**#Course : CS 513**

**#Vandna Yadav**

#####################################################################################

rm(list=ls())

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*loading the dataset\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

data<-read.csv("C:/Users/vandna/Desktop/Stevens/SEM 2/513/Project/Code/dataset/2008.csv")

odataset<-data

attach(odataset)

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*categorizing Departure Delay \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

odataset$DepDelay\_cat[DepDelay < -5] <- "EARLY"

odataset$DepDelay\_cat[DepDelay >= -5 & DepDelay <= 5] <- "ONTIME"

odataset$DepDelay\_cat[DepDelay > 5] <- "LATE"

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*categorizing Arrival Delay\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

odataset$ArrDelay\_cat[ArrDelay < -5] <- "EARLY"

odataset$ArrDelay\_cat[ArrDelay >= -5 & ArrDelay <= 10] <- "ONTIME"

odataset$ArrDelay\_cat[ArrDelay > 10]<-"LATE"

#\*\*\*\*\*\*\*\*\*\*\*creating a data frame

filtereddataset <- data.frame(Month,DayofMonth,DayOfWeek,DepTime,CRSDepTime,ArrTime,CRSArrTime,UniqueCarrier,Origin,Dest,Distance,Diverted,Cancelled,CarrierDelay,WeatherDelay,NASDelay,SecurityDelay,LateAircraftDelay,ArrDelay,DepDelay,"ArrDelay\_cat"=odataset$ArrDelay\_cat,"DepDelay\_cat"=odataset$DepDelay\_cat)

detach (odataset)

attach(filtereddataset)

#we are filtering out the flights which are cancelled

filtereddataset<-filtereddataset[filtereddataset[,"Cancelled"]==0,]

#we are filtering out the flights which are Diverted

filtereddataset<-filtereddataset[filtereddataset[,"Diverted"]==0,]

#we are filtering out the flights which are latedeparture

filtereddataset<-filtereddataset[filtereddataset[,"DepDelay"]>=5,]

#\*\*\*\*\*we are taking 3 highest freq unique carrier

filtereddataset<-filtereddataset[filtereddataset$UniqueCarrier =="WN" | filtereddataset$UniqueCarrier =="OO" | filtereddataset$UniqueCarrier =="AA", ]

#\*\*\*\*\*we are taking 3 highest freq origin

filtereddataset<-filtereddataset[filtereddataset$Origin =="ATL" | filtereddataset$Origin =="ORD" | filtereddataset$Origin =="LAX", ]

#removing arrival time NA

filtereddataset<-filtereddataset[complete.cases(filtereddataset[,6]),]

attach(filtereddataset)

library(plyr)

data<-join(filtereddataset,count(filtereddataset,'Dest'))

attach(filtereddataset)

#\*\*\*\*\*\*\*categoraizing dest type according to the num of flights

filtereddataset$Dest\_Type[data$freq > 500 & data$freq<1000] <- "Med Busy"

filtereddataset$Dest\_Type[data$freq < 500] <- "Less Busy"

filtereddataset$Dest\_Type[data$freq >1000] <- "High Busy"

detach(odataset)

#\*\*\*\*\*\*\*\*\*\*\*\*normalization & removal of na values functions \*\*\*\*\*\*\*\*\*\*\*

na.zero <- function (x) {

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

return(x)

}

mmnorm <-function(x) {z<-((x-min(x))/(max(x)-min(x)));return(z) }

data<-filtereddataset

data\_new<-cbind(Month=mmnorm(data$Month),

DayOfMonth=mmnorm(data$DayofMonth),

DayOfWeek=mmnorm(data$DayOfWeek),

CRSDepTime=mmnorm(data$CRSDepTime),

CRSArrTime=mmnorm(data$CRSArrTime) ,

UniqueCarrier=mmnorm(as.numeric(factor(data$UniqueCarrier))),

Origin=mmnorm(as.numeric(factor(data$Origin))),

Dest=mmnorm(as.numeric(factor(data$Dest\_Type))),

ArrDelay\_cat=as.character(data$ArrDelay\_cat)

)

#\*\*\*\*\*taking 5000 entries\*\*\*\*\*\*\*

idx1<-seq(1:5000)

data\_new<-data\_new[idx1,]

