#### Libraries ####

library(rgdal)

library(rgeos)

library(corrplot)

library(PerformanceAnalytics)

library(ggplot2)

library(MASS)

library(randomForest)

library(VGAM)

library(classInt)

library(RColorBrewer)

### Reading the dataset ####

LondonData <- read.csv("DataScienceProj.csv",stringsAsFactors=FALSE)

View(LondonData)

names(LondonData)

dim(LondonData)

str(LondonData)

summary(LondonData)

### suspicious values for RetiPct, Unemploy

#### Correlation Analysis ####

London\_Numerical <- as.data.frame(LondonData[, c(4,23:31)])

corr <- round(cor(London\_Numerical), 2)

res1 <- cor.mtest(London\_Numerical, conf.level = .95)

res2 <- cor.mtest(London\_Numerical, conf.level = .99)

p.mat <- cor.mtest(London\_Numerical)$p

col <- colorRampPalette(c("#BB4444", "#EE9988", "#FFFFFF", "#77AADD", "#4477AA"))

corrplot(corr, method = "color", col = col(200),

type = "upper", order = "hclust", number.cex = .7,

addCoef.col = "black", # Add coefficient of correlation

tl.col = "black", tl.srt = 90, # Text label color and rotation

# Combine with significance

p.mat = p.mat, sig.level = 0.05, insig = "blank",

# hide correlation coefficient on the principal diagonal

diag = FALSE)

### Analyse houses price with FlorArea

plot(LondonData$FlorArea,LondonData$Purprice/100000,pch=15,cex=0.4,

ylab="Price in Millions",xlab="FloorArea in sq mtr",col= greenmono )

lines(lowess(LondonData$FlorArea,LondonData$Purprice/100000),col="blue")

title("Scatterplot for House Price vs Floor Area")

### Saleunem with NoCarHh (+0.73) and CarspP

plot(LondonData$Saleunem,LondonData$NoCarHh,pch=15,cex=0.4,

ylab="Proportion of households without a car",xlab="Saleunem",col= greenmono )

lines(lowess(LondonData$Saleunem,LondonData$NoCarHh),col="blue")

title("Scatterplot for NoCarHh vs Saleunem")

plot(LondonData$Saleunem,LondonData$CarspP,pch=15,cex=0.4,

ylab="Cars per person in neighborhood",xlab="Saleunem",col= greenmono )

lines(lowess(LondonData$Saleunem,LondonData$CarspP),col="blue")

title("Scatterplot for CarspP vs Saleunem")

### NoCarHh with CarspP

plot(LondonData$NoCarHh,LondonData$CarspP,pch=15,cex=0.4,

ylab="CarspP",xlab="NoCarHh",col= greenmono )

lines(lowess(LondonData$NoCarHh,LondonData$CarspP),col="blue")

title("Scatterplot for CarspP vs NoCarHh")

#### Outlier Analysis ####

###

### Convert dummies to factors - more convenient for modelling

###

Dummy2Factor <- function(mat,lev1="Level1") {

mat <- as.matrix(mat)

factor((mat %\*% (1:ncol(mat))) + 1,

labels = c(lev1, colnames(mat)))

}

Age <- Dummy2Factor(LondonData[,5:9],"PreWW1")

Type <- Dummy2Factor(LondonData[,10:12],"Others")

Garage <- Dummy2Factor(LondonData[,13:14],"HardStnd")

Bedrooms <- Dummy2Factor(LondonData[,18:21],"BedOne")

MyData <- data.frame(LondonData[,c(2:4,15:17,22,23,26)],Age,Type,Garage,Bedrooms)

summary(MyData)

View(MyData)

# Purprice

boxplot(LondonData$Purprice/100000 ,col = redmono , ylab="Price in Millions" )

title("Boxplot for Purprice")

boxplot(LondonData[LondonData$Purprice < 600000,]$Purprice/100000,col = bluefocus, ylab="Price in Millions" )

title("Boxplot for Purprice with outlier removed")

LondonData <- LondonData[LondonData$Purprice < 600000,]

# Garage

ggplot(data = MyData, aes(y = Purprice/100000, x=Garage)) +

geom\_boxplot( )+

labs(title=" Boxplot of House Price ")+

ylab("Price in Millions") + facet\_wrap(~Type)

# Age

ggplot(data = MyData, aes(y = Purprice/100000, x=Age)) +

geom\_boxplot(fill=c(2:7) )+

labs(title=" Boxplot of House Price vs Age")+

ylab("Price in Millions")

# Garage

ggplot(data = MyData, aes(y = Purprice/100000, x=Garage)) +

geom\_boxplot(fill=c(3:5) )+

labs(title=" Boxplot of House Price vs Garage")+

ylab("Price in Millions")

