HOUSE PRICE PREDICTION

CAPSTONE PROJECT

FINAL BUSINESS REPORT

Done By

HARIHARAN MANICKAM

I. Introduction

The value of a house is dependent on various factors. Factors such as Plot Size, Interior space, no. of bedrooms & bathrooms, and many others. For example, a house on the beach/lakefront will most likely be valued higher than a house with no beach/lakefront.

Our goal is to create a machine learning model which will take as input the various factors affecting the value of a home and predict the value of the home.

Unless a local from the same area, many homebuyers or sellers find brokers to sell their house. Since middlemen are involved, there is less transparency & the price in the market can be manipulated by brokers. The lack of transparency also does not attract Foreign Institutional Investors to invest in the Indian housing market.

A Machine learning model will help buyers & sellers accurately predict a price range. It will help prevent price manipulation by middlemen. It will enable the government to oversee the transactions happening in the housing market and helping to prevent fraudulent transactions and punish violators. The transparency and good governing by the government helps attract Foreign Direct Investments.

II. EDA & Business Implication

Upon initial exploration of the dataset, we have found it to contain data for ***21613 transactions(rows)*** with ***23 features(columns).*** The transactions were from the years ***2014*** & ***2015***.

NOTE: *All variables were cleaned and treated for outliers. All the plots have been made using cleaned and pre-processed data. Cleaning & Pre Processing has been clearly explained in the Data Cleaning & Data Pre-processing section.*

i) **cid**

We observed that **cid** was a notation for a house. This variable would not be of any help in our model. Hence, we did not perform any univariate/bivariate analysis. We will remove this column during Data Cleaning.

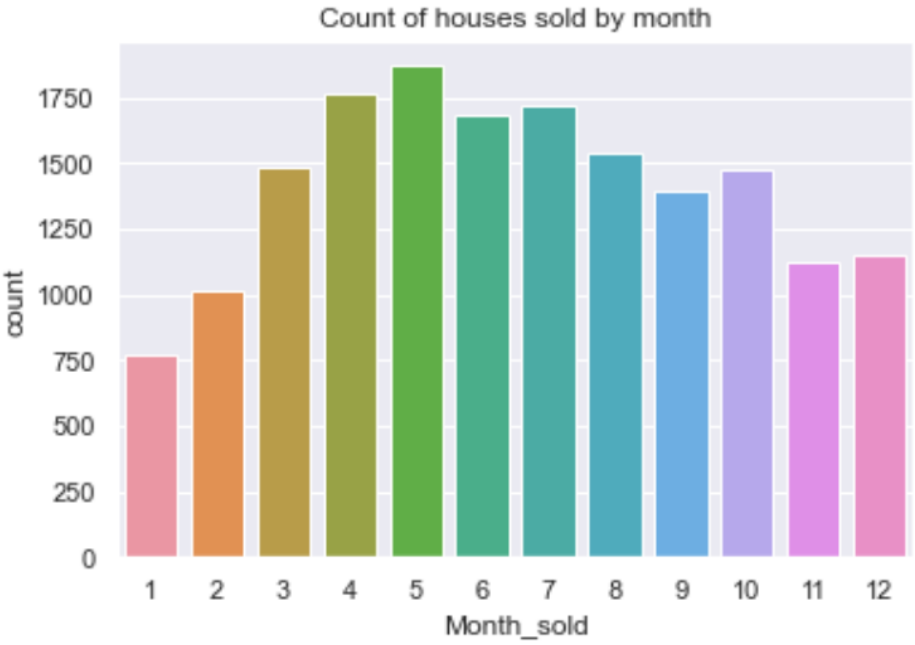
ii) **Dayhours**

We observed that **dayhours** contain the information of the day, month & year of the sale date. We have extracted the ***month*** and ***day*** and created 2 new variables.

Below are the count plots for the same.

Month_sold

(Diagram 1)



Observations:

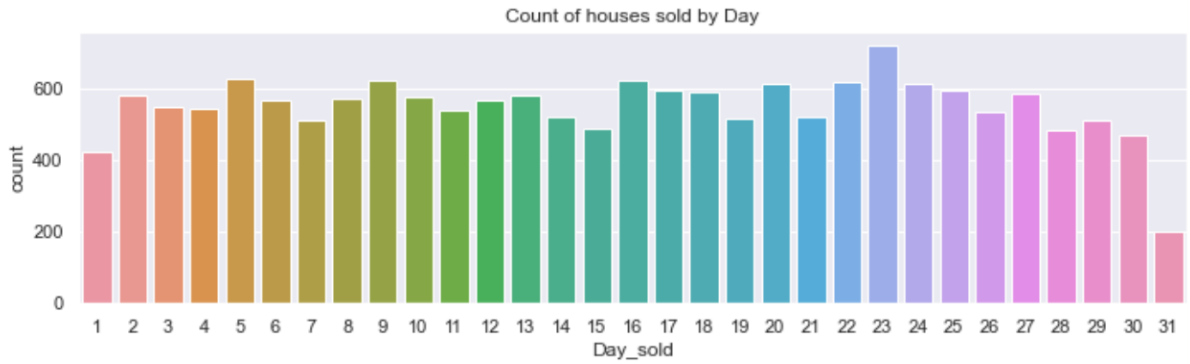
- Highest sales occur in the months of March & October.

Implication:

- There is a higher possibility of getting a better deal when the housing market is active rather when the housing market is inactive.

Day_sold

(Diagram 2)



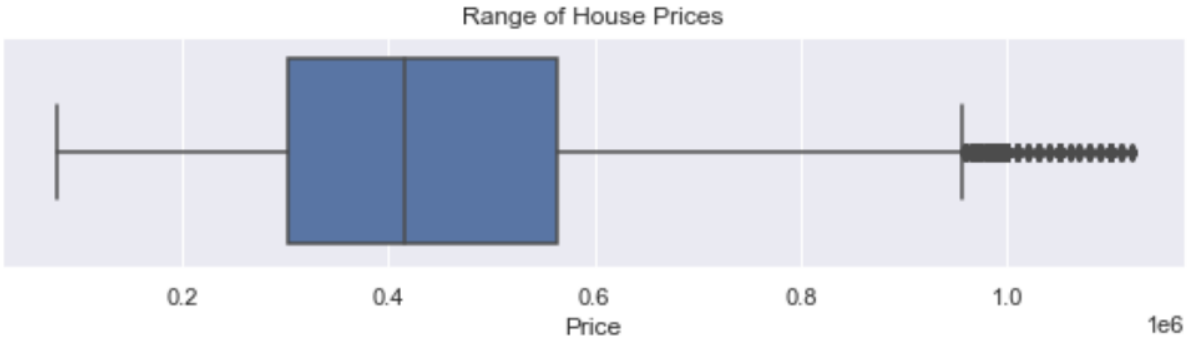
Observations:

- No visual trend.

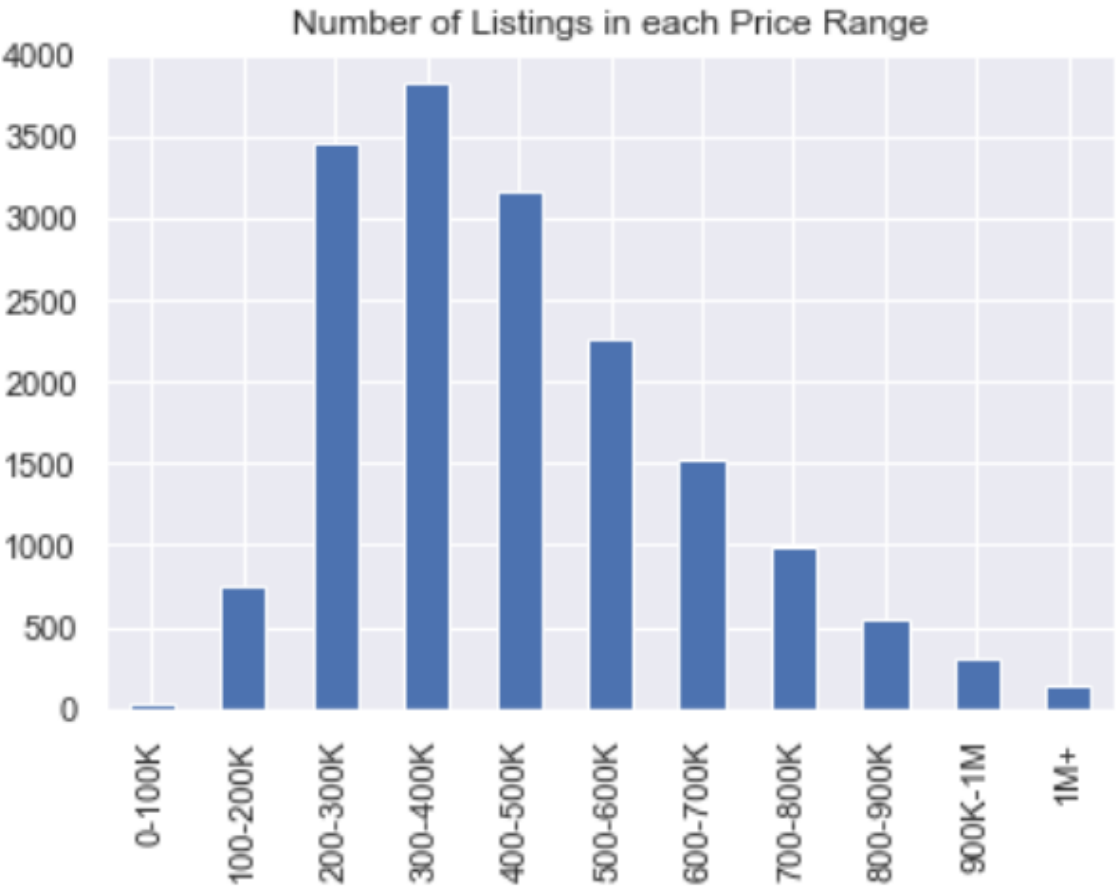
iii) Price

This is our target variable which we would like to predict using a model. Below are different plots made using price.

