**R-Code for KNN Classifier**

##### ***Storing the data from raw csv files into different data******frames*** ####################

data1<-read.csv("C:\\Users\\Jash Shah\\Documents\\Mining\_Project\\Raw\_Data\\Train\\2006.csv")

data2<-read.csv("C:\\Users\\Jash Shah\\Documents\\Mining\_Project\\Raw\_Data\\Train\\2007.csv")

###### ***Merging the raw data files of different years*** ################################

my\_data<-rbind(data1,data2)

############## ***Exploring the dataset*** #############################################

head(my\_data)

tail(my\_data)

str(my\_data)

####### ***Storing the required attribute names into a data frame*** ########################

select\_attribute<-c("UniqueCarrier","Month","DayofMonth","DayOfWeek","DepTime","DepDelay", "Origin")

######## ***Creating a subset from the main data set with necessary attribute*** ##############

data\_subset<-subset(my\_data,select = select\_attribute)

######## ***Preprocessing the data: Selecting only “SFO” and “OAK” airports as Origin*** #######

new\_data1<-subset(data\_subset,Origin=="SFO"| Origin=="OAK")

####### ***Selecting “WN – Southwest Airlines”, “ WN – Skywest Airlines” , “UA – United Airlines” , “ DL – Delta Airlines” , “AA – American Airlines” , “NW – Northwestern Airlines”*** #######

new\_data2<-subset(new\_data1,UniqueCarrier=="WN"|UniqueCarrier=="OO"|UniqueCarrier=="UA"|UniqueCarrier=="DL"|UniqueCarrier=="AA"|UniqueCarrier=="NW")

**####### *Removing the rows with missing values* ###################################**

cleansed\_dataset<-new\_data2[complete.cases(new\_data2),]

**####### *Removing the N/A values* #################################################**

Cleansed\_dataset<-na.omit(new\_data2)

**####### *Converting nominal value to numeric (UniqueCarrier)* #######################**

cleansed\_dataset$UniqueCarrier=ifelse(cleansed\_dataset$UniqueCarrier=="WN",1,

ifelse(cleansed\_dataset$UniqueCarrier=="OO",2,

ifelse(cleansed\_dataset$UniqueCarrier=="UA",3,

ifelse(cleansed\_dataset$UniqueCarrier=="DL",4,

ifelse(cleansed\_dataset$UniqueCarrier=="AA",5,

ifelse(cleansed\_dataset$UniqueCarrier=="NW",6, "NA"))))))

**######## *Adding Data Label for Departure Delay* ##################################**

cleansed\_dataset$Label\_Departure\_Delay\_Time<-ifelse(cleansed\_dataset$DepDelay>10, "Yes","No")

**####### *Converting categorical to numeric (Origin; SFO=1, OAK=0 )* ###################**

cleansed\_dataset$Origin<-ifelse(cleansed\_dataset$Origin=="SFO",1,0)

**####### *Filtering out DepDelay for KNN analysis* ##################**

cleansed\_dataset<-subset(cleansed\_dataset,select=-c(DepDelay))

**####### *Converting char to numeric for attribute UniqueCarrier* #######################**

cleansed\_dataset$UniqueCarrier=as.numeric(cleansed\_dataset$UniqueCarrier)

**####### *Creating function to normalize the values* ##################################**

function(x) {

return ((x - min(x)) / (max(x) - min(x))) }

**####### *Applying min-max normalization to cleansed\_dataset* ########################**

train\_normalize<-as.data.frame(lapply(cleansed\_dataset[1:6], normalize))

**####### *Storing the Cleansed\_Dataset Labels in a data frame* ########################**

train\_label<-cleansed\_dataset$Label\_Departure\_Delay\_Time

**######## *Installing package “class” for KNN classifier functions* ######################**

install.packages("class")

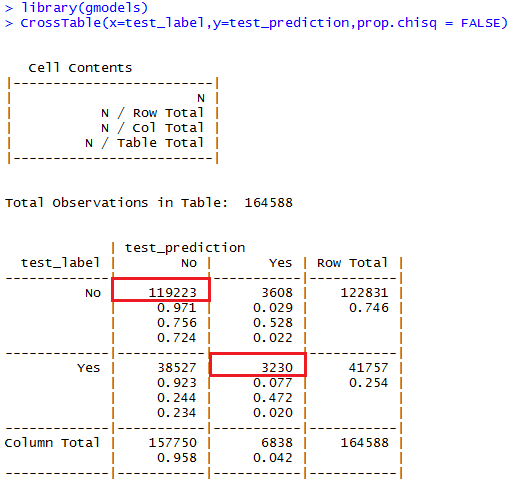
library(class)