# Type

ggplot(data = MyData, aes(y = Purprice/100000, x=Type)) +

geom\_boxplot(fill=c(5:8) )+

labs(title=" Boxplot of House Price vs Type")+

ylab("Price in Millions")

# Bedrooms

ggplot(data = MyData, aes(y = Purprice/100000, x=Bedrooms)) +

geom\_boxplot(fill=c(3:7) )+

labs(title=" Boxplot of House Price vs Bedrooms")+

ylab("Price in Millions")

# CenHeat

ggplot(data = MyData, aes(y = Purprice/100000, x=as.factor(CenHeat))) +

geom\_boxplot(fill=c(4:5) )+

labs(title=" Boxplot of House Price vs CenHeat")+

ylab("Price in Millions")

# BathTwo

ggplot(data = MyData, aes(y = Purprice/100000, x=as.factor (BathTwo))) +

geom\_boxplot(fill=c(4:7) )+

labs(title=" Boxplot of House Price vs BathTwo")+

ylab("Price in Millions")

# NewPropD

ggplot(data = MyData, aes(y = Purprice/100000, x=as.factor (NewPropD))) +

geom\_boxplot(fill=c(4:5) )+

labs(title=" Boxplot of House Price vs NewPropD")+

ylab("Price in Millions")

# Tenfree

ggplot(data = MyData, aes(y = Purprice/100000, x=as.factor(Tenfree))) +

geom\_boxplot(fill=c(4:5) )+

labs(title=" Boxplot of House Price vs Tenfree")+

ylab("Price in Millions")

#### Analysis Predictor Significance using AIC ####

### Fit models for a single variable and look at AICs

### - model with \*lowest\* AIC is closest to unknown 'true' model

AICs <- rep(NA,10)

Models <- vector("list",10)

Vars <- colnames(MyData)[4:13]

for(i in 1:10) {

Models[[i]] <- lm(formula(paste0("Purprice~",Vars[i])),data=MyData)

AICs[i] <- AIC(Models[[i]])

}

print(AICs)

minAIC <- which.min(AICs)

print(AICs[minAIC])

print(Vars[minAIC])

summary(Models[[minAIC]])

###

### have a look at the differences

###

names(AICs) <- Vars # add names

sAICs <- sort(AICs) # sort into order

print(sAICs)

plot(sAICs,xaxt="n") # plot

axis(1,labels=names(sAICs),at=1:length(Vars),las=2,cex.axis=.75)

for(i in 2:length(Vars)){ # compute differences

cat(paste(names(sAICs)[i],sAICs[i]-sAICs[i-1],"\n"))

}

###

###So the model with the lowest AIC is FlorArea - most variables add a little something

###

model.9v <- lm(Purprice~FlorArea+Bedrooms+Type+BathTwo+Garage+Tenfree+CenHeat+Age+ProfPct,data=MyData)

summary(model.9v) # adj r^2 ~ .56

###

###

delta <- AICs - min(AICs) # differences

w <- exp(-0.5\*delta)/sum(exp(-0.5\*delta)) # probabilitiies

names(AICs) <- Vars # add names

sAICs <- sort(AICs) # sort into order

print(sAICs)

plot(sAICs,xaxt="n") # plot

axis(1,labels=names(sAICs),at=1:length(Vars),las=2,cex.axis=.75)

for(i in 2:length(Vars)){ # compute differences

cat(paste(names(sAICs)[i],sAICs[i]-sAICs[i-1],"\n"))

}

#### Linear Model ####

MyData\_n <- MyData[,c(-1,-2)] # Removing Easting and Northing

fit\_lm\_r <- lm(Purprice ~ . ,data=MyData)

summary(fit\_lm\_r)

plot(fit\_lm\_r)

##############Applying Random Forest Algorithm##########

random\_forest<- randomForest(Purprice~.,data=MyData\_n)

summary(random\_forest)

varImpPlot(random\_forest, pch = 20, main = "Importance of Variables")

mean(random\_forest$rsq)

##Visualising density plots for significant Predictors

ggplot(MyData, aes(Purprice/100000))+ geom\_density(aes(fill=factor(Bedrooms)), alpha=0.7) +

labs(title="Density plot for Bedrooms vs Purprice",

x="PurPrice in Millions",

y="Density",

fill="Bedrooms")

ggplot(MyData, aes(Purprice/100000)) + geom\_density(aes(fill=factor(Type)), alpha=0.8) +

labs(title="Density plot for Type vs Purprice",

x="Purprice in Millions",

y="Density",

fill="Type")

ggplot(MyData, aes(Purprice/100000)) + geom\_density(aes(fill=factor(BathTwo)), alpha=0.8) +

labs(title="Density plot for Bathroom vs Purprice",

x="Purprice in Millions",

y="Density",

fill="Bathroom")