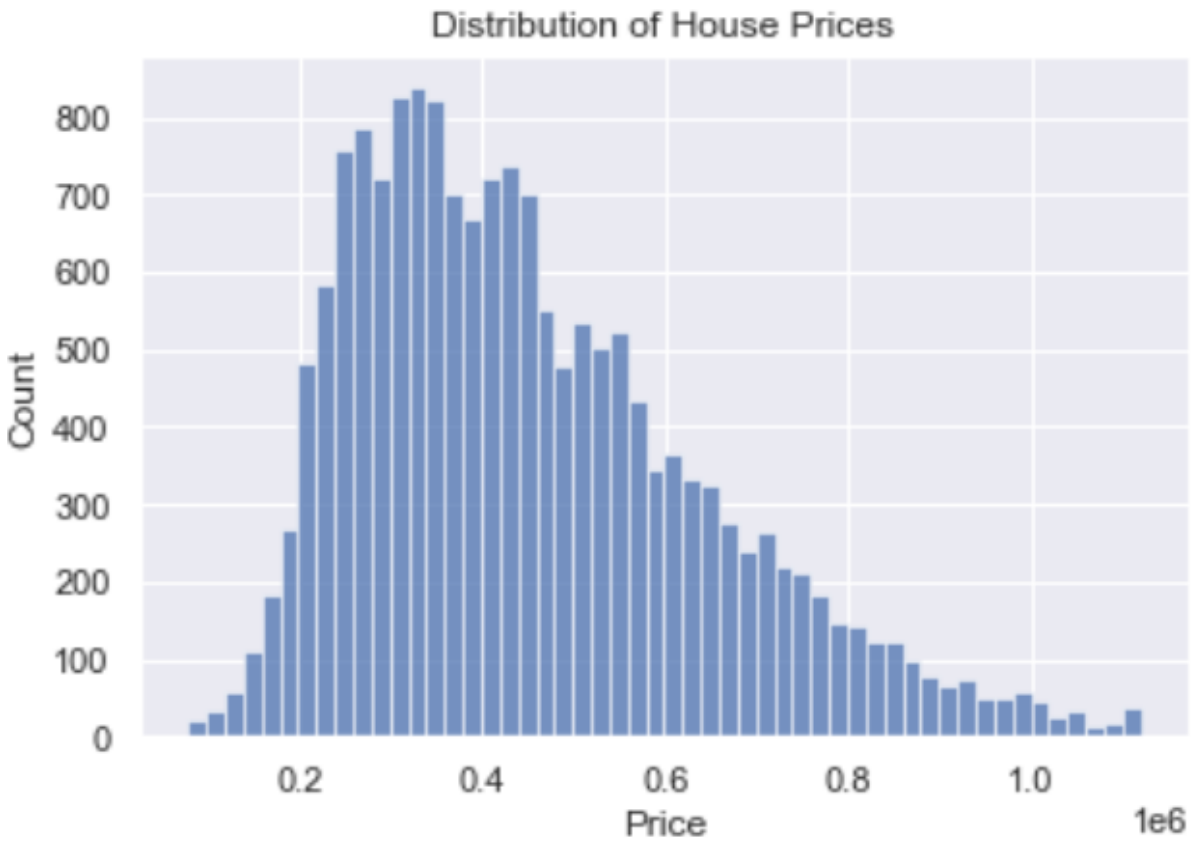
(Diagram 3)



(Diagram 4)



(Diagram 5)



Observations:

- 50% of houses are priced between 300K & 580K (boxplot).
- Most houses are sold between 200K & 600k. (distributed in 100k buckets).
- The distribution of price is slightly skewed to the right. This may require transformation at a later point.

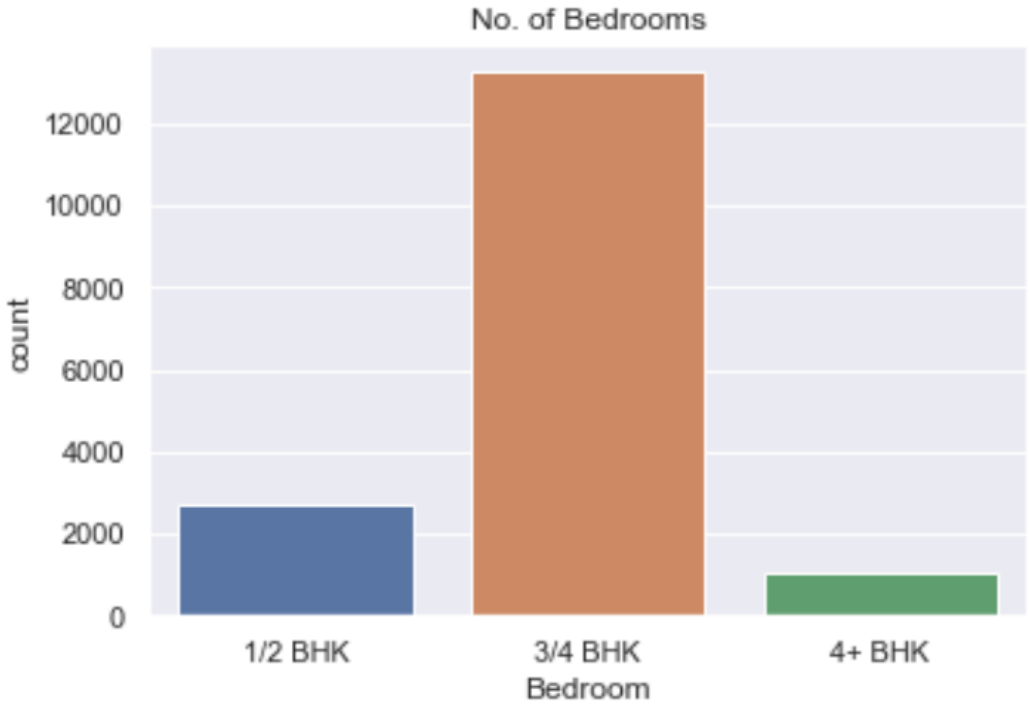
Implication:

- Chance of selling a house is very high when the price is between 200k & 600K.

iv) **Bedrooms**

This variable showed us how many bedrooms the unit had at the time of sale. I have grouped this variable into 3 groups (1/2 BHK, 3/4BHK, 4+BHK).

(Diagram 6)



Observation:

- Majority units have 3/4 Bedrooms.

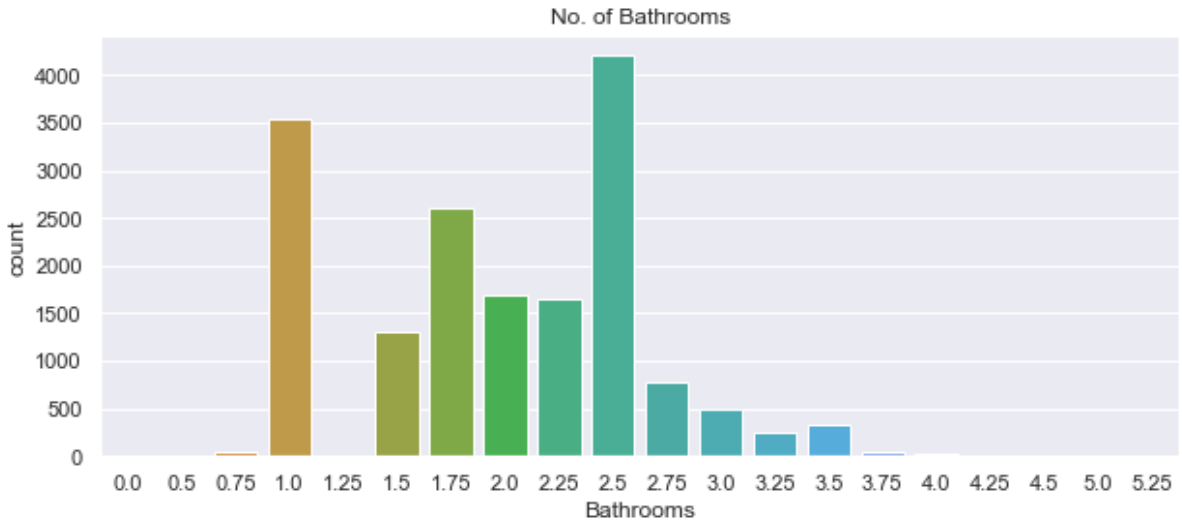
Implication:

- Chance of selling a house is very high when the unit has 3-4 Bedrooms.

v) Bathrooms

This variable shows us how many bathrooms the unit has.

