Analyze the data and make new variables such as:-

* By splitting genre in packages proposed for each id in training dataset.
* Making a variable total watch time of TV.
* Making avg. income per person variable which is Total monthly income divided by No. of adults.
* Making variables for each genres proportion with respect to total watch time.

Then get the total time of each Genre in test data set by making new variables on test data set and getting each genre total watch time for respective ID by TV viewership dataset.

Then allot Current cable operator subscription fees w.r.t The regions given in test dataset and also remove the variables which are not important such as “Home property type”.

Then predict the probabilities of getting each genre with respect to other variables.

Then sort these Probabilites of all 8 genres in decreasing order.

Then check the packages that we need to assign(P1,P2,P3,P4,P5,P6) containing the first element of the sorted list.

Then check for 2nd Highest priority genre in the list, similarly discard the packages and finally by repeating this at last you will get the package that we should allot.

After this get the proposed cost with respect to the allotted package.

Then analyze the important variable needed by different techniques such as graphical approach , boruto package ,By checking correlation and other methods.

Then Normalize the important variables needed for prediction of survey response.

After this predict the survey response in test dataset by creating model by training data in R using randomForest method.

After getting the prediction For those who have prediction “Not interested” ,for them get the total revenue considering “Current cable operator subscription fees (Rs. per month)” along with the Ad revenues (Rs. per hour per subscription) through ads which we will get by multiplying Ad revenues by respective genre watching time.

Their sum will give you Total expected revenue.

Similary for “Yes, I will switch! ” everything remains the same except that instead of “Current cable operator subscription fees” we will consider ” Proposed subscription fees” which we will get through our packages proposed.

R CODE:-

train<-read.csv("training1.csv",stringsAsFactors = FALSE)

test1<-read.csv("testing1.csv",stringsAsFactors = FALSE)

train$one<-0

train$two<-0

train$three<-0

train$four<-0

train$five<-0

train$six<-0

train$six<-0

train$seven<-0

train$eight<-0

for(i in 1:12313)

{

a <-c(strsplit(train$Proposed.Plan..Genres.included.[i],","))

a <- as.data.frame(a)

z<-lengths(a)

for(t in 1:z){

if(a[t,1]== 1)

{

train$one[i]<-1

}

else if(a[t,1]== 2)

{

train$two[i]<-1

}

else if(a[t,1]== 3)

{

train$three[i]<-1

}

else if(a[t,1]== 4)

{

train$four[i]<-1

}

else if(a[t,1]== 5)

{

train$five[i]<-1

}

else if(a[t,1]== 6)

{

train$six[i]<-1

}

else if(a[t,1]== 7)

{

train$seven[i]<-1

}

else if(a[t,1]== 8)

{

train$eight[i]<-1

}

}

}

for(i in 1 : 12313)

{

v<-strsplit(train$Monthly.household.income..in.Rs..[i] , ",")

v<-unlist(v)

v<-paste(v,collapse = "")

train$v[i]<-v

}

for(i in 1 : 1000)

{

v<-strsplit(test1$Monthly.household.income..in.Rs..[i] , ",")

v<-unlist(v)

v<-paste(v,collapse = "")

test1$v[i]<-v

}

test1$v<-as.integer(test1$v)

train$v<-as.integer(train$v)

train$Survey.Response<-as.factor(train$Survey.Response)

train$k <- (train$v - mean(train$v)) / sd(train$v)

train$k1 <- (train$Proposed.subscription.fee..Rs..per.month. - mean(train$Proposed.subscription.fee..Rs..per.month.)) / sd(train$Proposed.subscription.fee..Rs..per.month.)

train$k2 <- (train$Current.cable.operator.subscription.fees..Rs..per.month. - mean(train$Current.cable.operator.subscription.fees..Rs..per.month.)) / sd(train$Current.cable.operator.subscription.fees..Rs..per.month.)

