#loaded needed packages

library(ISLR)

library(caret)

library(randomForest)

library(datasets)

library(dplyr)

library(readxl)

#Import Needed Files

price <- read\_excel("/Users/chrismcmanus/Documents/Effective Rent By Submarket .xlsx")

food <- read.csv("/Users/chrismcmanus/Desktop/Attributes/foods1.csv")

highway <- read.csv("/Users/chrismcmanus/Desktop/Attributes/highway.csv")

historic <- read.csv("/Users/chrismcmanus/Desktop/Attributes/hp.csv")

entertainment <- read.csv("/Users/chrismcmanus/Desktop/Attributes/entertainment.csv")

medical <- read.csv("/Users/chrismcmanus/Desktop/Attributes/mf.csv")

office <- read.csv("/Users/chrismcmanus/Desktop/Attributes/offices.csv")

park <- read.csv("/Users/chrismcmanus/Desktop/Attributes/park.csv")

education <- read.csv("/Users/chrismcmanus/Desktop/Attributes/schools.csv")

shop <- read.csv("/Users/chrismcmanus/Desktop/Attributes/shops.csv")

bus <- read.csv("/Users/chrismcmanus/Desktop/Attributes/bs.csv")

#Merge files together

attribute <- merge.data.frame(food, highway, by="ID", all.x=TRUE, all.y=TRUE)

attribute <- merge.data.frame(attribute, historic, by="ID", all.x=TRUE, all.y=TRUE)

attribute <- merge.data.frame(attribute, entertainment, by="ID", all.x=TRUE, all.y=TRUE)

attribute <- merge.data.frame(attribute, medical, by="ID", all.x=TRUE, all.y=TRUE)

attribute <- merge.data.frame(attribute, office, by="ID", all.x=TRUE, all.y=TRUE)

attribute <- merge.data.frame(attribute, park, by="ID", all.x=TRUE, all.y=TRUE)

attribute <- merge.data.frame(attribute, education, by="ID", all.x=TRUE, all.y=TRUE)

attribute <- merge.data.frame(attribute, shop, by="ID", all.x=TRUE, all.y=TRUE)

attribute <- merge.data.frame(attribute, bus, by="ID", all.x=TRUE, all.y=TRUE)

attribute <- merge.data.frame(attribute, price, by="ID", all.x=TRUE, all.y=TRUE)

#Setting NA's to 0

attribute$Density.of.Dining.and.Beverage.Options[is.na(attribute$Density.of.Dining.and.Beverage.Options)] <- 0

attribute$Density.of.Highways[is.na(attribute$Density.of.Highways)] <- 0

attribute$Density.of.Historical.Properties[is.na(attribute$Density.of.Historical.Properties)] <- 0

attribute$Density.of.Entertainment.Options[is.na(attribute$Density.of.Entertainment.Options)] <- 0

attribute$Density.of.Medical.Facilities[is.na(attribute$Density.of.Medical.Facilities)] <- 0

attribute$Density.of.Offices[is.na(attribute$Density.of.Offices)] <- 0

attribute$Density.of.Parks[is.na(attribute$Density.of.Parks)] <- 0

attribute$Density.of.Educational.Institutions[is.na(attribute$Density.of.Educational.Institutions)] <- 0

attribute$Density.of.Shops[is.na(attribute$Density.of.Shops)] <- 0

attribute$Density.of.Bus.Stops[is.na(attribute$Density.of.Bus.Stops)] <- 0

#Creating new data frames for each property type

One\_Bed\_Property <- attribute %>% select(c(Density.of.Dining.and.Beverage.Options,

Density.of.Highways,

Density.of.Historical.Properties,

Density.of.Entertainment.Options,

Density.of.Medical.Facilities,

Density.of.Offices,

Density.of.Parks,

Density.of.Educational.Institutions,

Density.of.Shops,

Density.of.Bus.Stops,

Proximity\_to\_Uptown,

Proximity\_to\_Light\_Rail,

wac\_Number\_of\_Jobs\_in\_Income\_Bracket\_One,

wac\_Number\_of\_Jobs\_in\_Income\_Bracket\_Two,

wac\_Number\_of\_Jobs\_in\_Income\_Bracket\_Three,

wac\_Number\_of\_Manufacturing\_Jobs,

wac\_Number\_of\_Finance\_and\_Insurance\_Jobs,

wac\_Number\_of\_Health\_Care\_and\_Social\_Assitance\_Jobs,

wac\_Number\_of\_Technical\_Services\_Jobs,

wac\_Number\_of\_Retail\_Jobs,

wac\_Number\_of\_Accomodation\_and\_Food\_Services,

rac\_Number\_of\_Jobs\_in\_Income\_Bracket\_One,

rac\_Number\_of\_Jobs\_in\_Income\_Bracket\_Two,

rac\_Number\_of\_Jobs\_in\_Income\_Bracket\_Three,

rac\_Number\_of\_Manufacturing\_Jobs,

rac\_Number\_of\_Finance\_and\_Insurance\_Jobs,

rac\_Number\_of\_Health\_Care\_and\_Social\_Assitance\_Jobs,

rac\_Number\_of\_Technical\_Services\_Jobs,

rac\_Number\_of\_Retail\_Jobs,

rac\_Number\_of\_Accomodation\_and\_Food\_Services,

`1\_Bed\_Target`))