(Diagram 7)



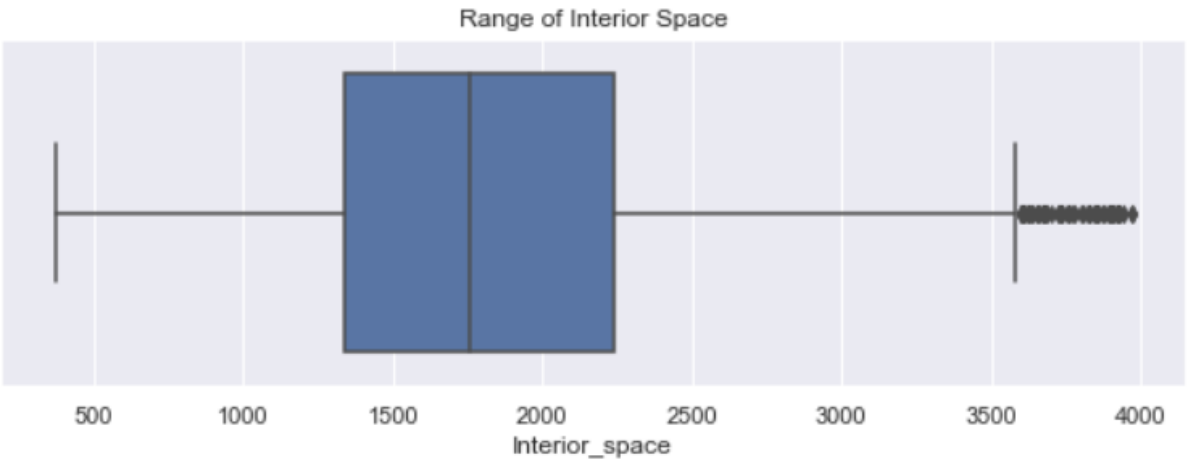
Observation:

- Most units have 1 or 2.5 Bathrooms.

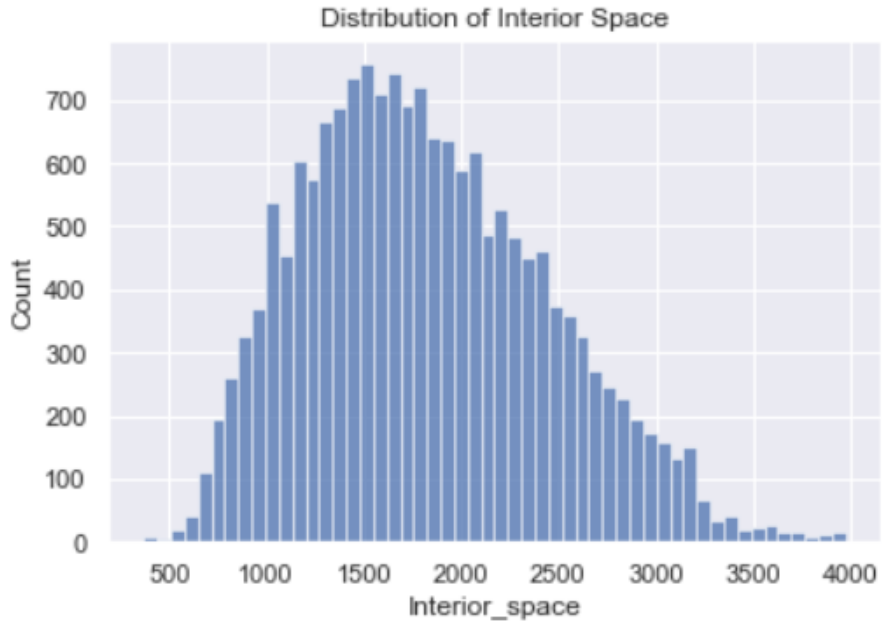
vi) Interior Space

This variable is the measure of interior space of the unit in square ft.

(Diagram 8)



(Diagram 9)



Observations:

- 50% of houses fall between 1300sqft & 2300sqft
- The distribution seems to be normally distributed, hence does not require transformation

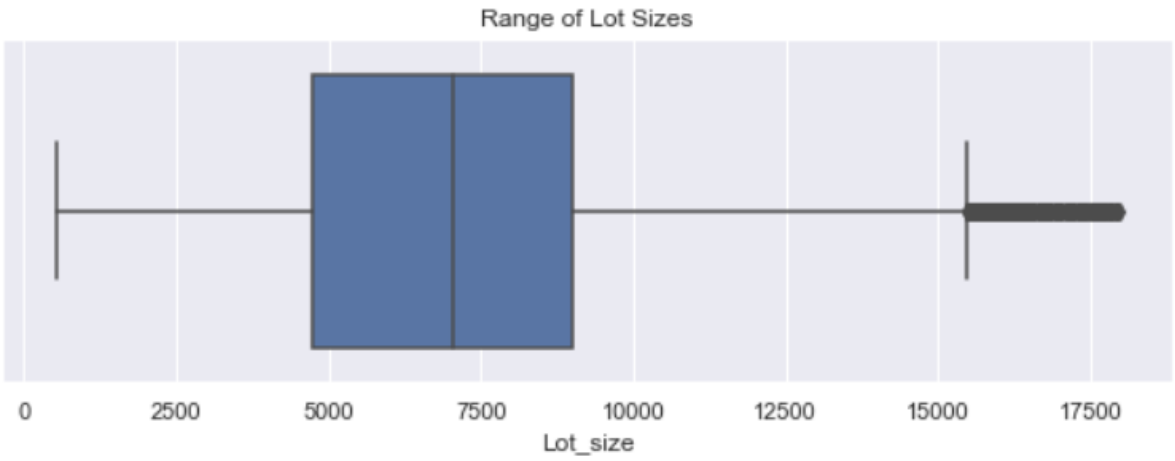
Implication:

- We can sort of assess what size of house will be a good fit for the market

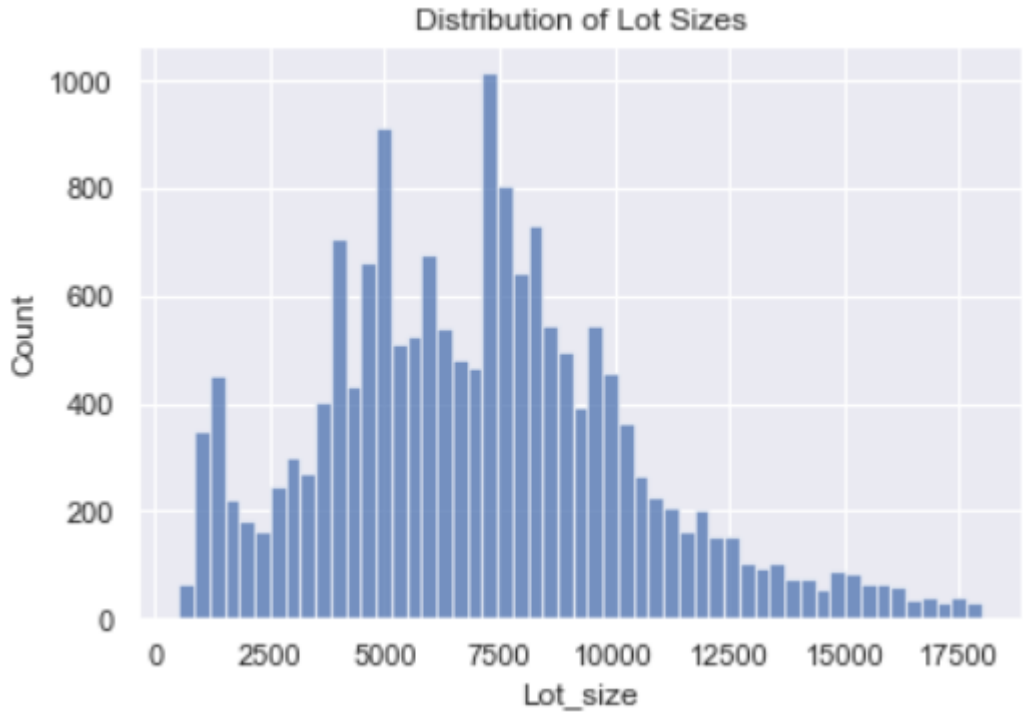
vii) Lot Size

This variable is the measure of Lot size in square ft.