train$k3 <- (train$Genre.2 - mean(train$Genre.2)) / sd(train$Genre.2)

train$k4 <- (train$Genre.3 - mean(train$Genre.3)) / sd(train$Genre.3)

train$k5 <- (train$Genre.4 - mean(train$Genre.4)) / sd(train$Genre.4)

train$k6 <- (train$Genre.5 - mean(train$Genre.5)) / sd(train$Genre.5)

train$k7 <- (train$Genre.6 - mean(train$Genre.6)) / sd(train$Genre.6)

train$k8 <- (train$Genre.7 - mean(train$Genre.7)) / sd(train$Genre.7)

train$k9 <- (train$Genre.1 - mean(train$Genre.1)) / sd(train$Genre.1)

train$k10 <- (train$seven - mean(train$seven)) / sd(train$seven)

train$k11 <- (train$five - mean(train$five)) / sd(train$five)

train$k12 <- (train$one - mean(train$one)) / sd(train$one)

train$k13 <- (train$two - mean(train$two)) / sd(train$two)

train$k14 <- (train$four - mean(train$four)) / sd(train$four)

train$k15 <- (train$Avg.Genre22 - mean(train$Avg.Genre22)) / sd(train$Avg.Genre22)

train$k16 <- (train$Avg.Genre33 - mean(train$Avg.Genre33)) / sd(train$Avg.Genre33)

train$k17 <- (train$Avg.Genre44 - mean(train$Avg.Genre44)) / sd(train$Avg.Genre44)

train$k18 <- (train$Avg.Genre55 - mean(train$Avg.Genre55)) / sd(train$Avg.Genre55)

train$k19 <- (train$Avg.Genre66 - mean(train$Avg.Genre66)) / sd(train$Avg.Genre66)

train$k20 <- (train$Avg.Genre77 - mean(train$Avg.Genre77)) / sd(train$Avg.Genre77)

train$k21 <- (train$Avg.Genre11 - mean(train$Avg.Genre11)) / sd(train$Avg.Genre11)

train$k22 <- (train$Total - mean(train$Total)) / sd(train$Total)

library(randomForest)

model1 <- randomForest(one ~ Genre.1+Genre.2+Genre.3+Genre.4+Genre.5+Genre.6+Genre.7+Genre.8,data = train,ntree=200)

test1$prob1<-predict(model1,newdata=test1)

model2 <- randomForest(two ~ Genre.1+Genre.2+Genre.3+Genre.4+Genre.5+Genre.6+Genre.7+Genre.8,data = train,ntree=200)

test1$prob2<-predict(model2,newdata=test1)

model3 <- randomForest(three ~ Genre.1+Genre.2+Genre.3+Genre.4+Genre.5+Genre.6+Genre.7+Genre.8,data = train,ntree=200)

test1$prob3<-predict(model3,newdata=test1)

model4 <- randomForest(four ~ Genre.1+Genre.2+Genre.3+Genre.4+Genre.5+Genre.6+Genre.7+Genre.8,data = train,ntree=200)

test1$prob4<-predict(model4,newdata=test1)

model5 <- randomForest(five ~ Genre.1+Genre.2+Genre.3+Genre.4+Genre.5+Genre.6+Genre.7+Genre.8,data = train,ntree=200)

test1$prob5<-predict(model5,newdata=test1)

model6 <- randomForest(six ~ Genre.1+Genre.2+Genre.3+Genre.4+Genre.5+Genre.6+Genre.7+Genre.8,data = train,ntree=200)

test1$prob6<-predict(model6,newdata=test1)

model7 <- randomForest(seven ~ Genre.1+Genre.2+Genre.3+Genre.4+Genre.5+Genre.6+Genre.7+Genre.8,data = train,ntree=200)

test1$prob7<-predict(model7,newdata=test1)

model8 <- randomForest(eight ~ Genre.1+Genre.2+Genre.3+Genre.4+Genre.5+Genre.6+Genre.7+Genre.8,data = train,ntree=200)

test1$prob8<-predict(model8,newdata=test1)

p1<-c(1,3,4,5)

p2<-c(2,4,7,5)

p3<-c(1,3,6,8)

p4<-c(1,3,7,2)

p5<-c(2,3,5,8)

p6<-c(1,2,4,6)

for(i in 1:1000){

q<-rev(unlist(strsplit(colnames(sort(test1[i,25:32])),"prob")))

q<-as.integer(q[-c(which(q==""))])

p<-list(p1,p2,p3,p4,p5,p6)

w<-grep(q[1],p)

if(length(w) != 0){

p<-p[w]}else if( length(w)==0){p<-p}

w<-grep(q[2],p)