#\*\*\*\*sampling data\*\*\*\*\*\*\*\*\*\*\*\*\*\*

idx<-sample(nrow(data\_new),as.integer(.70\*nrow(data\_new)))

#\*\*\*\*training & test dataset\*\*\*\*\*\*\*\*\*\*\*

training<-data\_new[idx,]

test<-data\_new[-idx,]

library(class)

####to find proper k value::

#running knn 50 time for itterative k starting from k=1 to k=20

# here which k's average error rate is minimumm,that k is best.

for (j in 1:40){

counter<- 0

total<-0

for (i in 1:50) {

newpredict<-knn(training[,-9],test[,-9],training[,9],k <- j)

newresults<-cbind(test,as.character(newpredict) )

wrong<-newresults[,9]!=newresults[,10]

rate<-sum(wrong)/length(wrong)

rates<-rbind(rate,rate)

total<-total+rate

counter<-counter+1

}

print(j)

avg=total/counter

print(avg)

}

######################

#\*\*\*\*\*\*applying knn\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

newpredict<-knn(training[,-9],test[,-9],training[,9],k=30)

newresults<-cbind(test,as.character(newpredict) )

head(newresults)

table(newresults[,9],newresults[,10])

#############################################################################

#####################-----------KKnn---------------------

rm(list=ls())

library(kknn)

?kknn

#normalization and na removal functions

na.zero <- function (x) {

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

return(x)

}

mmnorm <-function(x) {z<-((x-min(x))/(max(x)-min(x)));return(z) }

#loading dataset

filtereddataset<-read.csv("C:/Users/vandna/Desktop/Stevens/SEM 2/513/Project/Code/dataset/new4.csv")

attach(filtereddataset)

#categorizing Arrival delay

filtereddataset$ArrDelay\_catnew[ArrDelay < -5] <- "Early"

filtereddataset$ArrDelay\_catnew[ArrDelay >= -5 & ArrDelay <= 5] <- "ONTIME"

filtereddataset$ArrDelay\_catnew[ArrDelay > 5 & ArrDelay < 20] <- "late"

filtereddataset$ArrDelay\_catnew[ArrDelay >= 20] <- "VERY\_late"

data<-filtereddataset

data\_new\_k<-cbind(

DayOfMonth=mmnorm(data$DayofMonth),

DayOfWeek=mmnorm(data$DayOfWeek),

CRSDepTime=mmnorm(data$CRSDepTime),

CRSArrTime=mmnorm(data$CRSArrTime) ,

UniqueCarrier=mmnorm(as.numeric(factor(data$UniqueCarrier))),

Origin=mmnorm(as.numeric(factor(data$Origin))),

Dest=mmnorm(as.numeric(factor(data$Dest\_Type))),

ArrDelay\_cat=as.character(data$ArrDelay\_catnew)

)

data\_new\_k<-as.data.frame(data\_new\_k)

data\_new\_k<-na.omit(data\_new\_k)

factor(data\_new\_k$ArrDelay\_cat)

is.data.frame(data\_new\_k)

idx1<-seq(1:5000)

data\_new\_k<-data\_new\_k[idx1,]

idx<-sample(nrow(data\_new\_k),as.integer(.70\*nrow(data\_new\_k)))

trainingk<-data\_new\_k[idx,]

testk<-data\_new\_k[-idx,]

is.data.frame(trainingk)

#applying kknn

predict\_1 <- kknn(formula=ArrDelay\_cat~., trainingk, testk, k=38,kernel="optimal")

head(predict\_1)

fitWalc <- fitted(predict\_1)

results <- cbind(testk$ArrDelay\_cat, fitWalc)

wrong <- results[,1]!=results[,2]

rateWalc <- sum(wrong)/length(wrong)

rateWalc

rm(list=ls())

#install.packages("plyr")

library(plyr)

#defining the function for normalization

mmnorm<-function(x)

{

z<-((x-min(x))/(max(x)-min(x)))

return(z)

}

#defining the na.zero function

na.zero <- function (x) {

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

return(x)