(Diagram 10)



(Diagram 11)



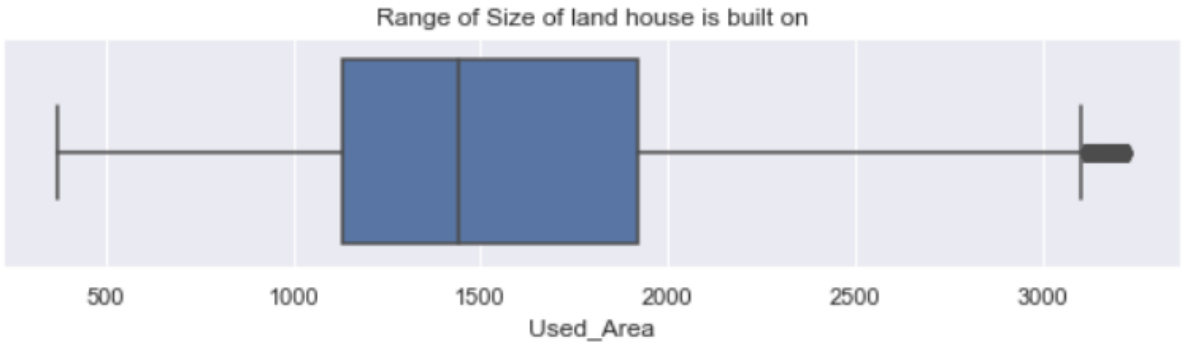
Observations:

- 50% of the lot sizes fall between 4800sqft & 9000sqft
- No pattern present in the distribution.

viii) Used Area

This variable is the square footage used on the full plot to build the house.

(Diagram 12)



(Diagram 13)

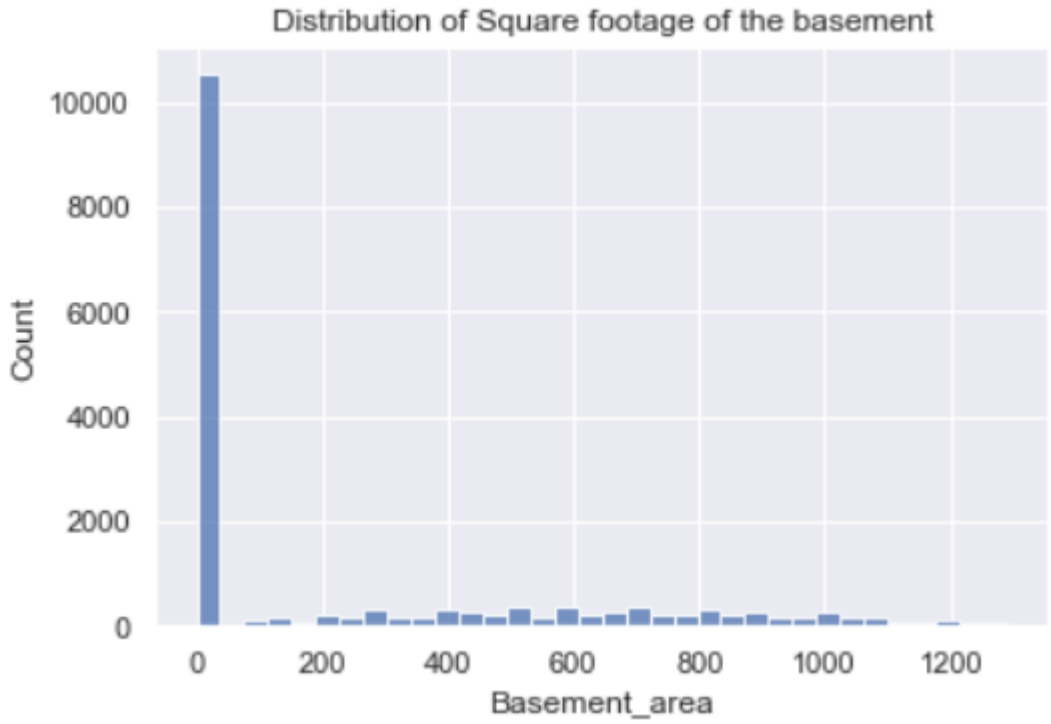
Observations:

- 50% of the Used area size fall between 1100sqft & 1900sqft
- Distribution looks to be slightly skewed to the right. We ca check if transformation is required at a later point.

ix) Basement Area

This variable is the square footage of the basement of the house.

(Diagram 14)

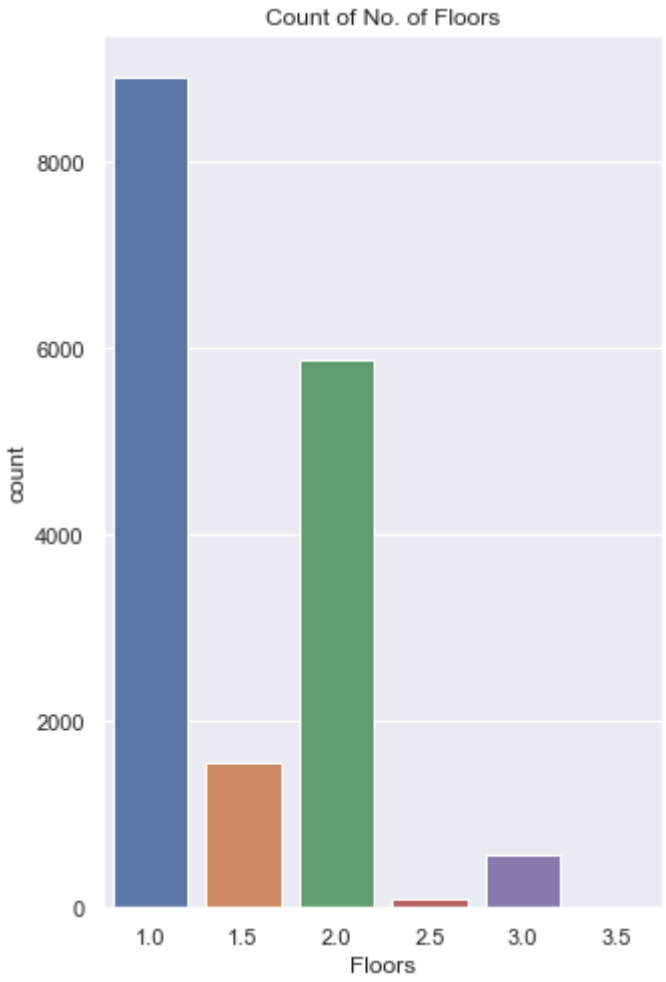


Most houses do not have a basement. Hence, we will remove it during data cleaning.

x) No. of Floors

This is the variable which shows us the no of floors a unit has.

(Diagram 15)



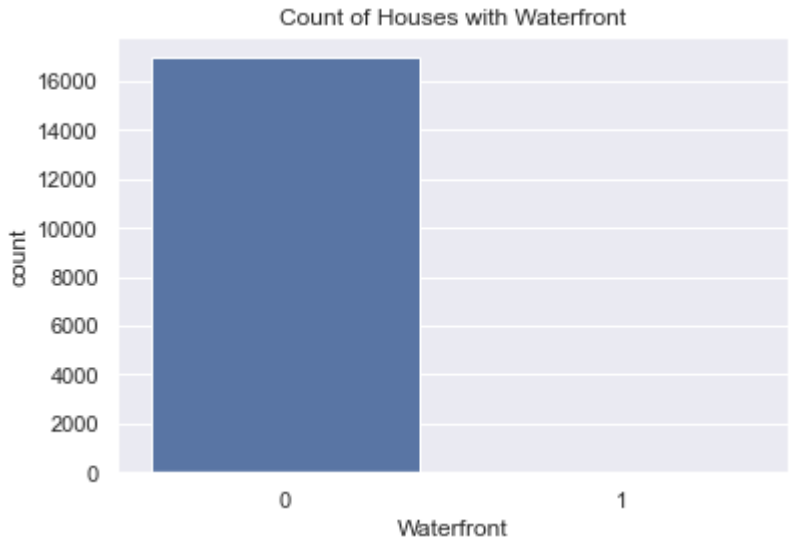
Observations:

- More than 50% of buildings has only a single floor
- Almost 30% of buildings have 2 floors

xi) Waterfront

This variable shows us which houses have a waterfront or not

(Diagram 16)