if(length(w) !=0){

p<-p[w]} else if( length(w)==0){p<-p}

w<-grep(q[3],p)

if(length(w) !=0){

p<-p[w]} else if( length(w)==0){p<-p}

w<-grep(q[4],p)

if(length(w) !=0){

p<-p[w]} else if( length(w)==0){p<-p}

w<-grep(q[5],p)

if(length(w) !=0){

p<-p[w]} else if( length(w)==0){p<-p}

w<-grep(q[6],p)

if(length(w) !=0){

p<-p[w]} else if( length(w)==0){p<-p}

w<-grep(q[7],p)

if(length(w) !=0){

p<-p[w]} else if( length(w)==0){p<-p}

w<-grep(q[8],p)

if(length(w) !=0){

p<-p[w]} else if( length(w)==0){p<-p}

p<-as.data.frame(p)

p11<-as.data.frame(p1)

p22<-as.data.frame(p2)

p33<-as.data.frame(p3)

p44<-as.data.frame(p4)

p55<-as.data.frame(p5)

p66<-as.data.frame(p6)

if(sum(p[,1]==p11[,1])==4){test1$pack[i]<-"p1"}

else if(sum(p[,1]==p22[,1])==4){test1$pack[i]<-"p2"}

else if(sum(p[,1]==p33[,1])==4){test1$pack[i]<-"p3"}

else if(sum(p[,1]==p44[,1])==4){test1$pack[i]<-"p4"}

else if(sum(p[,1]==p55[,1])==4){test1$pack[i]<-"p5"}

else if(sum(p[,1]==p66[,1])==4){test1$pack[i]<-"p6"}

}

test1$newcost<-0

test1$newcost[which(test1$pack=="p1")]<-350

test1$newcost[which(test1$pack=="p2")]<-280

test1$newcost[which(test1$pack=="p3")]<-360

test1$newcost[which(test1$pack=="p4")]<-380

test1$newcost[which(test1$pack=="p5")]<-300

test1$newcost[which(test1$pack=="p6")]<-360

test1$Proposed.subscription.fee..Rs..per.month.<-test1$newcost

train$Survey.Response<-as.integer(train$Survey.Response)

train$Survey.Response[which(train$Survey.Response==1)]<-0

train$Survey.Response[which(train$Survey.Response==2)]<-1

test1$k <- (test1$v - mean(test1$v)) / sd(test1$v)

test1$k1 <- (test1$Proposed.subscription.fee..Rs..per.month. - mean(test1$Proposed.subscription.fee..Rs..per.month.)) / sd(test1$Proposed.subscription.fee..Rs..per.month.)

test1$k2 <- (test1$Current.cable.operator.subscription.fees..Rs..per.month. - mean(test1$Current.cable.operator.subscription.fees..Rs..per.month.)) / sd(test1$Current.cable.operator.subscription.fees..Rs..per.month.)

test1$k3 <- (test1$Genre.2 - mean(test1$Genre.2)) / sd(test1$Genre.2)

test1$k4 <- (test1$Genre.3 - mean(test1$Genre.3)) / sd(test1$Genre.3)

test1$k5 <- (test1$Genre.4 - mean(test1$Genre.4)) / sd(test1$Genre.4)

test1$k6 <- (test1$Genre.5 - mean(test1$Genre.5)) / sd(test1$Genre.5)

test1$k7 <- (test1$Genre.6 - mean(test1$Genre.6)) / sd(test1$Genre.6)

test1$k8 <- (test1$Genre.7 - mean(test1$Genre.7)) / sd(test1$Genre.7)

test1$k9 <- (test1$Genre.1 - mean(test1$Genre.1)) / sd(test1$Genre.1)

test1$k10 <- (test1$prob7 - mean(test1$prob7)) / sd(test1$prob7)

test1$k11 <- (test1$prob5 - mean(test1$prob5)) / sd(test1$prob5)

test1$k12 <- (test1$prob1 - mean(test1$prob1)) / sd(test1$prob1)

test1$k13 <- (test1$prob2 - mean(test1$prob2)) / sd(test1$prob2)

test1$k14 <- (test1$prob4 - mean(test1$prob4)) / sd(test1$prob4)

test1$k15 <- (test1$Avg.Genre22 - mean(test1$Avg.Genre22)) / sd(test1$Avg.Genre22)