**####### *Predicting the test labels using trained dataset* #############################**

test\_prediction<-knn(train=train\_normalize,test=test\_normalize,cl=train\_label,k=400)

**######### *Installing package “gmodels” to apply cross-table validations* #############**

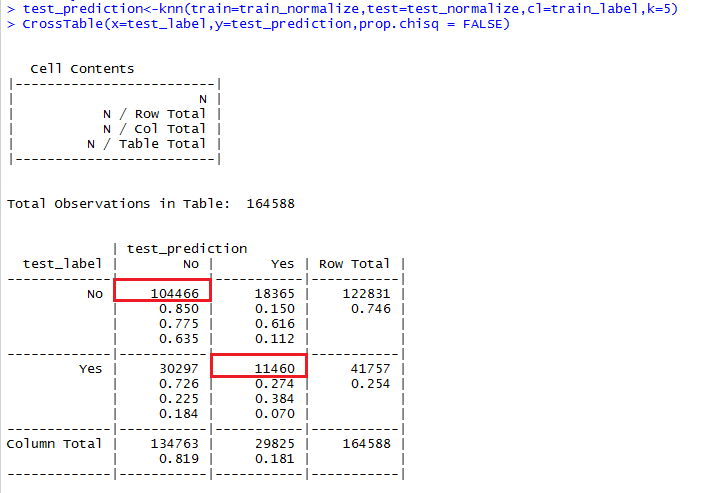
install.packages("gmodels")



**Efficiency = 122453/164588 =74.399%**

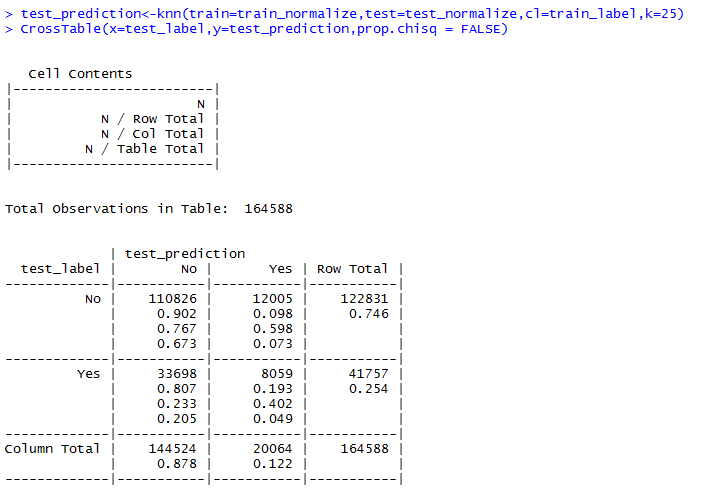
**Calculating Efficiency for different values of K**