}

data<-read.csv("C:/Users/vandna/Desktop/Stevens/SEM 2/513/Project/Code/dataset/20April.csv")

filtereddataset<-sample(data)

filtereddataset$CarrierDelay[is.na(filtereddataset$CarrierDelay)] <- 0

filtereddataset$WeatherDelay[is.na(filtereddataset$WeatherDelay)] <- 0

filtereddataset$NASDelay[is.na(filtereddataset$NASDelay)] <- 0

filtereddataset$SecurityDelay[is.na(filtereddataset$SecurityDelay)] <- 0

filtereddataset$LateAircraftDelay[is.na(filtereddataset$LateAircraftDelay)] <- 0

attach(filtereddataset)

#\*\*\*\*\*\*categorizing Day of Month

filtereddataset$DayofMonth\_cat[DayofMonth > 15] <- "second\_half"

filtereddataset$DayofMonth\_cat[DayofMonth <= 15] <- "First\_half"

#\*\*\*\*\*\*\*categorizing Day OF Week

filtereddataset$week\_cat[DayOfWeek == 1 | DayOfWeek ==7] <- "weekend"

filtereddataset$week\_cat[DayOfWeek < 7 & DayOfWeek > 1] <- "weekday"

#\*\*\*\*\*\*\*categorizing Arrival Delay

filtereddataset$ArrDelay\_cat[ArrDelay <= 5] <- "EARLY"

filtereddataset$ArrDelay\_cat[ArrDelay > 5 & ArrDelay <= 60] <- "ONTIME"

filtereddataset$ArrDelay\_cat[ArrDelay > 60] <- "LATE"

#\*\*\*\*\*categorizing Distance

filtereddataset$Distance\_cat[Distance <= 1000] <- "short\_distance"

filtereddataset$Distance\_cat[Distance > 1000 & Distance <= 2000] <- "Mid\_distance"

filtereddataset$Distance\_cat[Distance > 2000] <- "Long\_distance"

filtereddataset<-join(filtereddataset,count(filtereddataset,'Dest'))

#\*\*\*\*\*\*categorizing Dest type

filtereddataset$Dest\_Type[filtereddataset$freq < 500] <- "Less Busy"

filtereddataset$Dest\_Type[filtereddataset$freq > 500 & filtereddataset$freq<1000] <- "Med Busy"

filtereddataset$Dest\_Type[filtereddataset$freq >1000] <- "High Busy"

#creating data frame

data\_new<-data.frame(DayOfMonth=(filtereddataset$DayofMonth\_cat),

DayOfWeek=(filtereddataset$week\_cat),

CRSDepTime=(na.zero(filtereddataset$CRSDepTime)),

CRSArrTime=(na.zero(filtereddataset$CRSArrTime)),

UniqueCarrier=(filtereddataset$UniqueCarrier),

Origin=(filtereddataset$Origin),

Dest=(filtereddataset$Dest\_Type),

Distance=(filtereddataset$Distance\_cat),

ArrDelay\_cat=(filtereddataset$ArrDelay\_cat)

)

set.seed(9850)

#creating training and test dataset based on 70%-30% ratio

idx<-sample(nrow(data\_new),as.integer(.70\*nrow(data\_new)))

training<-data\_new[idx,]

test<-data\_new[-idx,]

#fitting the classification tree using the rpart function

library(rpart)

dtm <- rpart(ArrDelay\_cat~.,data= training, method= "class")

printcp(dtm)

#gives the summary

summary(dtm)

#plotting the tree

library(rpart.plot)

#predicting for test data ad testing the accuracy

rpart.plot(dtm, type=1, extra=101)

p3 <- predict(dtm,test,type="class")

table(test[,9], predicted= p3)

rm(list=ls())

library(plyr)

#defining the function for normalization

mmnorm<-function(x)

{

z<-((x-min(x))/(max(x)-min(x)))

return(z)

}

#defining the na.zero function

na.zero <- function (x) {

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

return(x)

}

#\*\*\*\*loading dataset

data<-read.csv("C:/Users/vandna/Desktop/Stevens/SEM 2/513/Project/Code/dataset/new4.csv")