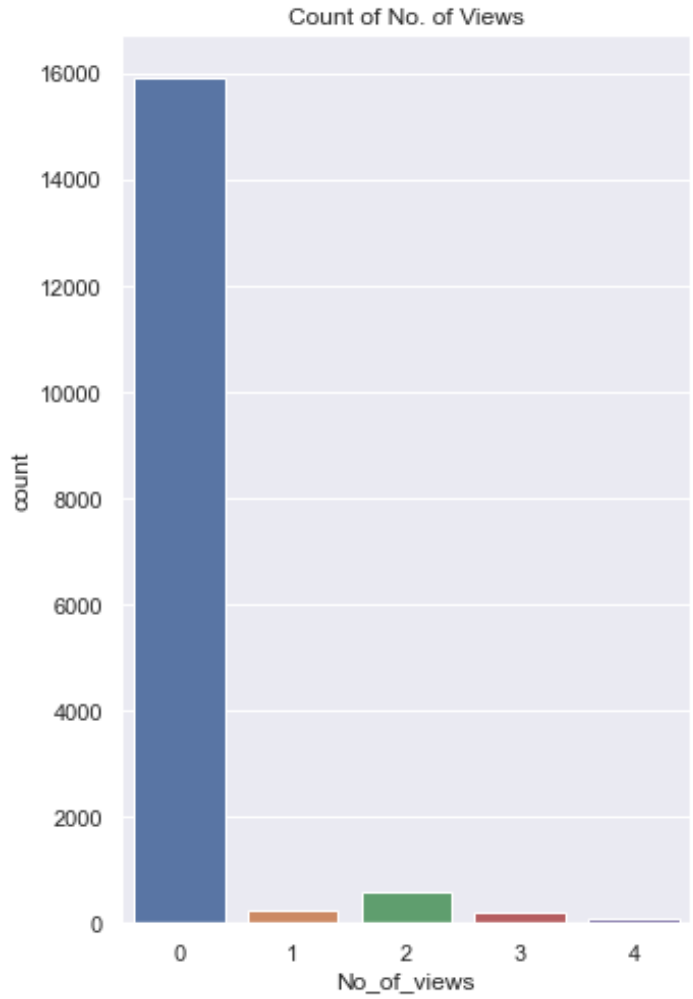
Observation:

- More than 95% of the houses do not have a waterfront. Hence, we will drop this column during the data cleaning step.

xii) No of Views

This variable shows if the house has been viewed or not.

(Diagram 17)



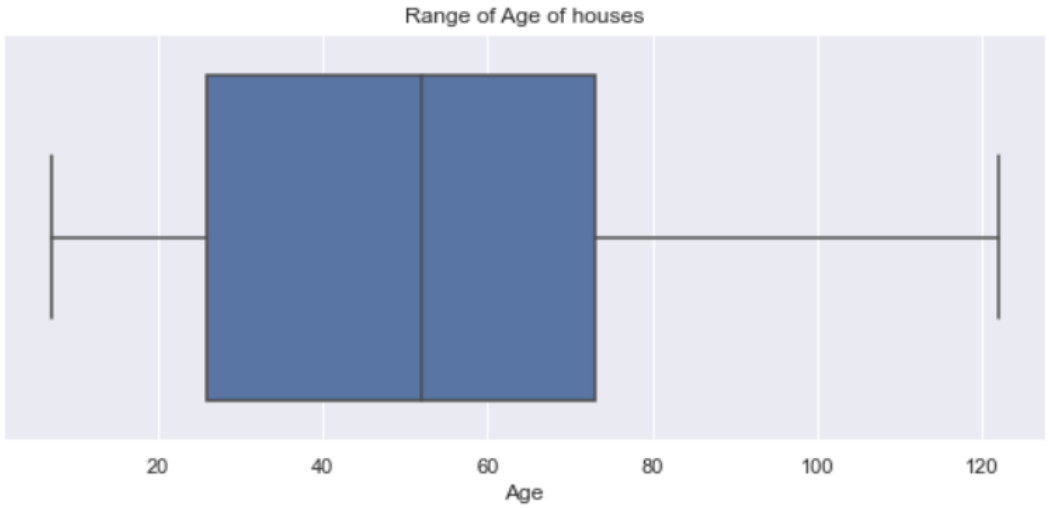
Observations:

- Majority of the houses have not been viewed. Hence, we will remove this column

xiii) Age of the House

This variable shows the Age of the house.

(Diagram 18)



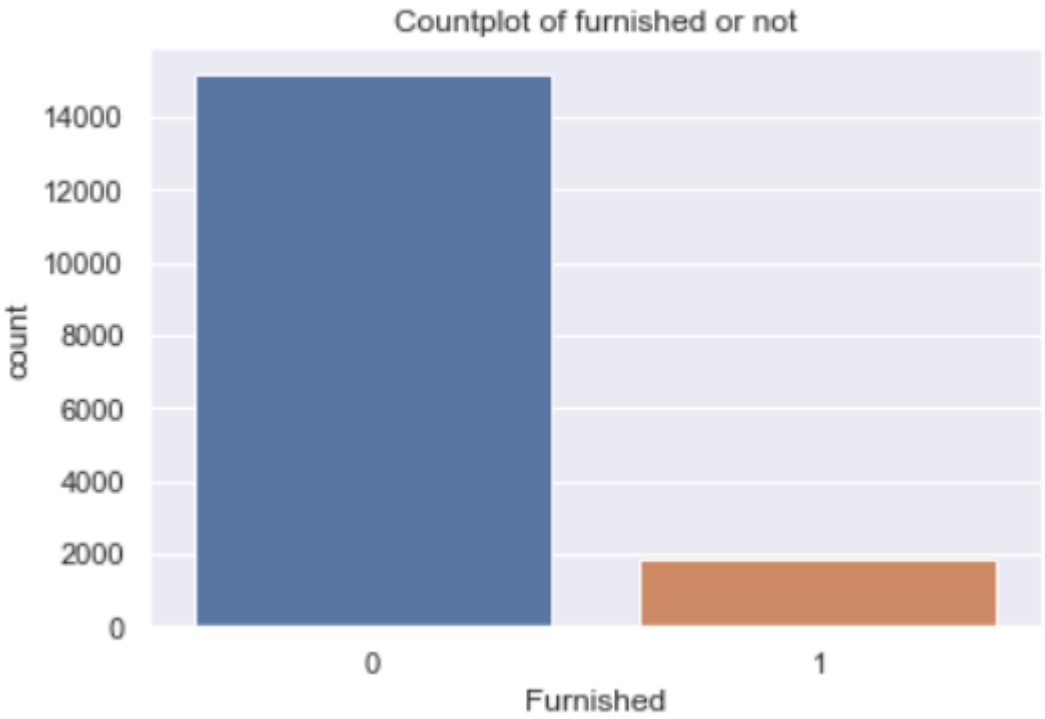
Observations:

- The average age of a house is 52.
- Most houses are 24 to 76 years old.

xiv) Furnished

This variable shows if the house is furnished or not

(Diagram 19)



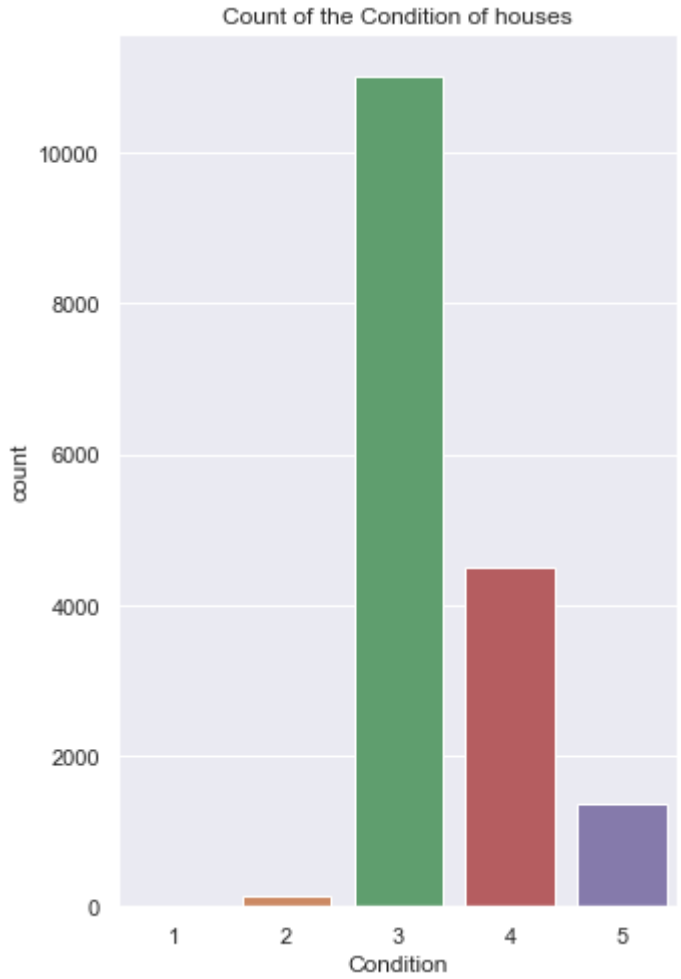
Observations:

- Most houses have not been furnished; we will compare this with prices to see if furnished houses are more expensive compared to unfurnished houses.

xv) Condition

This variable describes the condition of the house on a scale of 1 to 5.

