# datovr\_Mean - Mean revenue of data overage.

# datovr\_Range - Range of revenue of data overage

# drop\_dat\_Mean - Mean no. of dropped / failed data calls

# The above variables express data usage connectivity.

quantile(tele$plcd\_dat\_Mean,prob=c(0.10,0.20,0.30,0.40,0.50,0.60,0.70,0.80,0.82,0.84,0.85,0.90,1))

# The Data Quality Report for all the above variables show that only 10% to 15% customers are actualy

# making data calls or using the internet.

# This could be a matter of concern since the global market survey report shows "Subscribers who

# have switched operators in recent months reported two key information sources in their decision:

# the Internet and recommendation of family and friends..

# In this case it seems customers are not really using the internet. So it would be good to work

# towards attaining more customers to use data and also towards proving quality network connectivity

# and service to provide maximum customer satisfaction and reduce Churn.

# Since there is not enough usable data for the above variables they are not showing any influence

# on the Churn Behaviour at Mobicom.

# 3. Would you recommend rate plan migration as a proactive retention strategy?

# Variable ovrrev\_Mean has beta coefficient of 0.007265908.

# var ovrrev\_Mean = DATOVR\_MEAN + VCEOVR\_MEAN i.e. 'Mean overage revenue'

# It is the sum of data and voice overage revenues representing the overage revenue earned

# from customers after billing the same to them.

# The Beta coefficient is not showing a strong impact of overage billing as an influencer

# of churn behaviour.

# Though this might be a matter of concern for few individual customers and they could be

# catered to on case to case basis. But overall rate plan migration as a proactive retention strategy

# might not help much at Mobicom.

# 4. What would be your recommendation on how to use this churn model for prioritisation

# of customers for a proactive retention campaigns in the future?

# Solution:

#Gains Chart

library(gains)

gains(test$churn,predict(mod6,type="response",newdata=test),groups = 10)

#the Gains Chart shows that the top 20% of the probabilities contain 29.5% customers that are highly likely to churn.

# Selecting Customers with high churn rate

test$prob<-predict(mod6,type="response",newdata=test)

quantile(test$prob,prob=c(0.10,0.20,0.30,0.40,0.50,0.60,0.70,0.80,0.90,1))

# Top 20% of the probabilities lie between 0.3042058 and 0.7529329

# Applying cutoff value to predict customers who Will Churn

pred4<-predict(mod6, type="response", newdata=test)

pred4<-ifelse(pred4>=0.3042058 , 1, 0)

table(pred4,test$churn)

Targeted<-test[test$prob>0.3042058 & test$prob<=0.7529329 & test$churn=="1","Customer\_ID"]

Targeted<-as.data.frame(Targeted)

nrow(Targeted)

write.csv(Targeted,"Target\_Customers.csv",row.names = F)

# Thus Using the model can be used to predict customers with high probability of Churn and extract the

# target list using their "Customer ID".

# 5. What would be the target segments for proactive retention campaigns?

# Falling ARPU forecast is also a concern and therefore, Mobicom would like to save their high revenue

# customers besides managing churn. Given a budget constraint of a contact list of 20% of the subscriber pool,

# which subscribers should prioritized if "revenue saves" is also a priority besides controlling churn.

# In other words, controlling churn is the primary objective and revenue saves is the secondary objective.

# Solution:

pred5<-predict(mod6, type="response", newdata=test)

test$prob<-predict(mod6,type="response",newdata=test)

quantile(test$prob,prob=c(0.10,0.20,0.30,0.40,0.50,0.60,0.70,0.80,0.90,1))

pred6<-ifelse(pred5<0.20,"Low\_Score",ifelse(pred5>=0.20 & pred5<0.30,"Medium\_Score","High\_Score"))

table(pred6,test$churn)

str(test$totrev)

quantile(test$totrev,prob=c(0.10,0.20,0.30,0.40,0.50,0.60,0.70,0.80,0.90,1))

Revenue\_Levels<-ifelse(test$totrev<670.660,"Low\_Revenue",ifelse(test$totrev>=670.660 &

test$totrev<1034.281,"Medium\_Revenue","High\_Revenue"))

table(Revenue\_Levels)

table(pred6,Revenue\_Levels)

## Thus this table can be used to select the levels of customers are to be targeted

## and the Target list can be extracted as follows:

test$prob\_levels<-ifelse(pred5<0.20,"Low\_Score",ifelse(pred5>=0.20 & pred5<0.30,"Medium\_Score","High\_Score"))

test$Revenue\_Levels<-ifelse(test$totrev<670.660,"Low\_Revenue",ifelse(test$totrev>=670.660 &

test$totrev<1034.281,"Medium\_Revenue","High\_Revenue"))

Targeted1<-test[test$prob\_levels=="High\_Score" & test$Revenue\_Levels=="High\_Revenue","Customer\_ID"]

Targeted1<-as.data.frame(Targeted1)

nrow(Targeted1)

write.csv(Targeted1,"High\_Revenue\_Target\_Customers.csv",row.names = F)