### Spatial Analysis

nClass = 5

Palette <- rev(brewer.pal(nClass,"Spectral"))

Classes <- classIntervals(MyData$Purprice,nClass,"quantile")

Colours <- findColours(Classes,Palette)

ggplot(data = MyData, aes(y = Northing, x=Easting)) +

geom\_point(col=Colours)+

labs(title=" Spatial distribution of data in Greater London ")

### Explore variation by borough

LB <- readOGR(dsn=".",layer="LondonBoroughs",stringsAsFactors=FALSE) # Boroughs

LH <- SpatialPointsDataFrame(MyData[,1:2],MyData) # Houses

proj4string(LH) <- CRS(proj4string(LB)) # copy CRS

plot(LB)

points(LH,pch=16,cex=0.5,col=Colours,ylab="Price in Millions")

box()

title("Spatial distribution of Houses in Greater London")

###

### Add Brough names to data - explore by type and borough

###

LHLB <- over(LH,LB) # spatial join: points first, then polygons

MyData$Borough <- gsub(" London Boro","",LHLB$NAME) # get the borough name

ggplot(data = MyData, aes(x=reorder(Borough,Purprice/100000),y = Purprice/100000 )) +

geom\_boxplot(col=2:35)+

labs(title=" Boxplot for Borough vs Purprice")+ylab("Purprice in Millions") +xlab("Borough")+

coord\_flip()

Boroughs <- names(table(MyData$Borough))

NB <- length(Boroughs)

ggplot(data = MyData, aes(x=reorder(Borough,log(Purprice)),y = log(Purprice) )) +

geom\_boxplot(col=2:35)+

labs(title=" Boxplot for Borough vs Log(Purprice)")+ylab("Log(Purprice)") +xlab("Borough")+

coord\_flip()

ggplot(data = MyData, aes(x=reorder(Borough,log(Purprice)),y = log(Purprice) )) +

geom\_col(col=3)+

labs(title=" Boxplot for Borough vs Log(Purprice) with Bedrooms")+ylab("Log(Purprice)") +

xlab("Borough")+

coord\_flip() + facet\_wrap(~Bedrooms)

###

### Map of Boroughs with names

###

head(LB$NAME)

Bname <- gsub(" London Boro","",LB$NAME)

xy <- coordinates(LB)

quickMap <- function(Var,nClass=10){

require(classInt)

require(RColorBrewer)

Classes <- classIntervals(Var,nClass,method="quantile")

Palette <- brewer.pal(nClass,"Reds")

Colours <- findColours(Classes,Palette)

plot(y)

points(x.sdf2,cex=0.5,pch=16,col=Colours)

}

###

### Borough specific models

###

data.frame(Bname,LB$NAME) # check ordering of names

head(MyData) # and MyData

NB <- length(LB) # number of boroughs

results <- matrix(0,NB,2) # storage for borough legfel coefficients

results

for(i in 1:NB) {

m.x <- lm(Purprice~FlorArea,data=MyData[MyData$Borough == Bname[i],])

results[i,] <- coef(m.x)

}

rownames(results) <- Bname # add in names

Bname

colnames(results) <- c("Intercept","FlorArea")

print(results)

hist(results[,2]) # look at FlorArea coefficient

boxplot(results[,2])

###

### borough levels plots with legend

###

quickMap2 <- function(Var,nClass=5,dp=0,plotNames=FALSE,title){

require(classInt)

require(RColorBrewer)

Classes <- classIntervals(Var,nClass,method="quantile",dataPrecision=dp)

Palette <- brewer.pal(nClass,"Blues")

Colours <- findColours(Classes,Palette)

plot(LB,col=Colours)

legend("bottomright",

legend=names(attr(Colours,"table")),

fill=attr(Colours,"palette"),

cex=0.8,bty="n")

box()

title(title)

if(plotNames) {

xy <- coordinates(LB)

text(xy[,1],xy[,2],Bname,col="black",cex=0.6)

}

}

quickMap2(results[,2],plotNames=TRUE,title="Map of Purprice in Greater London ") # with borough names

###

### Residuals from the model (borough medians)

###

MyData$stdres.9v <- stdres(model.9v)

quickMap2(tapply(MyData$stdres.9v,MyData$Borough,median),plotNames=TRUE,dp=3,

title="Map of Purprice residuals in Greater London ")