(Diagram 20)



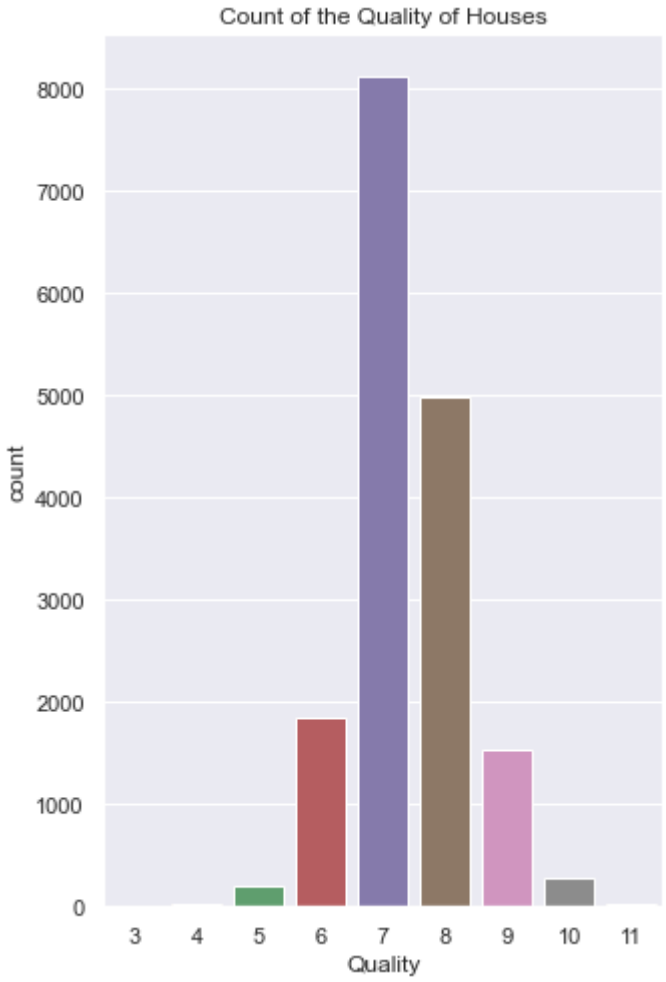
Observations:

- Most houses have a condition of 3 or better. We will see if this variable impacts the price in anyway.

xvi) **Quality**

This variable describes the quality of the house on a scale of 1 to 10.

(Diagram 21)



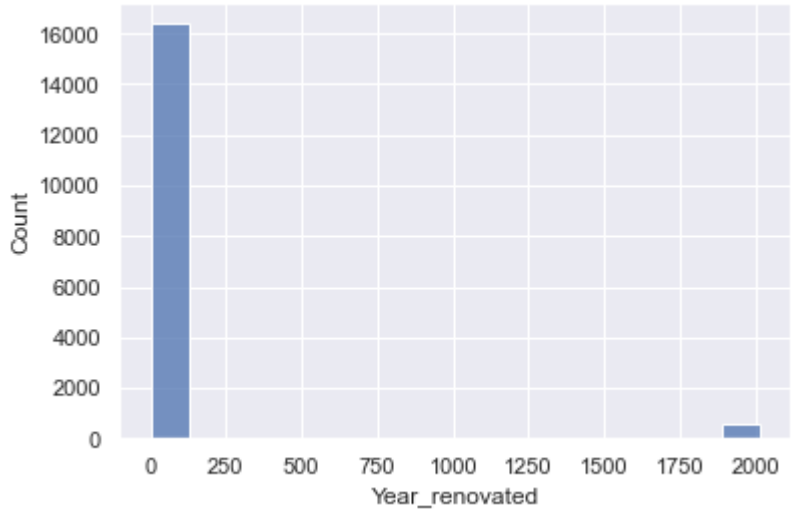
Observations:

- Most houses have a quality from 6 to 9. We will see if this variable affects the price in any manner.

xvii) Year Renovated

This variable tells us which year the house was renovated.

(Diagram 22)



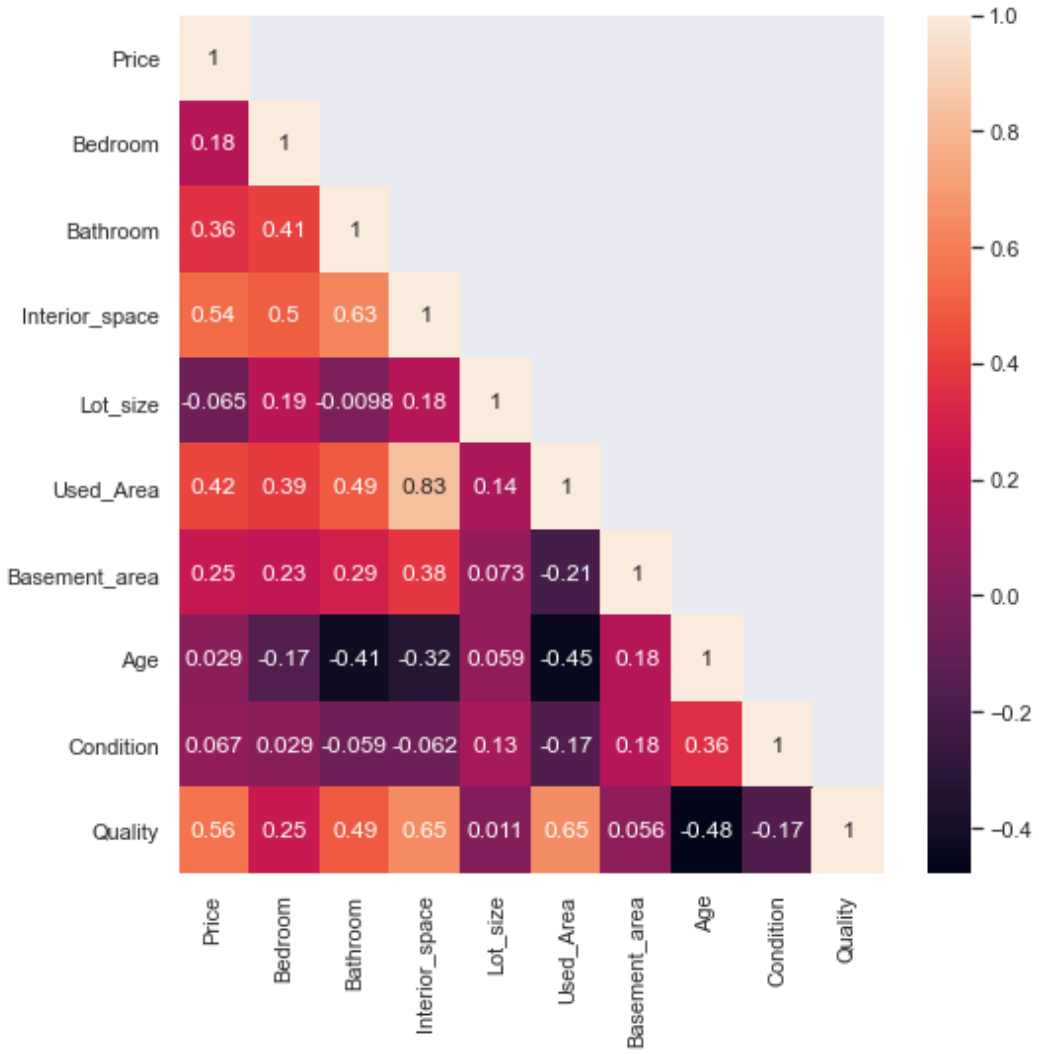
Observations:

- Most of the houses have not been renovated. Hence, we will drop this column.

Bivariate Analysis

Below, we observe the correlation between variables through a heatmap.

(Diagram 23)



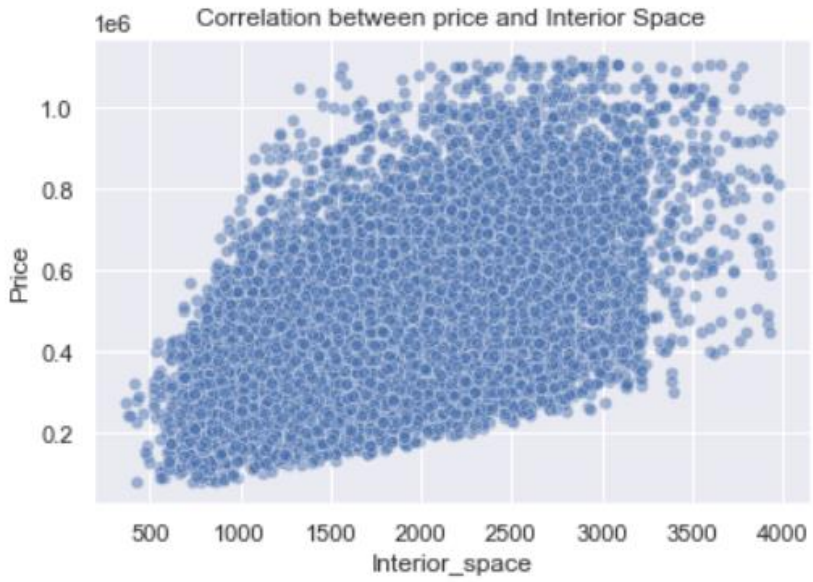
We compare variables with correlation to Price (target variable) higher than 0.4 or lesser than -0.4.