Two\_Bed\_Property <- attribute %>% select(c(Density.of.Dining.and.Beverage.Options,

Density.of.Highways,

Density.of.Historical.Properties,

Density.of.Entertainment.Options,

Density.of.Medical.Facilities,

Density.of.Offices,

Density.of.Parks,

Density.of.Educational.Institutions,

Density.of.Shops,

Density.of.Bus.Stops,

Proximity\_to\_Uptown,

Proximity\_to\_Light\_Rail,

wac\_Number\_of\_Jobs\_in\_Income\_Bracket\_One,

wac\_Number\_of\_Jobs\_in\_Income\_Bracket\_Two,

wac\_Number\_of\_Jobs\_in\_Income\_Bracket\_Three,

wac\_Number\_of\_Manufacturing\_Jobs,

wac\_Number\_of\_Finance\_and\_Insurance\_Jobs,

wac\_Number\_of\_Health\_Care\_and\_Social\_Assitance\_Jobs,

wac\_Number\_of\_Technical\_Services\_Jobs,

wac\_Number\_of\_Retail\_Jobs,

wac\_Number\_of\_Accomodation\_and\_Food\_Services,

rac\_Number\_of\_Jobs\_in\_Income\_Bracket\_One,

rac\_Number\_of\_Jobs\_in\_Income\_Bracket\_Two,

rac\_Number\_of\_Jobs\_in\_Income\_Bracket\_Three,

rac\_Number\_of\_Manufacturing\_Jobs,

rac\_Number\_of\_Finance\_and\_Insurance\_Jobs,

rac\_Number\_of\_Health\_Care\_and\_Social\_Assitance\_Jobs,

rac\_Number\_of\_Technical\_Services\_Jobs,

rac\_Number\_of\_Retail\_Jobs,

rac\_Number\_of\_Accomodation\_and\_Food\_Services,

`2\_Bed\_Target`))

Three\_Bed\_Property <- attribute %>% select(c(Density.of.Dining.and.Beverage.Options,

Density.of.Highways,

Density.of.Historical.Properties,

Density.of.Entertainment.Options,

Density.of.Medical.Facilities,

Density.of.Offices,

Density.of.Parks,

Density.of.Educational.Institutions,

Density.of.Shops,

Density.of.Bus.Stops,

Proximity\_to\_Uptown,

Proximity\_to\_Light\_Rail,

wac\_Number\_of\_Jobs\_in\_Income\_Bracket\_One,

wac\_Number\_of\_Jobs\_in\_Income\_Bracket\_Two,

wac\_Number\_of\_Jobs\_in\_Income\_Bracket\_Three,

wac\_Number\_of\_Manufacturing\_Jobs,

wac\_Number\_of\_Finance\_and\_Insurance\_Jobs,

wac\_Number\_of\_Health\_Care\_and\_Social\_Assitance\_Jobs,

wac\_Number\_of\_Technical\_Services\_Jobs,

wac\_Number\_of\_Retail\_Jobs,

wac\_Number\_of\_Accomodation\_and\_Food\_Services,

rac\_Number\_of\_Jobs\_in\_Income\_Bracket\_One,

rac\_Number\_of\_Jobs\_in\_Income\_Bracket\_Two,

rac\_Number\_of\_Jobs\_in\_Income\_Bracket\_Three,

rac\_Number\_of\_Manufacturing\_Jobs,

rac\_Number\_of\_Finance\_and\_Insurance\_Jobs,

rac\_Number\_of\_Health\_Care\_and\_Social\_Assitance\_Jobs,

rac\_Number\_of\_Technical\_Services\_Jobs,

rac\_Number\_of\_Retail\_Jobs,

rac\_Number\_of\_Accomodation\_and\_Food\_Services,

`3\_Bed\_Target`))

#Dropping NA's from Target Variable

One\_Bed\_Property <- na.omit(One\_Bed\_Property)

Two\_Bed\_Property <- na.omit(Two\_Bed\_Property)

Three\_Bed\_Property <- na.omit(Three\_Bed\_Property)

#Initial Exploration

str(One\_Bed\_Property)

str(Two\_Bed\_Property)

str(Three\_Bed\_Property)

#Changing DV data types

One\_Bed\_Property$`1\_Bed\_Target` <- as.factor(One\_Bed\_Property$`1\_Bed\_Target`)

Two\_Bed\_Property$`2\_Bed\_Target` <- as.factor(Two\_Bed\_Property$`2\_Bed\_Target`)

Three\_Bed\_Property$`3\_Bed\_Target` <- as.factor(Three\_Bed\_Property$`3\_Bed\_Target`)

str(One\_Bed\_Property)

str(Two\_Bed\_Property)

str(Three\_Bed\_Property)