#\*\*\*Sampling data

filtereddataset<-sample(data)

filtereddataset$CarrierDelay[is.na(filtereddataset$CarrierDelay)] <- 0

filtereddataset$WeatherDelay[is.na(filtereddataset$WeatherDelay)] <- 0

filtereddataset$NASDelay[is.na(filtereddataset$NASDelay)] <- 0

filtereddataset$SecurityDelay[is.na(filtereddataset$SecurityDelay)] <- 0

filtereddataset$LateAircraftDelay[is.na(filtereddataset$LateAircraftDelay)] <- 0

attach(filtereddataset)

#\*\*\*\*\*\*categorizing Day Of month

filtereddataset$DayofMonth\_cat[DayofMonth > 15] <- "second\_half"

filtereddataset$DayofMonth\_cat[DayofMonth <= 15] <- "First\_half"

#\*\*\*\*\*\*categorizing Day of week

filtereddataset$week\_cat[DayOfWeek == 1 | DayOfWeek ==7] <- "weekend"

filtereddataset$week\_cat[DayOfWeek < 7 & DayOfWeek > 1] <- "weekday"

#\*\*\*\*\*\*categorizing Arrival Delay

filtereddataset$ArrDelay\_cat[ArrDelay <= 5] <- "EARLY"

filtereddataset$ArrDelay\_cat[ArrDelay > 5 & ArrDelay <= 60] <- "ONTIME"

filtereddataset$ArrDelay\_cat[ArrDelay > 60] <- "LATE"

#\*\*\*\*\*\*categorizing Distance

filtereddataset$Distance\_cat[Distance <= 1000] <- "short\_distance"

filtereddataset$Distance\_cat[Distance > 1000 & Distance <= 2000] <- "Mid\_distance"

filtereddataset$Distance\_cat[Distance > 2000] <- "Long\_distance"

filtereddataset<-join(filtereddataset,count(filtereddataset,'Dest'))

#\*\*\*\*\*\*categorizing Destination

filtereddataset$Dest\_Type[filtereddataset$freq < 500] <- "Less Busy"

filtereddataset$Dest\_Type[filtereddataset$freq > 500 & filtereddataset$freq<1000] <- "Med Busy"

filtereddataset$Dest\_Type[filtereddataset$freq >1000] <- "High Busy"

#\*\*\*\*\*\*categorizing Arrival Delay value

filtereddataset$ArrDelay\_value[filtereddataset$ArrDelay\_cat == "ONTIME"] <- 1

filtereddataset$ArrDelay\_value[filtereddataset$ArrDelay\_cat == "Mid Delay"] <- 2

filtereddataset$ArrDelay\_value[filtereddataset$ArrDelay\_cat == "High Delay"] <- 3

data <-filtereddataset

data\_new<-cbind(

DayOfMonth=data$DayofMonth,

DayOfWeek=data$DayOfWeek,

CRSDepTime=data$CRSDepTime,

CRSArrTime=data$CRSArrTime,

UniqueCarrier=data$UniqueCarrier,

Origin=data$Origin,

Dest=data$Dest\_Type,

DepDelay=data$DepDelay,

weatherdly=data$WeatherDelay,

NASdly=data$NASDelay,

Securitydly=data$SecurityDelay,

LateAircraftdly=data$LateAircraftDelay,

ArrDelay\_cat=as.numeric(factor(data$ArrDelay\_cat))

)

detach (filtereddataset)

attach(data\_new)

#install.packages("C50")

require(C50)

set.seed(9850)

idx<-sample(nrow(data\_new),as.integer(.70\*nrow(data\_new)))

#generating training dataset

training<-data\_new[idx,]

nrow(training)

#generating test dataset

test<-data\_new[-idx,]

#applying C5.0

m1 <- C5.0(training[,-13],factor(training[,13]))

#gives summary of the tree

summary(m1)

#plotting the tree

plot(m1)

#to check the accuracy of model

result<-predict(m1,test,type="class")

rTable<-table(predict=result,test=test[,13])

accuracy=(sum(diag(rTable))/nrow(test))

accuracy

rm(list=ls())

library(plyr)

#defining the function for normalization

mmnorm<-function(x)

{

z<-((x-min(x))/(max(x)-min(x)))

return(z)