***For K=5***



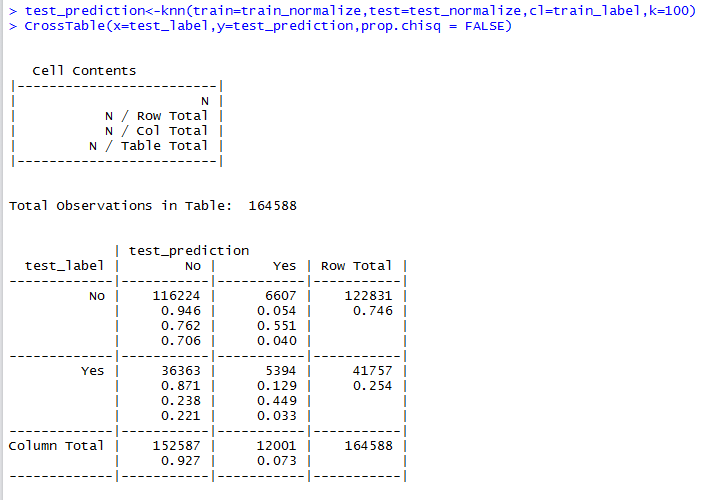
**Accuracy= 115926/164588 = 70.43%**

***For k=25***



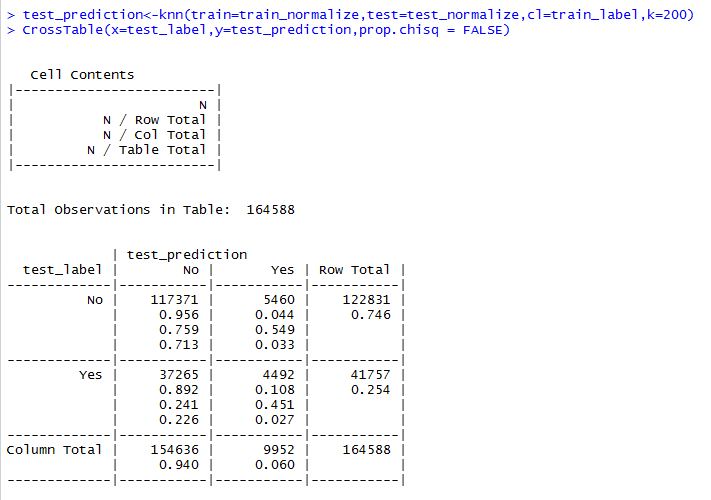
**Accuracy= 118885/164588 = 72.23%**

***For K=100***



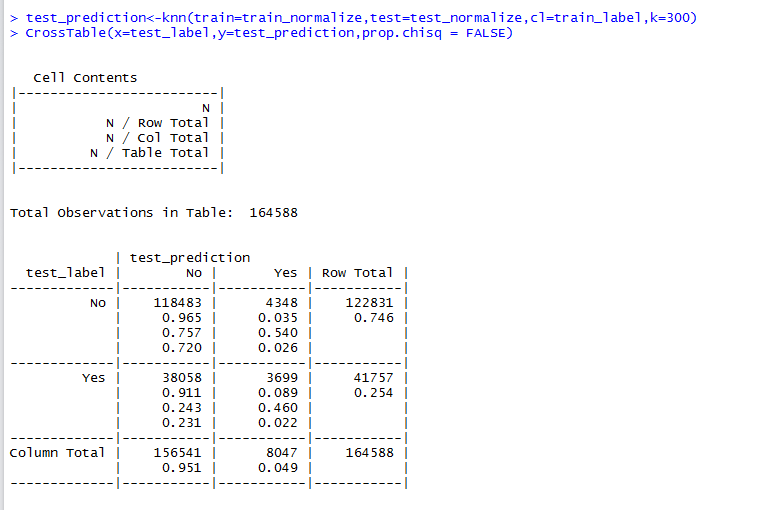
**Accuracy= 121618/164588 = 73.89%**

***For K=200***



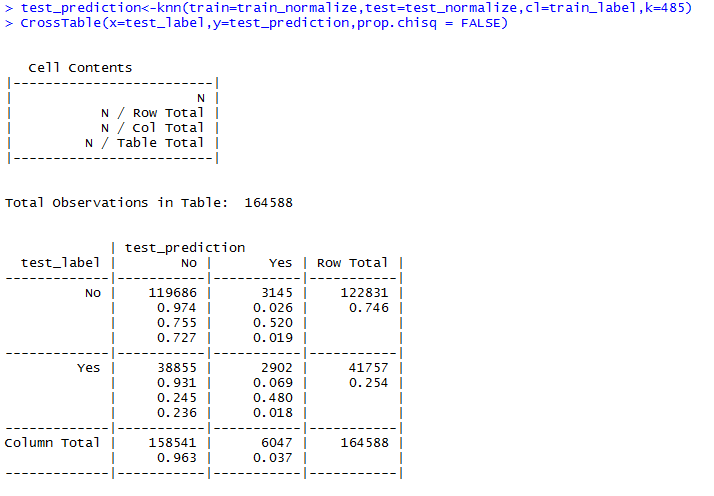
**Accuracy= 121863/164588 = 74.04%**

***K=300***



**Accuracy = 122182/164588 = 74.23**

***For K=485***



**Accuracy = 12588/164588 = 74.48%**