#Create data partition for "1 Bedroom Properties"

set.seed(101)

One\_Bed\_trainIndex <- createDataPartition(One\_Bed\_Property$`1\_Bed\_Target`,

p=0.7,

list=FALSE,

times=1)

#Create data partition for "2 Bedroom Properties"

set.seed(101)

Two\_Bed\_trainIndex <- createDataPartition(Two\_Bed\_Property$`2\_Bed\_Target`,

p=0.7,

list=FALSE,

times=1)

#Create data partition for "3 Bedroom Properties"

set.seed(101)

Three\_Bed\_trainIndex <- createDataPartition(Three\_Bed\_Property$`3\_Bed\_Target`,

p=0.7,

list=FALSE,

times=1)

#Creating Training Data for "1 Bedroom Properties"

One\_Bed.train <- One\_Bed\_Property[One\_Bed\_trainIndex,]

#Creating Validation Data for "1 Bedroom Properties"

One\_Bed.valid <- One\_Bed\_Property[-One\_Bed\_trainIndex,]

#Creating Training Data for "2 Bedroom Properties"

Two\_Bed.train <- Two\_Bed\_Property[Two\_Bed\_trainIndex,]

#Creating Validation Data for "1 Bedroom Properties"

Two\_Bed.valid <- Two\_Bed\_Property[-Two\_Bed\_trainIndex,]

#Creating Training Data for "3 Bedroom Properties"

Three\_Bed.train <- Three\_Bed\_Property[Three\_Bed\_trainIndex,]

#Creating Validation Data for "3 Bedroom Properties"

Three\_Bed.valid <- Three\_Bed\_Property[-Three\_Bed\_trainIndex,]

#Creating a default rfm for "1 Bedroom Properties"

One\_Bed\_rf\_default <- train(`1\_Bed\_Target`~.,

data=One\_Bed.train,

method="rf",

metric="Accuracy",

ntree=100)

print(One\_Bed\_rf\_default)

#More detailed model tuning to search for the best mtry for "1\_Bed\_Properties"

One\_Bed\_tune\_Grid <- expand.grid(.mtry=c(1:17))

One\_Bed\_rf\_mtry <- train(`1\_Bed\_Target`~.,

data=One\_Bed.train,

method="rf",

metric="Accuracy",

tuneGrid=One\_Bed\_tune\_Grid,

ntree=100)

print(One\_Bed\_rf\_mtry)

plot(One\_Bed\_rf\_mtry)

#Evaluate One\_Bed performance

One\_Bed\_prediction <- predict(One\_Bed\_rf\_mtry, One\_Bed.valid)

confusionMatrix(One\_Bed\_prediction, One\_Bed.valid$`1\_Bed\_Target`, positive="1")

# One\_Bed variable importance

varImp(One\_Bed\_rf\_mtry)

#Creating a default rfm for "2 Bedroom Properties"

Two\_Bed\_rf\_default <- train(`2\_Bed\_Target`~.,

data=Two\_Bed.train,

method="rf",

metric="Accuracy",

ntree=100)

print(Two\_Bed\_rf\_default)

#More detailed model tuning to search for the best mtry for "2\_Bed\_Properties"

Two\_Bed\_tune\_Grid <- expand.grid(.mtry=c(1:17))

Two\_Bed\_rf\_mtry <- train(`2\_Bed\_Target`~.,

data=Two\_Bed.train,

method="rf",

metric="Accuracy",

tuneGrid=Two\_Bed\_tune\_Grid,

ntree=100)

#Evaluate Two\_Bed performance

Two\_Bed\_prediction <- predict(Two\_Bed\_rf\_mtry, Two\_Bed.valid)

confusionMatrix(Two\_Bed\_prediction, Two\_Bed.valid$`2\_Bed\_Target`, positive="1")

# Two\_Bed variable importance

varImp(Two\_Bed\_rf\_mtry)

#Creating a default rfm for "3 Bedroom Properties"

Three\_Bed\_rf\_default <- train(`3\_Bed\_Target`~.,

data=Three\_Bed.train,

method="rf",

metric="Accuracy",

ntree=100)

print(Three\_Bed\_rf\_default)

#More detailed model tuning to search for the best mtry for "3\_Bed\_Properties"

Three\_Bed\_tune\_Grid <- expand.grid(.mtry=c(1:17))

Three\_Bed\_rf\_mtry <- train(`3\_Bed\_Target`~.,

data=Three\_Bed.train,

method="rf",

metric="Accuracy",

tuneGrid=Three\_Bed\_tune\_Grid,

ntree=100)

#Evaluate Three\_Bed performance

Three\_Bed\_prediction <- predict(Three\_Bed\_rf\_mtry, Three\_Bed.valid)

confusionMatrix(Three\_Bed\_prediction, Three\_Bed.valid$`3\_Bed\_Target`, positive="1")

# Three\_Bed variable importance

varImp(Three\_Bed\_rf\_mtry)