}

#defining the na.zero function

na.zero <- function (x) {

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

return(x)

}

#\*\*\*\*loading dataset

data<-read.csv("C:/Users/vandna/Desktop/Stevens/SEM 2/513/Project/Code/dataset/new4.csv")

filtereddataset<-sample(data)

filtereddataset$CarrierDelay[is.na(filtereddataset$CarrierDelay)] <- 0

filtereddataset$WeatherDelay[is.na(filtereddataset$WeatherDelay)] <- 0

filtereddataset$NASDelay[is.na(filtereddataset$NASDelay)] <- 0

filtereddataset$SecurityDelay[is.na(filtereddataset$SecurityDelay)] <- 0

filtereddataset$LateAircraftDelay[is.na(filtereddataset$LateAircraftDelay)] <- 0

attach(filtereddataset)

filtereddataset$DayofMonth\_cat[DayofMonth > 15] <- "second\_half"

filtereddataset$DayofMonth\_cat[DayofMonth <= 15] <- "First\_half"

filtereddataset$week\_cat[DayOfWeek == 1 | DayOfWeek ==7] <- "weekend"

filtereddataset$week\_cat[DayOfWeek < 7 & DayOfWeek > 1] <- "weekday"

filtereddataset$ArrDelay\_cat[ArrDelay <= 5] <- "EARLY"

filtereddataset$ArrDelay\_cat[ArrDelay > 5 & ArrDelay <= 60] <- "ONTIME"

filtereddataset$ArrDelay\_cat[ArrDelay > 60] <- "LATE"

filtereddataset$Distance\_cat[Distance <= 1000] <- "short\_distance"

filtereddataset$Distance\_cat[Distance > 1000 & Distance <= 2000] <- "Mid\_distance"

filtereddataset$Distance\_cat[Distance > 2000] <- "Long\_distance"

filtereddataset<-join(filtereddataset,count(filtereddataset,'Dest'))

filtereddataset$Dest\_Type[filtereddataset$freq < 500] <- "Less Busy"

filtereddataset$Dest\_Type[filtereddataset$freq > 500 & filtereddataset$freq<1000] <- "Med Busy"

filtereddataset$Dest\_Type[filtereddataset$freq >1000] <- "High Busy"

filtereddataset$ArrDelay\_value[filtereddataset$ArrDelay\_cat == "ONTIME"] <- 1

filtereddataset$ArrDelay\_value[filtereddataset$ArrDelay\_cat == "Mid Delay"] <- 2

filtereddataset$ArrDelay\_value[filtereddataset$ArrDelay\_cat == "High Delay"] <- 3

data <-filtereddataset

data\_new<-cbind(UniqueCarrier=data$UniqueCarrier,

DepDelay=data$DepDelay,

weatherdly=data$WeatherDelay,

NASdly=data$NASDelay,

Securitydly=data$SecurityDelay,

LateAircraftdly=data$LateAircraftDelay,

ArrDelay\_cat=as.numeric(factor(data$ArrDelay\_cat))

)

idx1<-seq(1:1000)

data\_new<-data\_new[idx1,]

idx<-sample(nrow(data\_new),as.integer(.70\*nrow(data\_new)))

#generating training dataset

training<-data\_new[idx,]

#generating test dataset

test<-data\_new[-idx,]

m1 <- C5.0(training[,-7],factor(training[,7]),trials=10)

#gives summary of the tree

summary(m1)

#plotting the tree

plot(m1)

#to check the accuracy of model

result<-predict(m1,test,type="class")

rTable<-table(predict=result,test=test[,7])

accuracy=(sum(diag(rTable))/nrow(test))

accuracy

rm(list=ls())

