BUAN 6356 Project Report

Brooklyn Home Sales Prediction

Group 10

Mohana Krishna Jaladi

Shiyou Li

Jinyu Min

Sandeep Rani

Contents

1. Executive Summary:

For analyzing and visualizing Brooklyn New York Housing Data, we chose 21 significant variables. Based on the data, we came out with how different variables impact the Sale Price. We followed three steps: Data Collection and Cleaning, Data Exploration, Modeling and Data Communication

First, for data collection, we collected the original data from NYC department. Then, we tried to clean the data and came out with important variables. For categorical variables, we performed EDA to find the relationship with Sale Price. For numeric variables, we used different plots to check their significance.

To figure out which variables have more impact on house price, we plot the correlation matrix. Based on the Correlation Matrix, the important numerical variables are gross square feet, total units and resident units.

We also found that the Sale Price intensely varies between different neighborhoods. Therefore, if we want to model Brooklyn home prices accurately, Neighborhood needs to be viewed as the first consideration. The highest among all was Dyker Heights having more than 1000000 USD. The lowest among these was Flat lands having less than 400000 USD. And the most popular neighborhood is Bedford Stuyvesant.

Building Class also has impact on the house price, as the amount of family dwellings increasing, the house price will increase. And the median house price of the family dwellings is higher than the price of the rest of class.

Finally, we chose these numeric and categorical variables to fit three models- Linear Regression, Ridge Regression and Lasso Regression. In terms of the accuracy of the model, we chose the Lasso Regression to predict the house price.

2. Project Motivation

The original data is from the NYC department of Finance (Housing Sales Data). The Department of Finance's Rolling Sales files properties that sold in the last twelve-month period in New York City for all tax classes. We compared different regression models using the same benchmark to decide the best model. Our challenge is "**Can you model Brooklyn home prices accurately?**"

3. Data Description

Brooklyn New York Housing and GIS data describes home sale prices by block in Brooklyn for the last 12 months. Dataset includes 21 variables, which describes housing conditions and features among 23,356 observations.

The following are the variables from the dataset

Borough: The name of the borough in which the property is located.

Neighborhood: Department of Finance assessors determine the neighborhood name while valuing properties. The common name of the neighborhood is generally the same as the name Finance designates. However, there may be slight differences in neighborhood boundary lines and some sub-neighborhoods may not be included.

Building Class Category: This is a field that we are including so that users of the Rolling Sales Files can easily identify similar properties by broad usage (e.g. One Family Homes) without looking up individual Building Classes. Files are sorted by Borough, Neighborhood, Building Class Category, Block and Lot.

Tax Class at Present: Every property in the city is assigned to one of four tax classes (Classes 1, 2, 3, and 4), based on the use of the property.

• Class 1: Includes most residential property of up to three units (such as one-, two-, and three-family homes and small stores or offices with one or two attached apartments), vacant land that is zoned for residential use, and most condominiums that are not more than three stories.

• Class 2: Includes all other property that is primarily residential, such as cooperatives and condominiums.

• Class 3: Includes property with equipment owned by a gas, telephone or electric company.

• Class 4: Includes all other properties not included in class 1,2, and 3, such as offices, factories, warehouses, garage buildings, etc.

Block: A Tax Block is a sub-division of the borough on which real properties are located. The Department of Finance uses a Borough-Block-Lot classification to label all real property in the City. “Whereas” addresses describe the street location of a property, the block and lot distinguish one unit of real property from another, such as the different condominiums in a single building. Also, block and lots are not subject to name changes based on which side of the parcel the building puts its entrance on.

Lot: A Tax Lot is a subdivision of a Tax Block and represents the property unique location.

Easement: An easement is a right, such as a right of way, which allows an entity to make limited use of another’s real property. For example: MTA railroad tracks that run across a portion of another property.

Building Class at Present: The Building Classification is used to describe a property’s constructive use. The first position of the Building Class is a letter that is used to describe a general class of properties (for example “A” signifies one-family homes, “O” signifies office buildings. “R” signifies condominiums). The second position, a number, adds more specific information about the property’s use or construction style (using our previous examples “A0” is a Cape Cod style one family home, “O4” is a tower type office building and “R5” is a commercial condominium unit). The term Building Class used by the Department of Finance is interchangeable with the term Building Code used by the Department of Buildings. See NYC Building Classifications.

Address: The street address of the property as listed on the Sales File. Coop sales include the apartment number in the address field.

Zip Code: The property’s postal code

Residential Units: The number of residential units at the listed property.

Commercial Units: The number of commercial units at the listed property.

Total Units: The total number of units at the listed property.

Land Square Feet: The land area of the property listed in square feet.

Gross Square Feet: The total area of all the floors of a building as measured from the exterior surfaces of the outside walls of the building, including the land area and space within any building or structure on the property.

Year Built: Year the structure on the property was built.

Building Class at Time of Sale: The Building Classification is used to describe a property’s constructive use. The first position of the Building Class is a letter that is used to describe a general class of properties (for example “A” signifies one-family homes, “O” signifies office buildings. “R” signifies condominiums). The second position, a number, adds more specific information about the property’s use or construction style (using our previous examples “A0” is a Cape Cod style one family home, “O4” is a tower type office building and “R5” is a commercial condominium unit). The term Building Class as used by the Department of Finance is interchangeable with the term Building Code as used by the Department of Buildings.

Sales Price: Price paid for the property.

Sale Date: Date the property sold.

Note that the dataset has many transactions having a sale price of $0 or only a nominal amount far less than market value. These are likely property transfers to relatives and are excluded from cleaned dataset for analysis of market prices.

4. Exploratory Data Analysis:

We first wanted to analyze the price range of all the houses and hence created a histogram. It is evident that 1000000$ is the most preferred price budget. The count of houses starts to decrease as the price increase, thus making the plot positively skewed.

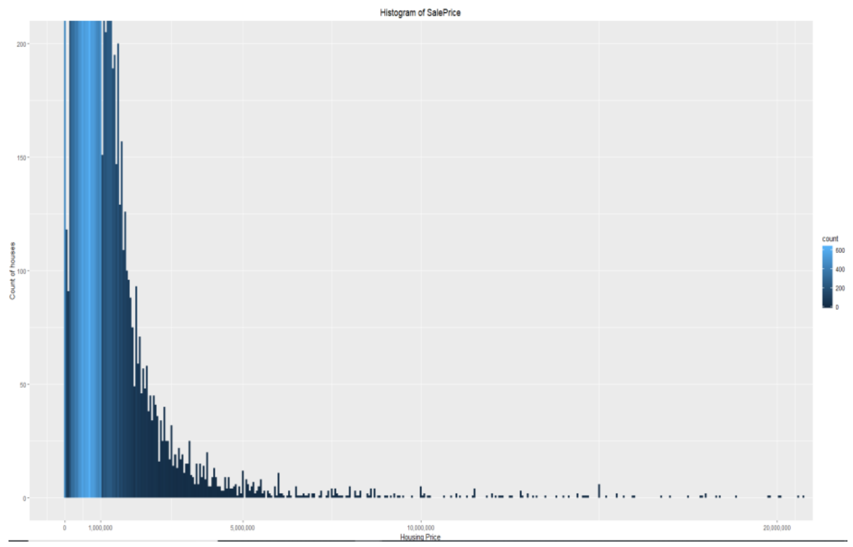


Figure 4.1 saleprice

As the next step, we plotted the correlation matrix to find out the variables influencing the Sales price the most. As seen below, figure 4.2, Total units, Gross square feet and residential units have the highest correlation of all. Apart from this we can see that there is correlation among the independent variables- Gross square feet, Residential units and Total units.