The variables are:

1. Interior Space
2. Used Area
3. Quality

1. **Price** vs **Interior Space**

(Diagram 24)

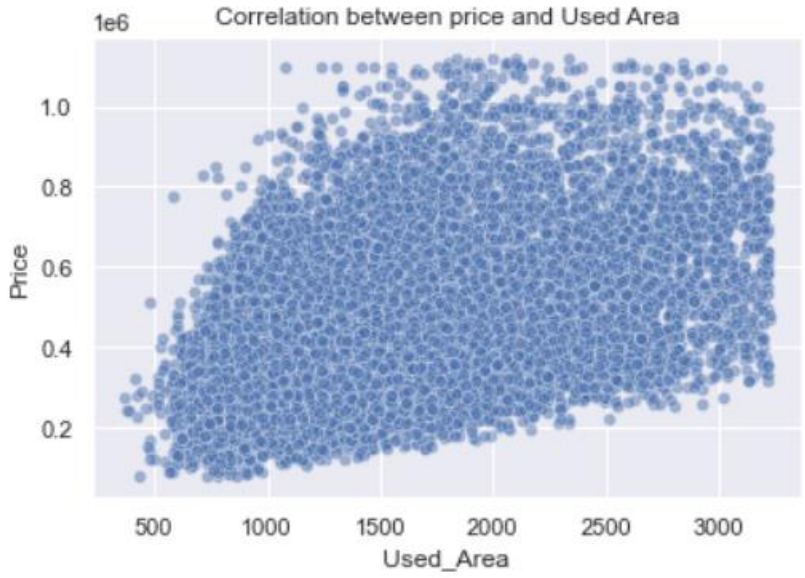


Observation:

- With increasing Interior Space, Price increases.

2. **Price** & **Used Area**.

(Diagram 25)

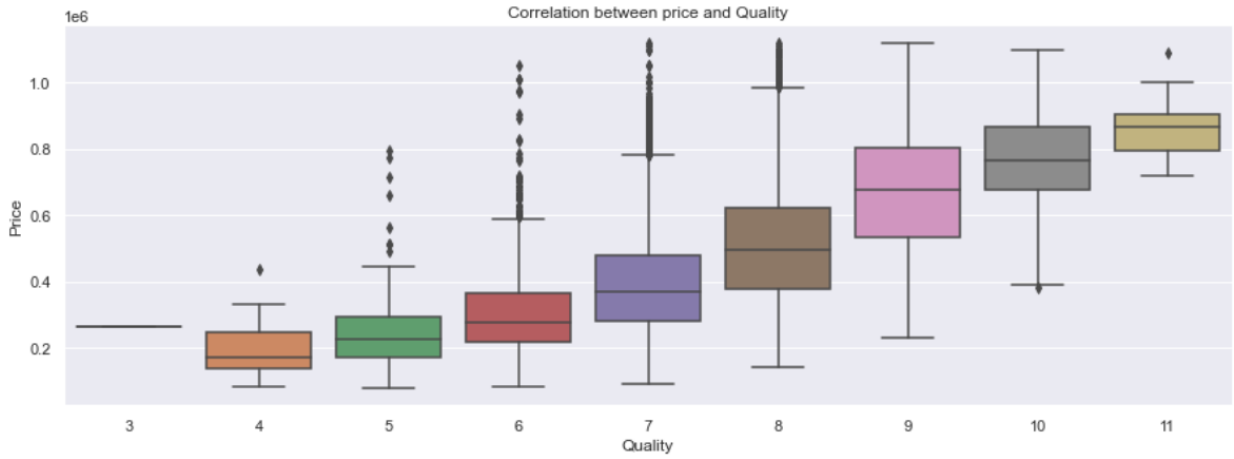


Observation:

With increasing Used Area, Price increases.

3. **Price** & **Quality**

(Diagram 26)



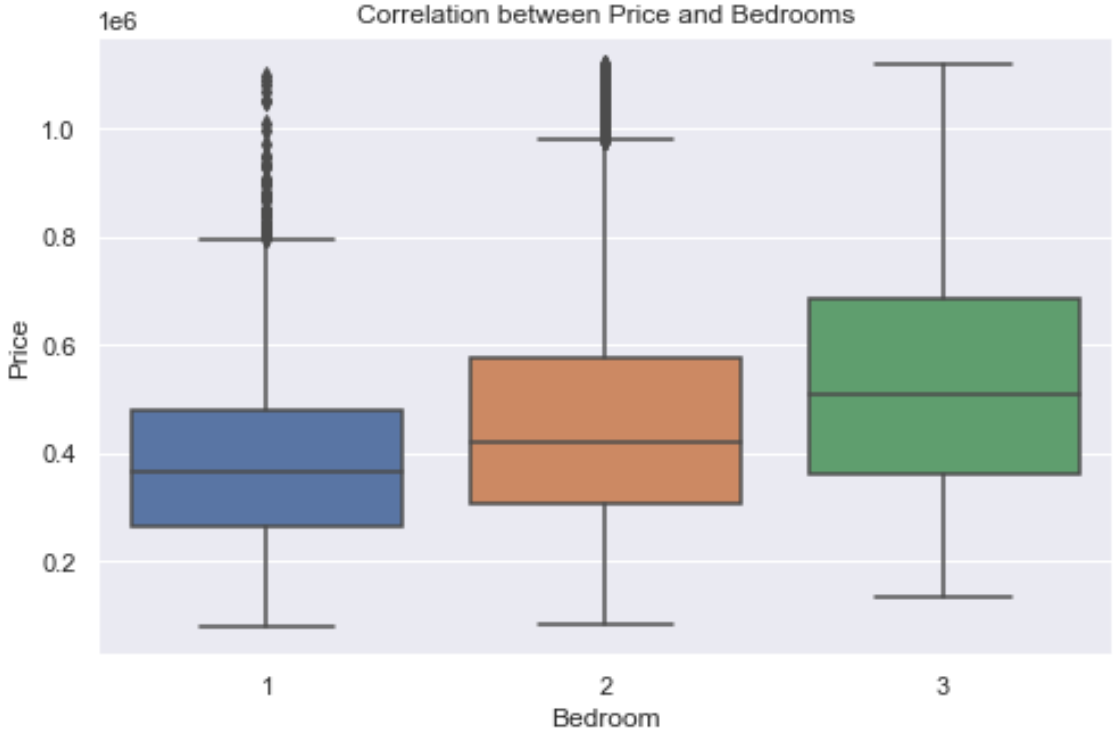
Observation:

With increasing Quality, Price increases.

4. **Price** & **Bedroom**

1 = 1/2 Bedrooms; 2 = 3/4 Bedrooms; 3 = 4+ Bedrooms

(Diagram 27)



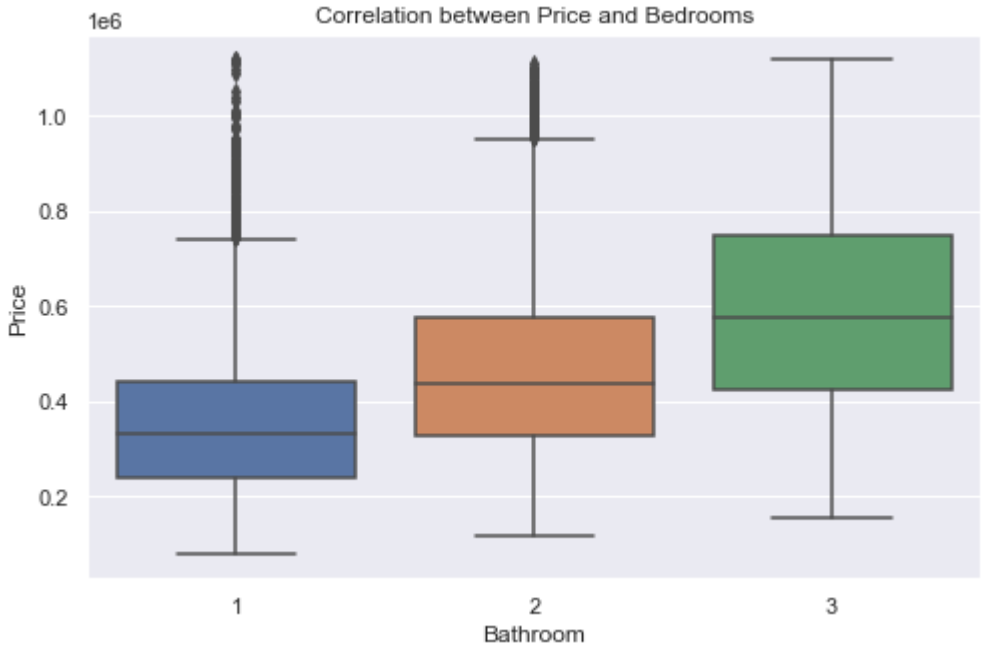
Observation:

With increasing bedrooms, average Price increases.

5. **Price** & **Bathroom**

1 = Less than 1.5 Bath; 2 = 1.5 to 2.5 Bath; 3 = 2.5+ Bath

(Diagram 28)



Observation:

With increasing Bathrooms, average Price increases.