#\*\*\*\*\*\*loading dataset

data<-read.csv("C:/Users/vandna/Desktop/Stevens/SEM 2/513/Project/Code/dataset/2008.csv")

odataset<-data

attach(odataset)

odataset$DepDelay\_cat[DepDelay < -5] <- "EARLY"

odataset$DepDelay\_cat[DepDelay >= -5 & DepDelay <= 5] <- "ONTIME"

odataset$DepDelay\_cat[DepDelay > 5] <- "LATE"

odataset$ArrDelay\_cat[ArrDelay < -5] <- "EARLY"

odataset$ArrDelay\_cat[ArrDelay >= -5 & ArrDelay <= 10] <- "ONTIME"

odataset$ArrDelay\_cat[ArrDelay > 10]<-"LATE"

filtereddataset <- data.frame(Month,DayofMonth,DayOfWeek,DepTime,CRSDepTime,ArrTime,CRSArrTime,UniqueCarrier,Origin,Dest,Distance,Diverted,Cancelled,CarrierDelay,WeatherDelay,NASDelay,SecurityDelay,LateAircraftDelay,ArrDelay,DepDelay,"ArrDelay\_cat"=odataset$ArrDelay\_cat,"DepDelay\_cat"=odataset$DepDelay\_cat)

detach (odataset)

attach(filtereddataset)

#departure delay

#Diverted

#we are filtering out the flights which are cancelled

filtereddataset<-filtereddataset[filtereddataset[,"Cancelled"]==0,]

#we are filtering out the flights which are Diverted

filtereddataset<-filtereddataset[filtereddataset[,"Diverted"]==0,]

#we are filtering out the flights which are latedeparture

filtereddataset<-filtereddataset[filtereddataset[,"DepDelay"]>=5,]

filtereddataset<-filtereddataset[filtereddataset$UniqueCarrier =="WN" | filtereddataset$UniqueCarrier =="OO" | filtereddataset$UniqueCarrier =="AA", ]

filtereddataset<-filtereddataset[filtereddataset$Origin =="ATL" | filtereddataset$Origin =="ORD" | filtereddataset$Origin =="LAX", ]

#removing arrival time NA

filtereddataset<-filtereddataset[complete.cases(filtereddataset[,6]),]

attach(filtereddataset)

library(plyr)

data<-join(filtereddataset,count(filtereddataset,'Dest'))

attach(filtereddataset)

filtereddataset$Dest\_Type[data$freq > 500 & data$freq<1000] <- "Med Busy"

filtereddataset$Dest\_Type[data$freq < 500] <- "Less Busy"

filtereddataset$Dest\_Type[data$freq >1000] <- "High Busy"

detach(odataset)

summary(filtereddataset)

na.zero <- function (x) {

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

return(x)

}

mmnorm <-function(x) {z<-((x-min(x))/(max(x)-min(x)));return(z) }

data<-filtereddataset

data\_new<-cbind(DayOfWeek=data$DayOfWeek,

CRSDepTime=data$CRSDepTime,

CRSArrTime=data$CRSArrTime,

UniqueCarrier=as.numeric(factor(data$UniqueCarrier)),

Origin=as.numeric(factor(data$Origin)),

Dest=as.numeric(factor(data$Dest\_Type)),

DepDelay=mmnorm(data$DepDelay),

ArrDelay\_cat=as.numeric(factor(data$ArrDelay\_catnew))

)

data\_new<-as.data.frame (data\_new)

data\_new<-na.omit(data\_new)

factor(data\_new$ArrDelay\_cat)

is.data.frame(data\_new)

idx1<-seq(1:100)

data\_new<-data\_new[idx1,]

idx<-sample(nrow(data\_new),as.integer(.70\*nrow(data\_new)))

training<-data\_new[idx,]

test<-data\_new[-idx,]

is.data.frame(training)

if(!require(neuralnet)) {

install.packages("neuralnet"); require(neuralnet)}

rate<-0

wrong<-0

net.ArrDelay <- neuralnet(ArrDelay\_cat~DayOfWeek+CRSDepTime+CRSArrTime+

UniqueCarrier+Origin+Dest+DepDelay,

training,hidden=5, threshold=0.01,stepmax=1e6)

#to plot the neuralnet

plot(net.ArrDelay)

net.result1 <- compute(net.ArrDelay, subset(test, select=-ArrDelay\_cat))

fit <- round(net.result1$net.result, digits = 0)

results <- cbind(test$ArrDelay\_cat, fit)

wrong <- results[,1]!=results[,2]

rate <- sum(wrong)/length(wrong)

rate

#####################################################################################