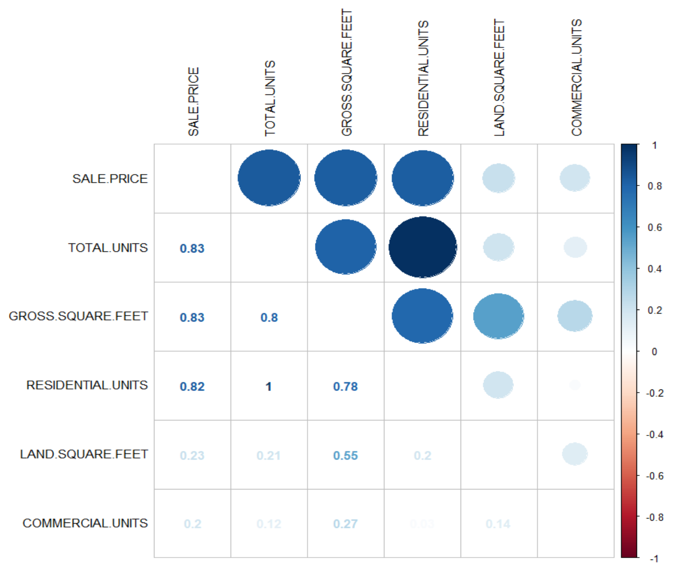


Figure 4.2 correlation matrix

Next, we start by plotting the highly correlated variables with the Sales price- Gross square feet, Total units and Residential units.

1. Gross Square Feet Vs Sale Price:

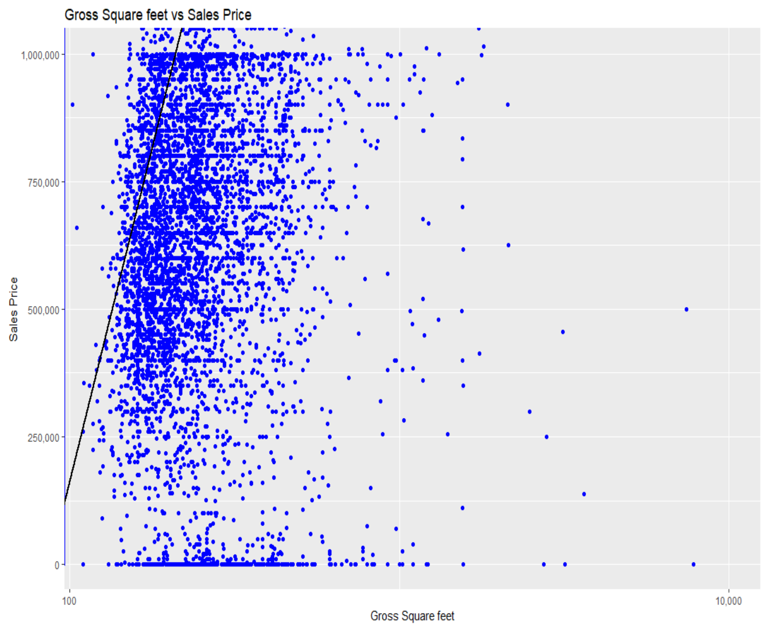


Figure 4.3 gross feet VS sale price

Figure 4.3 gross feet VS sale price: we can see that there is a strong linear relationship with the Sales price. For a 1000000$ house, the house size is mostly within 5000 sq. ft.

2. Total Units Vs Sale Price:

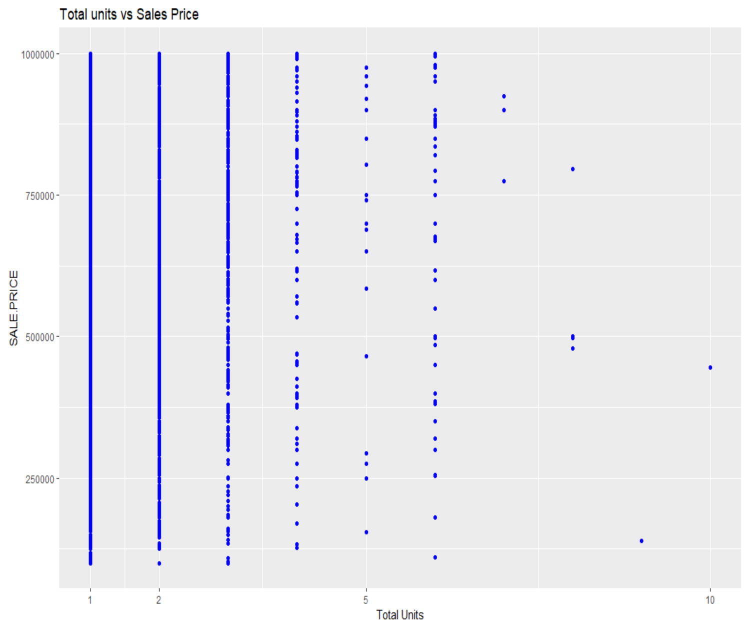


Figure 4.4 Total Units Vs Sale Price

Figure 4.4 Total Units Vs Sale Price: We can clearly see that for total units 1 and 2, the sales price ranges from 100000$ to 1000000$, but for higher total units, there are very few houses in that range.

3. Residential Units Vs Sale Price:

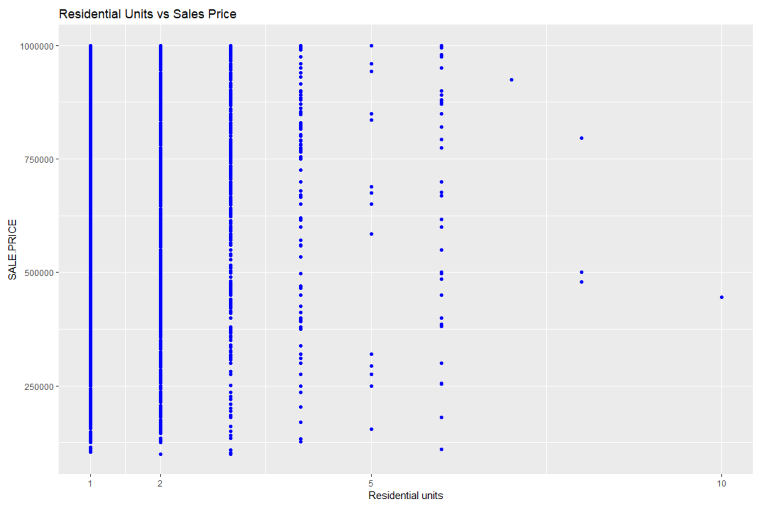


Figure 4.5 Residential Units Vs Sale Price

Figure 4.5 Residential Units Vs Sale Price: Residential units and Total units plots are almost similar since the total units is the sum of residential units and commercial units. Commercial units are almost zero in most of the cases.

IMPACT OF CATEGORICAL VARIABLES:

We have plotted the numeric variables now. There are many categorical variables – Neighborhood, Tax Class, Building Class Category which will surely impact the Sale Price. We will plot these variables now.

4. Impact of Neighborhood on Sale price:

Firstly, we analyze the count of houses in each neighborhood to see which are the most popular and least popular. We created a histogram to plot this.