test1$k16 <- (test1$Avg.Genre33 - mean(test1$Avg.Genre33)) / sd(test1$Avg.Genre33)

test1$k17 <- (test1$Avg.Genre44 - mean(test1$Avg.Genre44)) / sd(test1$Avg.Genre44)

test1$k18 <- (test1$Avg.Genre55 - mean(test1$Avg.Genre55)) / sd(test1$Avg.Genre55)

test1$k19 <- (test1$Avg.Genre66 - mean(test1$Avg.Genre66)) / sd(test1$Avg.Genre66)

test1$k20 <- (test1$Avg.Genre77 - mean(test1$Avg.Genre77)) / sd(test1$Avg.Genre77)

test1$k21 <- (test1$Avg.Genre11 - mean(test1$Avg.Genre11)) / sd(test1$Avg.Genre11)

test1$k22 <- (test1$Total - mean(test1$Total)) / sd(test1$Total)

finalmodel <- randomForest(Survey.Response~ k+k1+k2+k3+k4+k5+k6+k7+k8+k9+k10+k11+k12+k13+k14+k15+k16+k17+k18+k19+k20+k21+k22,data=train,ntree=200)

test1$Survey.Response<-predict(finalmodel,newdata=test1)

test1$Survey.Response[which(test1$Survey.Response>=0.5)]<-1

test1$Survey.Response[which(test1$Survey.Response<0.5)]<-0

test1$Revenue<-0

for(i in 1:1000)

{

if(test1$Survey.Response[i]== 0)

{

test1$Revenue[i]<-test1$Current.cable.operator.subscription.fees..Rs..per.month.[i] + (65.6/60)\*test1$Genre.1[i] + (67.1/60)\*test1$Genre.2[i] + (42.5/60)\*test1$Genre.3[i] + (35.7/60)\*test1$Genre.4[i] + (34.5/60)\*test1$Genre.5[i] + (28.9/60)\*test1$Genre.6[i] + (21.4/60)\*test1$Genre.7[i] + (19.5/60)\*test1$Genre.8[i]

}

if(test1$Survey.Response[i]== 1)

{

if(test1$pack == "p1"){test1$Revenue[i]<-test1$Proposed.subscription.fee..Rs..per.month.[i] + (65.6/60)\*test1$Genre.1[i] + (42.5/60)\*test1$Genre.3[i] + (35.7/60)\*test1$Genre.4[i] + (34.5/60)\*test1$Genre.5[i]

}

if(test1$pack == "p2"){test1$Revenue[i]<-test1$Proposed.subscription.fee..Rs..per.month.[i] + (67.1/60)\*test1$Genre.2[i] + (35.7/60)\*test1$Genre.4[i] + (34.5/60)\*test1$Genre.5[i] + (21.4/60)\*test1$Genre.7[i]

}

if(test1$pack == "p3"){test1$Revenue[i]<-test1$Proposed.subscription.fee..Rs..per.month.[i] + (65.6/60)\*test1$Genre.1[i] + (42.5/60)\*test1$Genre.3[i] + (28.9/60)\*test1$Genre.6[i] + (19.5/60)\*test1$Genre.8[i]

}

if(test1$pack == "p4"){test1$Revenue[i]<-test1$Proposed.subscription.fee..Rs..per.month.[i] + (65.6/60)\*test1$Genre.1[i] + (67.1/60)\*test1$Genre.2[i] + (42.5/60)\*test1$Genre.3[i] + (21.4/60)\*test1$Genre.7[i]

}

if(test1$pack == "p5"){test1$Revenue[i]<-test1$Proposed.subscription.fee..Rs..per.month.[i] + (67.1/60)\*test1$Genre.2[i] + (42.5/60)\*test1$Genre.3[i] + (34.5/60)\*test1$Genre.5[i] + (19.5/60)\*test1$Genre.8[i]

}

if(test1$pack == "p6"){test1$Revenue[i]<-test1$Proposed.subscription.fee..Rs..per.month.[i] + (65.6/60)\*test1$Genre.1[i] + (67.1/60)\*test1$Genre.2[i] + (35.7/60)\*test1$Genre.4[i] + (28.9/60)\*test1$Genre.6[i]

}

}

}