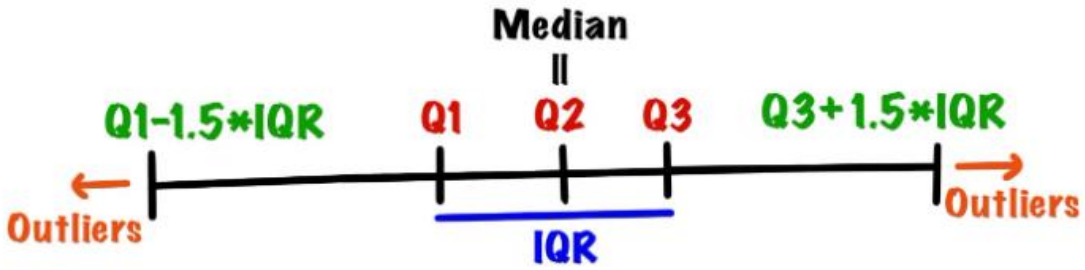
III. Data Cleaning & Pre-processing

i) Outlier Treatment

Method 1:

We perform outlier treatment using the 1.5 Inter Quartile Range method.

(Diagram 29)



This method uses a percentile to find out Q1 & Q3. It then finds the upper and lower limits for the data which is Q3 + (1.5*IQR) & Q1 – (1.5*IQR) respectively. It then eliminates all the values falling out of these range.

The variables that were treated for outliers using the 1.5 IQR method are the following:

1. Price
2. Interior Space
3. Lot Size
4. Used Area
5. Basement Area

Method 2:

We perform manual outlier treatments for certain variables which contain random values or characters.

1. Bedrooms:

We noticed that there was a transaction where a unit had **33** bedrooms. Upon closer analysis of the sale, the price was determined to be low compared to the number of units. Hence, we remove that transaction manually.

2. Waterfront

There were certain random characters which were manually removed.

3. No of Views

There were certain random characters which were manually removed.

4. Floors

There were certain random characters which were manually removed.

ii) Creating new variables

1. **Month_sold –** We extract the month of the sale from **dayhours** variable
2. **Day_sold –** We extract the day of the sale from **dayhours** variable
3. **Yonth_sold –** We exctract the year of the sale from **dayhours** variable
4. **Bedrooms –** We create a new variable called **Bedroom** from the variable **bedrooms** to easily classify the units into 3 groups (1/2BHK, 3/4BHK, 4+BHK).
5. **Bathroom –** We create a new variable called **Bathroom** from the variable **bathrooms** to easily classify the units into 3 groups (<1.5 Bath, 1.5-2.5 Bath, 2.5+ Bath).

iii) Removal of columns

1. **cid** – This represents a house ID, it is of no relevance for our model
2. **Total area** – This variable is the sum of Interior space & Lot size. In the real world, it wouldn't make sense to add both. Hence, we drop this column.
3. **Interior Space 15** – This is the interior space of the unit in 2015. The current Interior size is more relevant. Hence, we drop this column.
4. **Lot Size 15** – This is the lot size of the unit in 2015. The current Interior size is more relevant. Hence, we drop this column.

5. **No of Views** – This variable is irrelevant as majority of the houses have not been viewed. Hence, we drop this column.
6. **Bedrooms** – This column is no longer needed as we have created a new variable **Bedroom** which replaces the old variable.
7. **Bathrooms** – This column is no longer needed as we have created a new variable **Bathroom** which replaces the old variable.
8. **Latitude** – We drop this column as our system does not have the required packages to utilize this
9. **Longitude** - We drop this column as our system does not have the required packages to utilize this

iv) Null Value Treatment

We check all the columns for null values.

(Table 1)

Variable	***Null Values Present***
Bedrooms	78
Bathrooms	78
No_of_views	46
Condition	46
Furnished	23

As the number of null values present is very insignificant compared to the overall quantum of sales transactions, we drop all the rows containing null values.

v) Variable Transformation

We convert the following variables to use them as inputs in our machine learning model.

1. Bedrooms

We convert this into numerical.

(Table 2)

Old Value	***New Value***
1/2BHK	1
3/4BHK	2
4+BHK	3

2. Bathrooms

We convert this into numerical.

Old Value	***New Value***
<1.5 Bath	1
1.5-2.5 Bath	2
2.5+ Bath	3

3. Data Type conversion & Renaming

We convert all the variables into their appropriate data types to take as input for our machine learning model.

We finally have the following variables which is ready to be used as input to build our model.

(Table 3)

S No	***Variable***	***Data Type***
1	Years_since_txn	Integer
2	Price	Integer
3	Bedroom	Integer
4	Bathroom	Integer
5	Floors	Float
6	Interior_space	Integer
7	Lot_size	Integer
8	Used_Area	Integer
9	Basement_area	Integer
10	Waterfront	Integer
11	Age	Integer
12	Furnished	Integer
13	Condition	Integer
14	Quality	Integer
15	Zipcode	Integer

Our dataset now contained 16989 transactions. We now proceed to the next phase of our project.

vi) Checking for importance

Now, we perform a T test to check if the variables have a significant impact on the Price of a unit.

We do this by finding out the P value. If the P value is less than 0.05, it means that the variable does significantly affect the Price of a unit. The null hypothesis is that the variable does not significantly affect the Price of a unit, hence we reject the null hypothesis if the P value is less than 0.05.

(Table 4)

Variable	***P value***	***Reject Null?***
Waterfront	0.00	Yes
Furnished	0.00	Yes

After testing, we find that both **Waterfront** & **Furnished** have significant effect on the Price of a unit.

IV. Model Building

We split the dataset into Training and Testing datasets in the ratio of 70:30.

The training data is used to train the model.

The model is then tested on the testing data to predict the target variable.

We then compare the accuracy of the predicted target variable to get a score in Percentage.

Our goal is to find a model which has relatively close Training & Testing scores after which we can further tune it to increase accuracy.

We start by testing 3 models and find out their initial accuracy.

(Table 5)

Model	*Testing Score*	*Training Score*
Decision Tree Regressor	52.33%	100.00%
Decision Tree Regressor tuned	64.36%	84.85%
Random Forest Regressor	75.86%	96.83%
Gradient Boosting Regressor	72.53%	74.28%

1. Decision Tree Regressor

Decision tree builds regression or classification models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The result is a tree with decision nodes and leaf nodes.

During our testing, we observed that the model was overfitted as the Training score was 100%. We slightly tuned the model to reduce the overfitting, but still the model was quite overfitted as the Training score was significantly better than the Testing score.

2. Random Forest Regressor

A random forest regressor is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

During our testing, we observed that this model too was overfitted.

3. Gradient Boosting Regressor.

In Gradient boosting weak learners are decision trees. This is how the algorithm works.

Step1: Construct a base tree with single root node. It is the initial guess for all the samples.

Step2: Build a tree from errors of the previous tree.

Step3: Scale the tree by learning rate (value between 0 and 1). This learning rate determines the contribution of the tree in the prediction

Step4: Combine the new tree with all the previous trees to predict the result and repeat step 2 until maximum number of trees is achieved or until the new trees don't improve the fit.

The final prediction model is the combination of all the trees.

During our testing, this model seemed to be a perfect fit as the Training scores & Testing scores were very close to each other.

Hence, we choose Gradient Boosting Regressor as our model to further tune and improve model accuracy.

Gradient Boosting Regressor model has a lot of hyper parameters such as

i) loss : ['ls', 'huber', 'lad'],

ii) max_depth : range(5, 15, 30),

iii) max_features : ['auto','sqrt'],

iv) learning_rate : [0.01, 0.1, 0.2],

v) min_samples_leaf : [5, 7, 10],

vi) min_samples_split : [10, 100, 1000],

vii) n_estimators : [50, 100, 200],

viii) subsample : [0.6, 0.9]

To find out the best combination of all the parameters, we use a function called **GridSearchCV.**

GridSearchCV helps to loop through predefined hyperparameters and fit your estimator (model) on your training set. So, in the end, we can select the best parameters from the listed hyperparameters.

Below is our score after tuning our model.

(Table 6)

Model	*Testing Score*	*Training Score*
Gradient Boosting Regressor tuned	83.67%	91.54%

V. Model Validation

We have validated our model using the measure of accuracy. This means that we compare the predicted target variable with the actual value and see how accurate it is. We derive a percentage score to determine accuracy.

We also checked for feature importance to see it makes sense in the real world.

The following features cumulatively has 96% importance in the model. They are ranked in the following order.

1. Zip code
2. Interior Space
3. Age
4. Furnished
5. Lot Size
6. Quality
7. Used Area
8. Basement Area

VI. Final Interpretation / Recommendation

- We can conclude that our model has a high accuracy and can be used in the real world to predict the price of a unit.

- It builds in a short period of time giving us the ability to continuously train the model as and when we get new sales data.

Recommendations

- The most important features that determine the price of a house is the Zip code. Any buyer/seller must check the average price of the houses in their Zip code before making any transaction.

- The buyer/seller must give importance to the Interior Space of the unit when determining price.

- The buyer/seller must find out the Age of the house as the older the house, the higher probability of it requiring repair.

- The buyer/seller must visually assess the quality of the house before purchasing

- The buyer/seller must pay/ask for a premium if the house is furnished.

- The buyer/seller must also check how well the lot is utilized when the house was built. More utilization most likely means the area is in demand as the plot sizes are smaller.