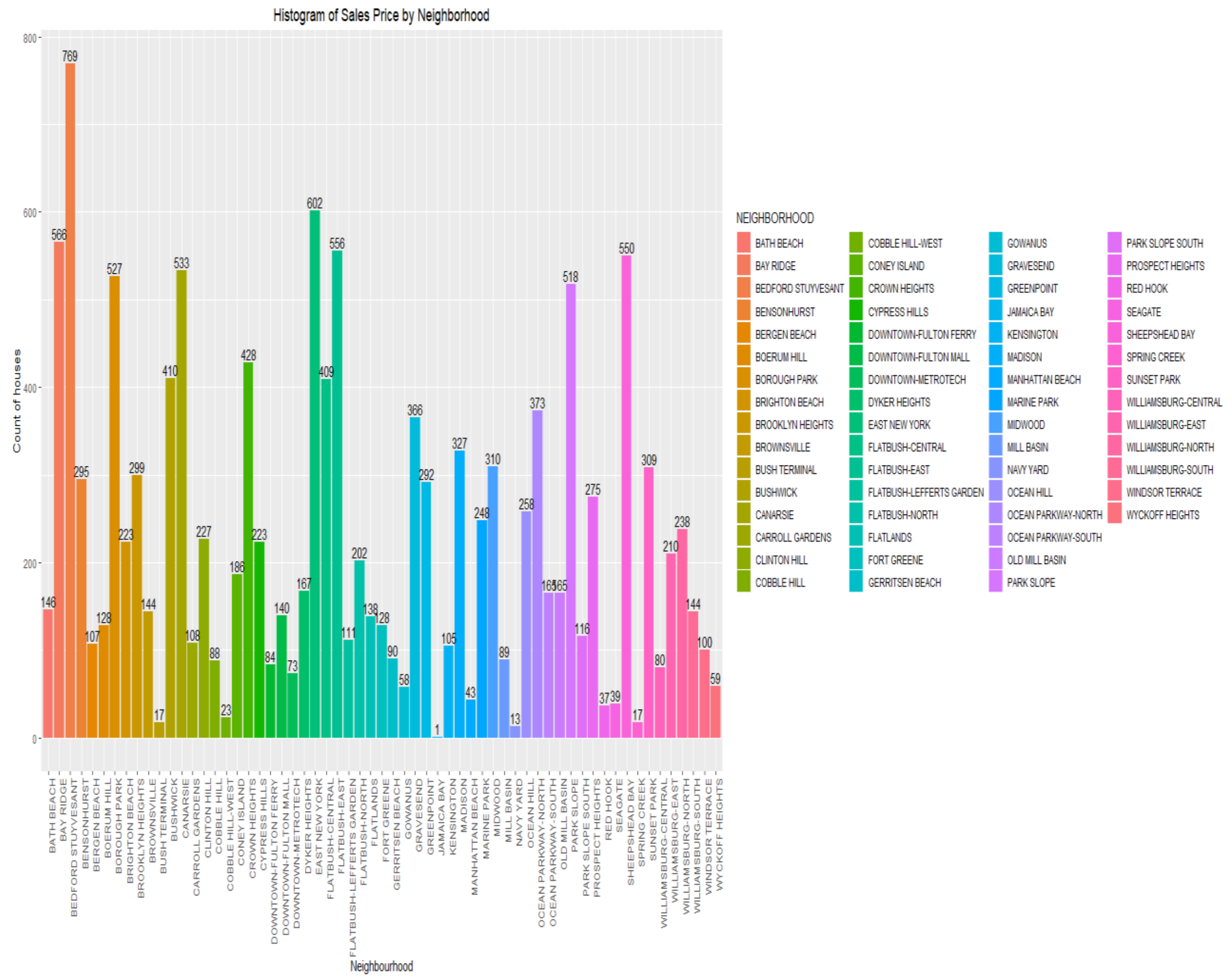


Figure 4.6 house count by Neighborhood

We concluded that the top most popular neighborhoods are as below:

- BEDFORD STUYVESANT
- EAST NEWYORK
- BAY RIDGE
- FLATBUSH CENTRAL
- SHEEPSHEAD BAY

And, the least preferred among these were:

- JAMAICA BAY
- NAVY YARD
- SPRING CREEK

Next, we created a boxplot to analyze the median prices of these neighborhoods and their pricing range.

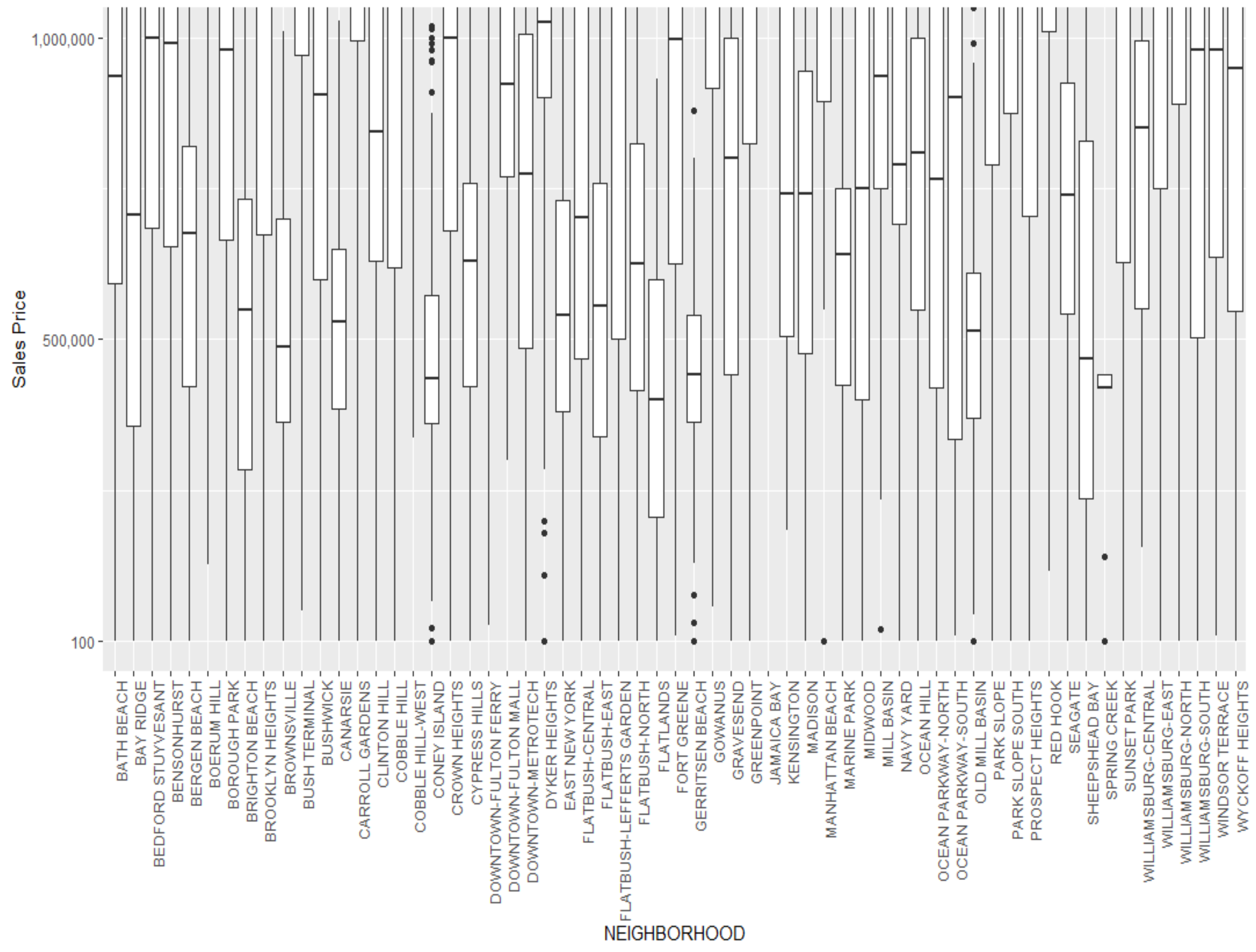


Figure 4.7 sale price by Neighborhood

Clearly, we can infer that the most popular neighborhoods have their median in the higher range. The highest among all was Dyker Heights having more than 1000000 USD. The lowest among these was Flat lands having less than 400000 USD.

5. Impact of Building Class Category on Sale Price:

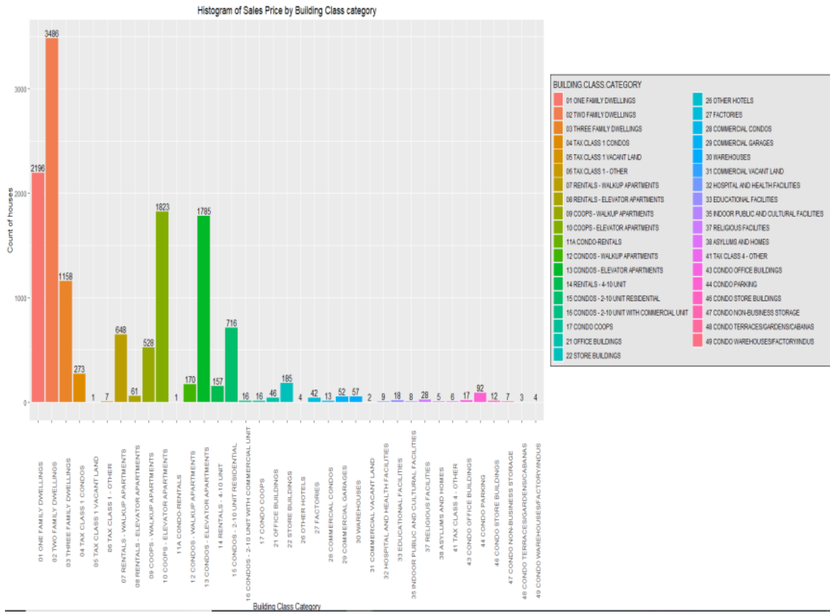


Figure 4.8 house count by Building class category

As we see above, Figure 4.8 house count by Building class category, the most common building class are the one family and two-family dwellings. These were followed by the elevator apartments- both in condos and coops. The three-family dwellings come next but due to the higher price ranges, the count of people preferring it is still less. Again, we drew a box plot to analyze the price range of these categories.

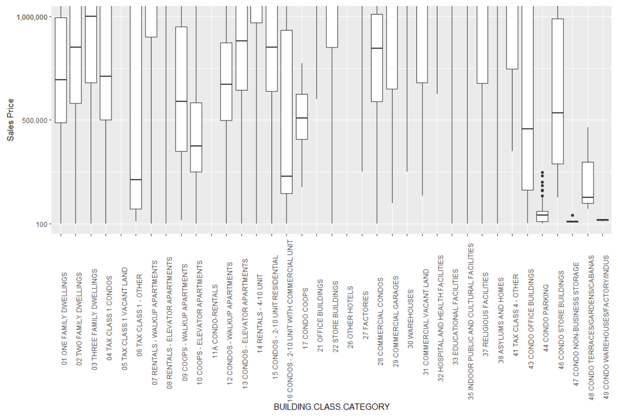


Figure4.9 sale price by Building class category

We can easily see, Figure4.9 sale price by Building class category, that the median prices of the one, two, three family dwellings are in the increasing order with the three family dwellings almost reaching $1000000. The condos-elevator apartments are almost in the same range as the two-family dwellings. An interesting fact is that the coops- elevator apartment is having a low median which is strange considering the number of people preferring it.

6. Impact of Tax Class at Time of Sale on Sale Price:

TAX CLASS 1- MOST RESIDENTIAL UNITS

TAX CLASS 2 – CONDOS AND APARTMENTS

TAX CLASS 4 – OFFICES, FACTORIES, etc.,

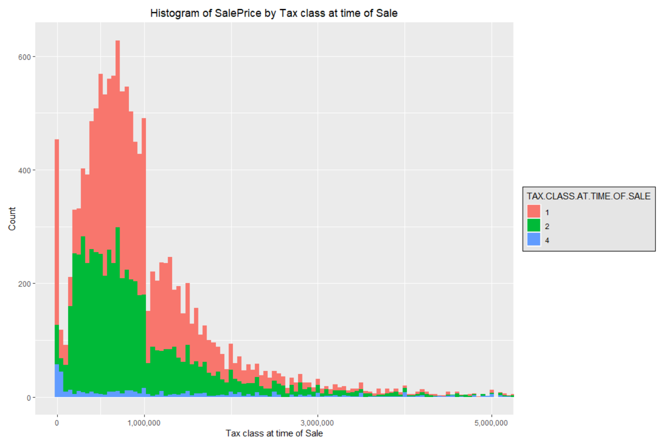


Figure 4.10 count by tax class

Figure 4.10 count by tax class: Tax class 1 with most residential houses (one, two, three family dwellings) have the have the highest number of houses followed by Tax class 2. Tax class 4 which is mainly consisting of commercial units have the least houses understandably.

Next, we created a box plot to understand the price ranges for the different categories.

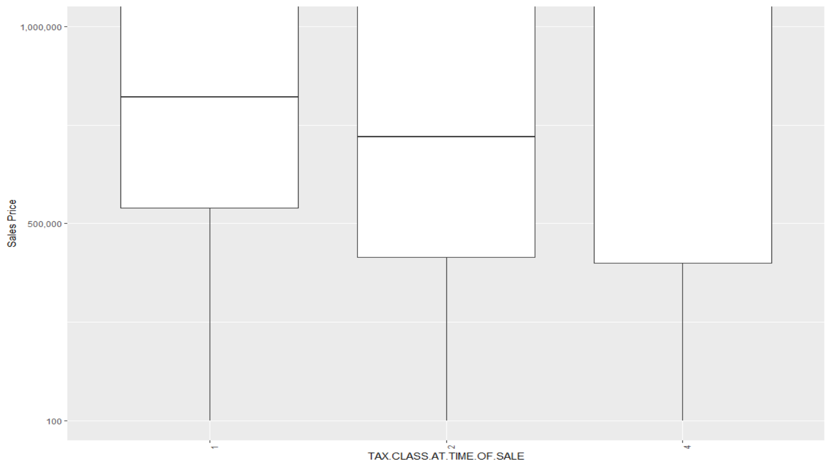


Figure 4.11 sale price by tax class

Figure 4.11 sale price by tax class: class 1 have a bit higher median prices than tax class 2. Tax class 4 clearly have the highest median which is not visible in the plot as the maximum range is 100000$ in the y axis.

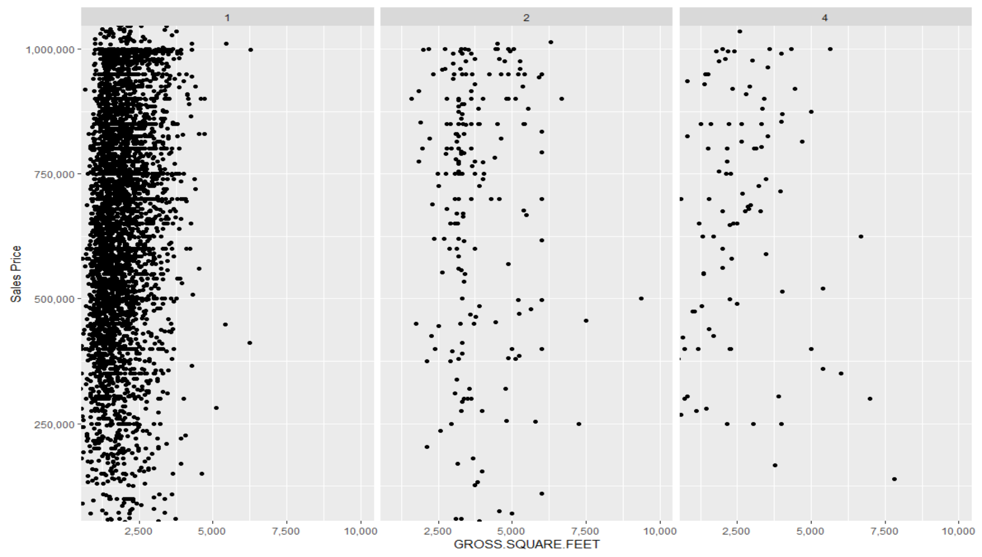


Figure 4.12 gross square feet vs sales price

We created the above plot to see the gross square feet for the different tax classes. Figure 4.12 gross square feet vs sales price.
Tax class 1 - Range mostly from 1000 to 4000 sq. ft, Tax class 2 – Range from 2500 to 6000 sq. ft, Tax class 4- Have different sizes from 1000- 7000 sq. ft.

7. Impact of Year Built on the Sale Price:

Next, we wanted to see if Year built has any impact on the sales price. But from the plot there is no pattern visible. Some of the houses built in 1800 have their average price very high whereas houses built in 1990-2000 have a very low average. Clearly year built doesn't have an impact on the sales price.

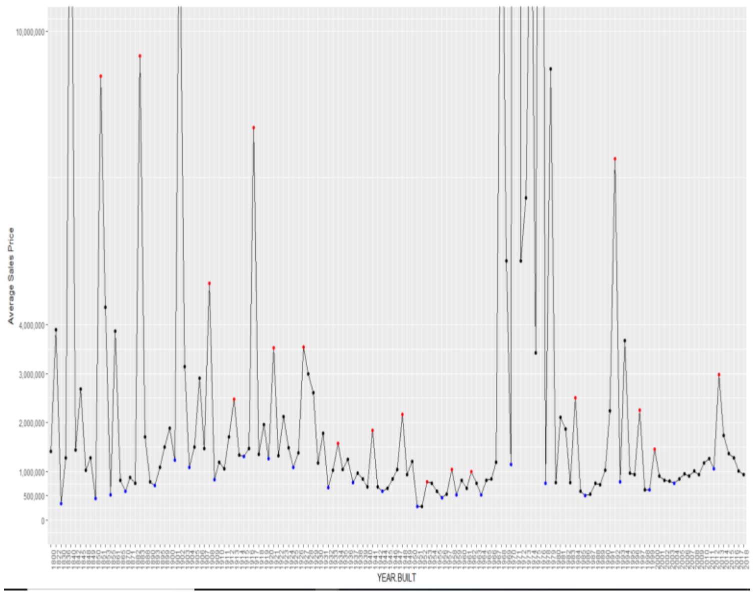


Figure 4.13 average sale price VS year

5. Models and analysis:

We built 4 prediction models for predicting the housing sales price in Brooklyn considering all the mentioned factors above.

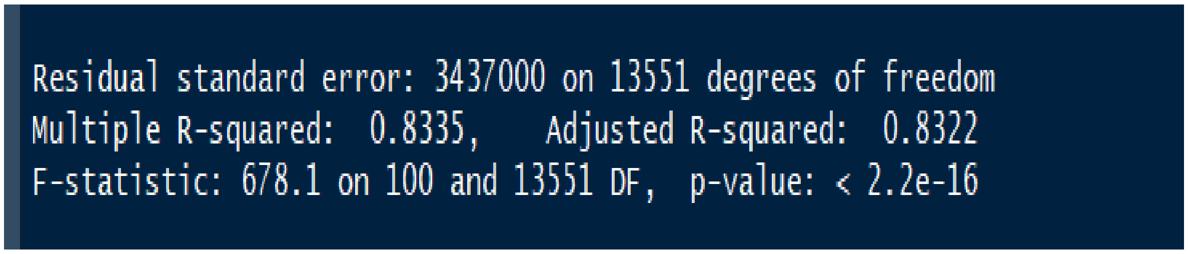
1. Linear Regression without Interaction:

```
mod1<-

SALE.PRICE~TOTAL.UNITS+GROSS.SQUARE.FEET+RESIDENTIAL.UNITS+LAND.SQUARE.FEET+

NEIGHBORHOOD+TAX.CLASS.AT.TIME.OF.SALE+BUILDING.CLASS.CATEGORY, data=data)
```

The above model produced a good result for us which is as below.



```
Residual standard error: 3437000 on 13551 degrees of freedom
Multiple R-squared:  0.8335,    Adjusted R-squared:  0.8322
F-statistic: 678.1 on 100 and 13551 DF,  p-value: < 2.2e-16
```

R-SQUARE - 0.8335

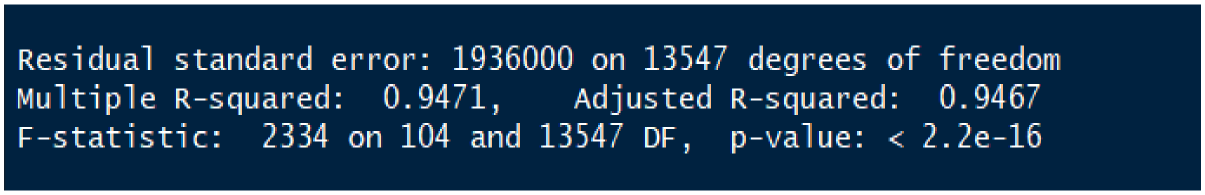
To estimate the test, mean square error, we performed a 10-fold cross validation and obtained the below result.

MSE – 3.394749e+13

2. Linear Regression with Interaction:

```
lmod<-
lm(SALE.PRICE~TOTAL.UNITS*GROSS.SQUARE.FEET*RESIDENTIAL.UNITS+LAND.SQUARE.FEET+NEIGHB
ORHOOD+TAX.CLASS.AT.TIME.OF.SALE+BUILDING.CLASS.CATEGORY,data=data)
```

The above model produced the below result.



```
Residual standard error: 1936000 on 13547 degrees of freedom
Multiple R-squared:  0.9471,    Adjusted R-squared:  0.9467
F-statistic:  2334 on 104 and 13547 DF,  p-value: < 2.2e-16
```

R-SQUARE - 0.9471

To estimate the test, mean square error, we performed a 10-fold cross validation and obtained the below result.

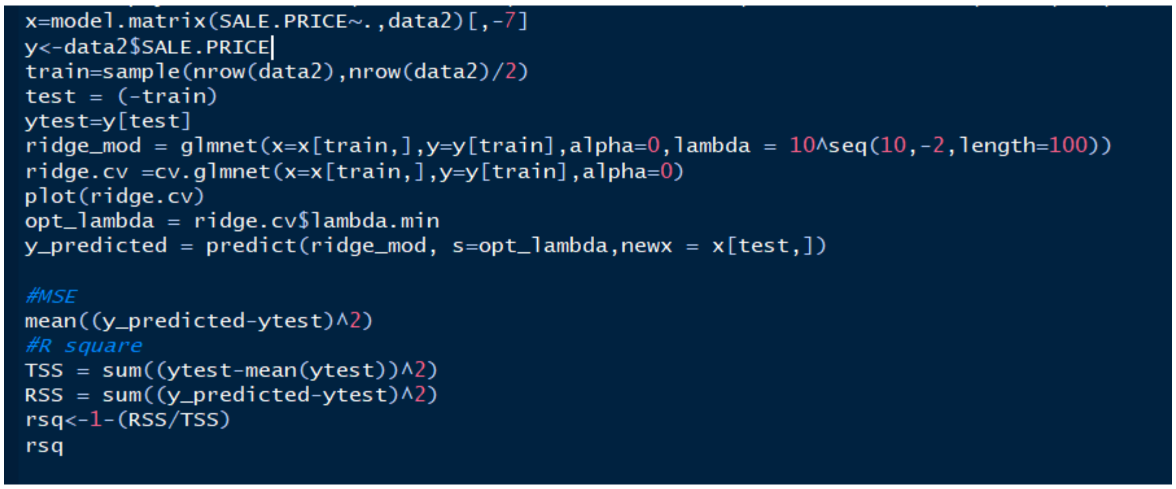
MSE - 2.527223e+13

Clearly, we can see that linear model with interaction produced better R-square and lesser MSE.

After performing linear regression, we wanted to reduce the effect of overfitting and standardize the magnitude of coefficients. Hence, we ran 2 other models- **Ridge and Lasso.**

3. Ridge Regression Model:

We ran the below ridge model. First, we ran a normal ridge model. Then we performed cross validation and chose the best lambda. Then we used this lambda to predict the sales price



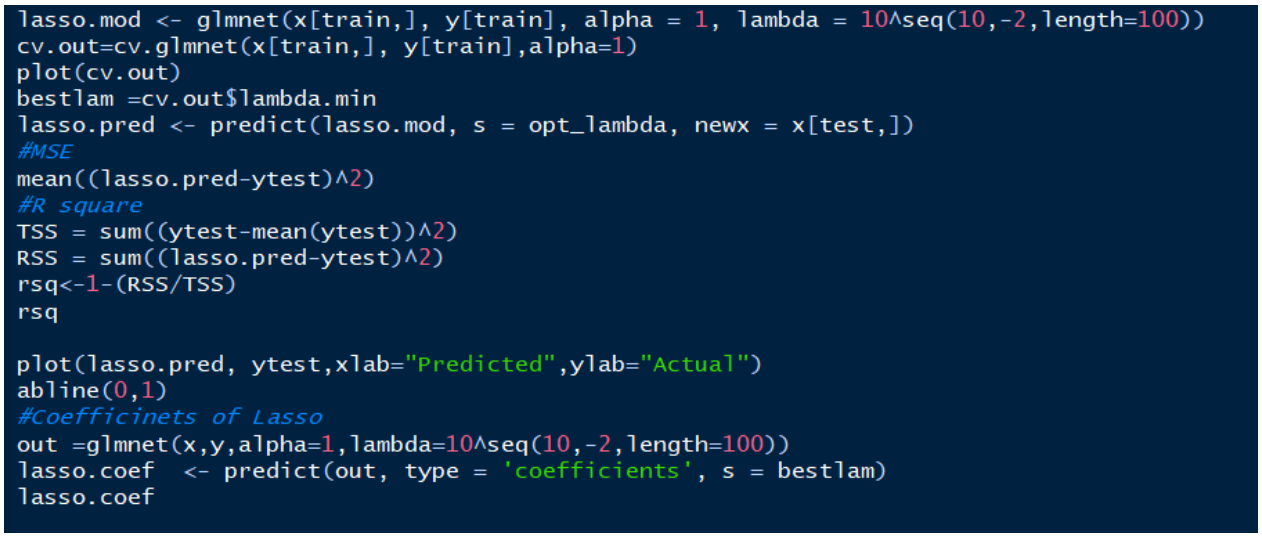
```
x=model.matrix(SALE.PRICE~.,data2)[,-7]
y<-data2$SALE.PRICE
train=sample(nrow(data2),nrow(data2)/2)
test = (-train)
ytest=y[test]
ridge_mod = glmnet(x=x[train,],y=y[train],alpha=0,lambda = 10^seq(10,-2,length=100))
ridge.cv =cv.glmnet(x=x[train,],y=y[train],alpha=0)
plot(ridge.cv)
opt_lambda = ridge.cv$lambda.min
y_predicted = predict(ridge_mod, s=opt_lambda,newx = x[test,])

#MSE
mean((y_predicted-ytest)^2)
#R square
TSS = sum((ytest-mean(ytest))^2)
RSS = sum((y_predicted-ytest)^2)
rsq<-1-(RSS/TSS)
rsq
```

MSE - 1.53405e+13

R-SQUARE - 0.2410698

4. Lasso Regression Model:



```
lasso.mod <- glmnet(x[train,], y[train], alpha = 1, lambda = 10^seq(10,-2,length=100))
cv.out=cv.glmnet(x[train,], y[train],alpha=1)
plot(cv.out)
bestlam =cv.out$lambda.min
lasso.pred <- predict(lasso.mod, s = opt_lambda, newx = x[test,])
#MSE
mean((lasso.pred-ytest)^2)
#R square
TSS = sum((ytest-mean(ytest))^2)
RSS = sum((lasso.pred-ytest)^2)
rsq<-1-(RSS/TSS)
rsq

plot(lasso.pred, ytest,xlab="Predicted",ylab="Actual")
abline(0,1)
#Coefficinets of Lasso
out =glmnet(x,y,alpha=1,lambda=10^seq(10,-2,length=100))
lasso.coef  <- predict(out, type = 'coefficients', s = bestlam)
lasso.coef
```

Like ridge regression, we used cross validation for choosing lambda and then predicting the output.

This produced the below MSE and R-square

```
MSE - 1.03e+13
```

```
R-SQUARE - 0.252665
```

6. Findings and Managerial Implications

- For Linear Regression without Interaction, the R-Square value is 0.83 and the Mean Square Error (MSE) is 3.39e+13.
- For Linear Regression with Interaction, the R-Square value is 0.94 and the Mean Square Error (MSE) is 2.53e+13.
- For Ridge Regression, the R-Square value is 0.24 and the Mean Square Error (MSE) is 1.53e+13.
- For Lasso Regression, the R-Square value is 0.24 and the MSE value is 1.53e+13.

	LINEAR (WITHOUT INTERACTON)	**LINEAR (WITH INTERACTION)**	**RIDGE**	**LASSO**
R-SQUARE	**0.83**	**0.94**	**0.24**	**0.25**
MSE	**3.39e+13**	**2.53e+13**	**1.53e+13**	**1.03e+13**

7. Conclusion

The model is evaluated based on R-Square and MSE values. Considering our Analysis and Evaluation, we chose Lasso Regression as the best fitting prediction model for our Dataset. It has 0.25 R-Square value and its MSE value is 1.03e+13. Though the Linear Regression with Interaction and Linear Regression without Interaction has higher R-Square values i.e., 0.94 and 0.83 respectively, their Mean Square Error values are too high.

Also 0.94 and 0.83 are quite higher R-Square values which result in overfitting. The R-Square value for Lasso Regression (0.25) is good enough to say that Lasso as a significant model. MSE value can be more reliable than the R-Square value to evaluate the Performance of a model. So, we consider Lasso Regression as the best predicting model for our Dataset based on MSE value.

8. Appendix:

```
library(tidyverse)

library(dplyr)

library(modeest)

library(xlsx)

library(ggplot2)

library(corrplot)

library(scales)

library(randomForest)

library(modelr)

library(MASS)

library(leaps)

library(ggpmisc)

library(forecast)

library(glmnet)

library(boot)


#Data cleaning

data<- read.csv('brooklyn.csv')

names(data)

data<-data[,-c(1,7)]

sapply(data,class)

write.csv(data,"cleaned_data.csv",row.names=F)

data<-read.csv("cleaned_data.csv")

data<-filter(data,data$SALE.PRICE>0)

data<-filter(data,data$YEAR.BUILT!=0 )
```

```
data$ZIP.CODE<-as.factor(data$ZIP.CODE)

data$TAX.CLASS.AT.TIME.OF.SALE<-as.factor(data$TAX.CLASS.AT.TIME.OF.SALE)

data$YEAR.BUILT<-as.factor(data$YEAR.BUILT)

sapply(data,class)


#Correlation matrix

numerics <- which(sapply(data, is.numeric))

numericsNames <- names(numerics) #saving names vector for use later on

sales2_num <- data[, numericsNames]

cor_num <- cor(sales2_num, use="pairwise.complete.obs") #correlations of all numeric
variables

corr_sorted <- as.matrix(sort(cor_num[,'SALE.PRICE'], decreasing = TRUE))

corr_sorted

cor_pos <- names(which(apply(corr_sorted, 1, function(x) abs(x)>0.1)))

cor_num <- cor_num[cor_pos, cor_pos]

corrplot.mixed(cor_num, tl.col="black", tl.pos = "lt")


#Explanatory data analysis

#Total units vs Sale price

ggplot(data=data,aes(x=TOTAL.UNITS,y=SALE.PRICE))+

  geom_point(col='blue')+

  scale_x_continuous(breaks=c(1,2,5,10),labels=comma,name = "Total Units")+

  scale_y_continuous(labels = comma,name="Sales Price")+

  coord_cartesian(xlim=c(1,10))+

  ylim(100000,1000000)+

  ggtitle("Total units vs Sales Price")
```

```
#Gross square feet vs Sale price
ggplot(data=data,aes(x=GROSS.SQUARE.FEET,y=SALE.PRICE))+
  geom_point(col='blue')+
  geom_smooth(method='lm',se=FALSE,color='black')+
  scale_x_continuous(breaks=c(100,10000),labels=comma,name="Gross Square feet")+
  scale_y_continuous(labels = comma,name="Sales Price")+
  coord_cartesian(xlim=c(500,10000),ylim=c(1000,1000000))+
  ggtitle("Gross Square feet vs Sales Price")

#Residential units vs Sale price
ggplot(data=data,aes(x=RESIDENTIAL.UNITS,y=SALE.PRICE))+
  geom_point(col='blue')+
  scale_x_continuous(breaks=c(1,2,5,10,50,100),labels=comma,name="Residential units")+
  scale_y_continuous(labels = comma,name="Sales Price")+
  coord_cartesian(xlim=c(1,10))+
  ylim(100000,1000000)+
  ggtitle("Residential Units vs Sales Price")

#Impact of Year built on Sales
avg_sales<-data%>%
  group_by(YEAR.BUILT)%>%
  summarize(avg=mean(SALE.PRICE))
ggplot(data = avg_sales, mapping=aes(x = YEAR.BUILT, y = avg,group=1))+
  geom_point()+
  geom_line()+
  stat_peaks(colour = "red") +
  stat_peaks(geom = "text", colour = "red",
            vjust = -0.5, x.label.fmt = "%Y") +
  stat_valleys(colour = "blue") +
  stat_valleys(geom = "text", colour = "blue", angle = 45,
             vjust = 1.5, hjust = 1,  x.label.fmt = "%Y")+
  coord_cartesian(ylim=c(0,10000000))+
  scale_y_continuous(breaks =
c(0,500000,1000000,2000000,3000000,4000000,10000000),labels = comma,name="Average
Sales Price")+
  theme(axis.text.x = element_text(angle = 90, hjust = 1))+
  ggtitle("Impact of Year Built on Sales")


#Building class category vs Sale price
ggplot(data=data,aes(x=BUILDING.CLASS.CATEGORY,y=SALE.PRICE))+
  geom_boxplot()+
  coord_cartesian(ylim=c(100,1000000))+
  scale_y_continuous(breaks = c(100,500000,1000000,1000000),labels = comma,name="Sales
Price")+
  theme(axis.text.x = element_text(angle = 90, hjust = 1))


#Tax Class vs Sale price
ggplot(data=data,aes(x=TAX.CLASS.AT.TIME.OF.SALE,y=SALE.PRICE))+
  geom_boxplot()+
  coord_cartesian(ylim=c(100,1000000))+
  scale_y_continuous(breaks = c(100,500000,1000000),labels = comma,name="Sales Price")+
  theme(axis.text.x = element_text(angle = 90, hjust = 1))


#Neighbourhood vs Sale price
ggplot(data=data,aes(x=NEIGHBORHOOD,y=SALE.PRICE))+
  geom_boxplot()+
  coord_cartesian(ylim=c(100,1000000))+
  scale_y_continuous(breaks = c(100,500000,1000000),labels = comma,name="Sales Price")+
  theme(axis.text.x = element_text(angle = 90, hjust = 1))

#Zip code vs Sale price
ggplot(data=data,aes(x=ZIP.CODE,y=SALE.PRICE))+
  geom_boxplot()+
  coord_cartesian(ylim=c(0,100000))+
  scale_y_continuous(breaks = c(0,500000,1000000),labels = comma,name="Sales Price")+
  theme(axis.text.x = element_text(angle = 90, hjust = 1))

#Gross square feet vs Sale price based on Tax Clas
attach(data)
ggplot(data=data,aes(x=GROSS.SQUARE.FEET,y=SALE.PRICE))+
  geom_point()+
  facet_grid(.~TAX.CLASS.AT.TIME.OF.SALE)+
  scale_x_continuous(labels=comma)+
  scale_y_continuous(labels = comma,name="Sales Price")+
  coord_cartesian(xlim=c(1000,10000),ylim=c(100000,1000000))

#Histogram of Sale price
options(scipen=10000)
ggplot(data, aes(x = SALE.PRICE, fill = ..count..)) +
  geom_histogram(binwidth = 50000) +
  ggtitle("Histogram of SalePrice") +
  ylab("Count of houses") +
  xlab("Housing Price") +
  coord_cartesian(xlim=c(0,20000000),ylim=c(0,200))+
  scale_x_continuous(breaks=c(1000000,5000000,10000000,20000000),labels=comma)+
  theme(plot.title = element_text(hjust = 0.5))
attach(data)

#Histogram of Sales Price by Neighborhood
ggplot(data, aes(x =NEIGHBORHOOD , fill = NEIGHBORHOOD )) +
  geom_bar()+
  ggtitle("Histogram of Sales Price by Neighborhood") +
  ylab("Count of houses") +
  xlab("Neighbourhood")+
  theme(plot.title = element_text(hjust = 0.5),legend.position="right",axis.text.x =
element_text(angle = 90, hjust = 1))+
  geom_text(stat='count',aes(label=..count..),vjust=-0.25)

#Histogram of Sales Price by Tax Class at time of sale
ggplot(data, aes(x =TAX.CLASS.AT.TIME.OF.SALE , fill = TAX.CLASS.AT.TIME.OF.SALE )) +
  geom_bar()+
  ggtitle("Histogram of Sales Price by Tax Class at time of sale") +
  ylab("Count of houses") +
  xlab("Tax Class at time of Sale")+
  theme(plot.title = element_text(hjust = 0.5),legend.position="right")+
  geom_text(stat='count',aes(label=..count..),vjust=-0.25)
```

```
#Histogram of Sales Price by Building Class category

ggplot(data, aes(x =BUILDING.CLASS.CATEGORY , fill = BUILDING.CLASS.CATEGORY )) +

  geom_bar()+

  ggtitle("Histogram of Sales Price by Building Class category") +

  ylab("Count of houses") +

  xlab("Building Class Category")+

  theme(plot.title = element_text(hjust = 0.5),axis.text.x = element_text(angle = 90, hjust =
1),legend.background = element_rect(fill="grey90",size=0.5, linetype="solid", colour ="black"))+

  geom_text(stat='count',aes(label=..count..),vjust=-0.25)


#Histogram of SalePrice by Tax class at present

ggplot(data, aes(x = SALE.PRICE,fill = TAX.CLASS.AT.PRESENT)) +

  geom_histogram(position = "stack", binwidth = 50000) +

  ggtitle("Histogram of SalePrice by Tax class at present") +

  ylab("Count") +

  xlab("Tax Class at present") +

  coord_cartesian(xlim=c(0,5000000))+

  scale_x_continuous(breaks = c(0,1000000,3000000,5000000),labels = comma)+

  theme(plot.title = element_text(hjust = 0.5),legend.background =
element_rect(fill="grey90",size=0.5, linetype="solid", colour ="black"))


#Histogram of SalePrice by Tax class at time of Sale

ggplot(data, aes(x = SALE.PRICE,fill = TAX.CLASS.AT.TIME.OF.SALE)) +

  geom_histogram(position = "stack", binwidth = 50000) +

  ggtitle("Histogram of SalePrice by Tax class at time of Sale") +

  ylab("Count") +

  xlab("Tax class at time of Sale") +

  coord_cartesian(xlim=c(0,5000000))+
  scale_x_continuous(breaks = c(0,1000000,3000000,5000000),labels = comma)+
  theme(plot.title = element_text(hjust = 0.5),legend.background =
element_rect(fill="grey90",size=0.5, linetype="solid", colour ="black"))
summary(data$TAX.CLASS.AT.TIME.OF.SALE)

#Random forest for variable selection
unique(data$YEAR.BUILT)
set.seed(1)
quick_RF <- randomForest(x=data[1:13652,-c(1,6,7,8,15,17,19,18)],
y=data$SALE.PRICE[1:13652], ntree=100,importance=TRUE)
imp_RF <- importance(quick_RF)
imp_DF <- data.frame(Variables = row.names(imp_RF), MSE = imp_RF[,1])
imp_DF <- imp_DF[order(imp_DF$MSE, decreasing = TRUE),]

ggplot(imp_DF[1:10,], aes(x=reorder(Variables, MSE), y=MSE, fill=MSE)) + geom_bar(stat =
'identity') +
  labs(x = 'Variables', y= '% increase in MSE',title="Variable Importance") +
  coord_flip() + theme(legend.position="none")

#Linear models
#Forward selection
#With interaction
leaps<-
regsubsets(SALE.PRICE~TOTAL.UNITS*GROSS.SQUARE.FEET*RESIDENTIAL.UNITS+LAND.SQUAR
E.FEET,data=data,nvmax = 10,method = "forward")
summary(leaps)
regsummary =summary(leaps)
regsummary$rsq
regsummary$adjr2
```

```
plot(regsummary$rsq,xlab="Number of variables", ylab="RSQ",type="l")

plot(regsummary$adjr2,xlab="Number of variables", ylab="Adjusted RSQ",type="l")

plot(regsummary$cp,xlab="Number of variables", ylab="Cp",type="l")

plot(regsummary$bic,xlab="Number of variables", ylab="bic",type="l")


plot(leaps,scale ="r2")

plot(leaps,scale ="adjr2")

plot(leaps,scale ="Cp")

plot(leaps,scale ="bic")

points(8, regsummary$adjr2[which.max(regsummary$adjr2)],col='red')


#Without interaction

leaps<-
regsubsets(SALE.PRICE~TOTAL.UNITS+GROSS.SQUARE.FEET+RESIDENTIAL.UNITS+LAND.SQUAR
E.FEET,data=data,nvmax = 10,method = "forward")

summary(leaps)

regsummary =summary(leaps)

regsummary$rsq

regsummary$adjr2

plot(regsummary$rsq,xlab="Number of variables", ylab="RSQ",type="l")

plot(regsummary$adjr2,xlab="Number of variables", ylab="Adjusted RSQ",type="l")

plot(regsummary$cp,xlab="Number of variables", ylab="Cp",type="l")

plot(regsummary$bic,xlab="Number of variables", ylab="bic",type="l")


#Linear regression

mod<-
glm(SALE.PRICE~TOTAL.UNITS*GROSS.SQUARE.FEET*RESIDENTIAL.UNITS+LAND.SQUARE.FEET,
data=data)
```

```
mod1<-
glm(SALE.PRICE~TOTAL.UNITS+GROSS.SQUARE.FEET+RESIDENTIAL.UNITS+LAND.SQUARE.FEET,
data=data)

cv.error = cv.glm(data,mod,K=10)

cv.error$delta

cv.error1 = cv.glm(data,mod1,K=10)

cv.error1$delta

lmod<-
lm(SALE.PRICE~TOTAL.UNITS*GROSS.SQUARE.FEET*RESIDENTIAL.UNITS+LAND.SQUARE.FEET,d
ata=data)

lmod1<-
lm(SALE.PRICE~TOTAL.UNITS+GROSS.SQUARE.FEET+RESIDENTIAL.UNITS+LAND.SQUARE.FEET,d
ata=data)

summary(lmod)

summary(mod1)


#Ridge regression

set.seed(1)

data2<-
dplyr::select(data,TOTAL.UNITS,GROSS.SQUARE.FEET,RESIDENTIAL.UNITS,BLOCK,LOT,LAND.SQ
UARE.FEET,SALE.PRICE)

x=model.matrix(SALE.PRICE~.,data2)[,-7]

y<-data2$SALE.PRICE

train=sample(nrow(data2),nrow(data2)/2)

test = (-train)

ytest=y[test]

ridge_mod = glmnet(x=x[train,],y=y[train],alpha=0,lambda = 10^seq(10,-2,length=100))

ridge.cv =cv.glmnet(x=x[train,],y=y[train],alpha=0)

plot(ridge.cv)

opt_lambda = ridge.cv$lambda.min
```

```
y_predicted = predict(ridge_mod, s=opt_lambda,newx = x[test,])


#MSE
mean((y_predicted-ytest)^2)
#R square
TSS = sum((ytest-mean(ytest))^2)
RSS = sum((y_predicted-ytest)^2)
rsq<-1-(RSS/TSS)
rsq


plot(y_predicted, ytest,xlab="Predicted",ylab="Actual")
abline(0,1)
#Coefficients of Ridge
out =glmnet(x,y,alpha=0,lambda=10^seq(10,-2,length=100))
lasso.coef  <- predict(out, type = 'coefficients', s = opt_lambda)
lasso.coef


#Lasso
set.seed(1)
lasso.mod <- glmnet(x[train,], y[train], alpha = 1, lambda = 10^seq(10,-2,length=100))
cv.out=cv.glmnet(x[train,], y[train],alpha=1)
plot(cv.out)
bestlam =cv.out$lambda.min
lasso.pred <- predict(lasso.mod, s = opt_lambda, newx = x[test,])
#MSE
mean((lasso.pred-ytest)^2)
#R square
```

```
TSS = sum((ytest-mean(ytest))^2)

RSS = sum((lasso.pred-ytest)^2)

rsq<-1-(RSS/TSS)

rsq


plot(lasso.pred, ytest,xlab="Predicted",ylab="Actual")

abline(0,1)

#Coefficinets of Lasso

out =glmnet(x,y,alpha=1,lambda=10^seq(10,-2,length=100))

lasso.coef  <- predict(out, type = 'coefficients', s = bestlam)

lasso.coef
```

9. References:

https://www.kaggle.com/tianhwu/brooklynhomes2003to2017

https://www1.nyc.gov/site/finance/taxes/property-rolling-sales-data.page

https://www1.nyc.gov/site/finance/taxes/property-annualized-sales-update.page

https://www1.nyc.gov/assets/finance/downloads/pdf/07pdf/glossary_rsf071607.pdf

https://www1.nyc.gov/assets/finance/jump/hlpbldgcode.html