House Price Prediction

Capstone Project

by Ashwin

Table of Content

1.	Introduction	2
2.	Problem statement	2
3.	EDA	3
3.1	Data Description	4
3.2	Missing Values	5
3.3	Univariate Analysis & Bivariate Analysis	5
3.4	Multivariate Analysis	19
4.	Data Cleaning and Pre-processing (Outliers)	21
4.1	Variable inclusion and removal	23
4.2	Variable Transformation	23
4.3	Scaling	23
5.	Model building	24
5.1	Evaluation Parameters	25
6.	Model Validation	25
6.1	Tuning	26
7.	Feature Importance	27
8.	Recommendation & Insights	28

1. Introduction

Real estate is an important sector with many stakeholders. This dataset consists of properties that are built around the city of Washington between the year 2014 and 2015.

Nowadays buying a house became a challenging task. Even though the bank offers housing loans, buying an own house is quite a large investment and we have to choose the best one in the market.

It is our role as a Data Analyst to likely predict how a buyer, seller or a real estate promotor would be benefitted while buying a house in this locality based of some factors. The aim of our project is to build a predictive model for change in house prices based on certain variables such as no. of bathrooms, no. of bedrooms, living room measurement, coastal view, whether it's furnished or not et cetera.

2. Problem statement

A house value is simply more than location and square footage. Like the features that make up a person, an educated party would want to know all aspects that give a house its value. For example, if we want to sell a house and we don't know the price which we may expect, it can't be too low or too high.

To find house price we usually try to find similar properties in our neighbourhood and based on gathered data we will try to assess the house price.

3. EDA

This dataset consists of 21613 rows and 23 columns and there are no duplicates. There are 7 categorical variables and 16 numerical variables.

cid	dayhours	price	room_bed	room_bath	living_measure	lot_measure	ceil	coast	sight
3876100940	20150427T000000	600000	4.0	1.75	3050.0	9440.0	1	0	0.0
3145600250	20150317T000000	190000	2.0	1.00	670.0	3101.0	1	0	0.0
7129303070	20140820T000000	735000	4.0	2.75	3040.0	2415.0	2	1	4.0
7338220280	20141010T000000	257000	3.0	2.50	1740.0	3721.0	2	0	0.0
7950300670	20150218T000000	450000	2.0	1.00	1120.0	4590.0	1	0	0.0

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21613 entries, 0 to 21612
Data columns (total 23 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   cid               21613 non-null  int64  
 1   dayhours          21613 non-null  object 
 2   price             21613 non-null  int64  
 3   room_bed          21505 non-null  float64
 4   room_bath         21505 non-null  float64
 5   living_measure    21596 non-null  float64
 6   lot_measure       21571 non-null  float64
 7   ceil              21571 non-null  object 
 8   coast             21612 non-null  object 
 9   sight             21556 non-null  float64
 10  condition         21556 non-null  object 
 11  quality           21612 non-null  float64
 12  ceil_measure      21612 non-null  float64
 13  basement          21612 non-null  float64
 14  yr_built          21612 non-null  object 
 15  yr_renovated      21613 non-null  int64  
 16  zipcode           21613 non-null  int64  
 17  lat               21613 non-null  float64
 18  long              21613 non-null  object 
 19  living_measure15  21447 non-null  float64
 20  lot_measure15     21584 non-null  float64
 21  furnished         21584 non-null  float64
 22  total_area        21584 non-null  object 
dtypes: float64(12), int64(4), object(7)
memory usage: 3.8+ MB
```

Some of these numerical variables are required to be converted to categorical in order to pre-process.

3.1 Data Description

	count	mean	std	min	25%	50%	75%	max
cid	21613.00	4580301520.86	2876565571.31	1000102.00	2123049194.00	3904930410.00	7308900445.00	9900000190.00
price	21613.00	540182.16	367362.23	75000.00	321950.00	450000.00	645000.00	7700000.00
room_bed	21505.00	3.37	0.93	0.00	3.00	3.00	4.00	33.00
room_bath	21505.00	2.12	0.77	0.00	1.75	2.25	2.50	8.00
living_measure	21596.00	2079.86	918.50	290.00	1429.25	1910.00	2550.00	13540.00
lot_measure	21571.00	15104.58	41423.62	520.00	5040.00	7618.00	10684.50	1651359.00
sight	21556.00	0.23	0.77	0.00	0.00	0.00	0.00	4.00
quality	21612.00	7.66	1.18	1.00	7.00	7.00	8.00	13.00
ceil_measure	21612.00	1788.37	828.10	290.00	1190.00	1560.00	2210.00	9410.00
basement	21612.00	291.52	442.58	0.00	0.00	0.00	560.00	4820.00
yr_renovated	21613.00	84.40	401.68	0.00	0.00	0.00	0.00	2015.00
zipcode	21613.00	98077.94	53.51	98001.00	98033.00	98065.00	98118.00	98199.00
lat	21613.00	47.56	0.14	47.16	47.47	47.57	47.68	47.78
living_measure15	21447.00	1987.07	685.52	399.00	1490.00	1840.00	2360.00	6210.00
lot_measure15	21584.00	12766.54	27286.99	651.00	5100.00	7620.00	10087.00	871200.00
furnished	21584.00	0.20	0.40	0.00	0.00	0.00	0.00	1.00

The 5-point summary of the dataset shows that that there are some variations in the dataset, such as the median value of room_bed is 3 and the max is 33, also basement has median of 0 square feet and max of 4,820 square feet. Hence there is clear evidence of outliers in the data.

As we can see, the frequency of the variables are not consistent. Some of them display high range values, while some display low range values. So scaling is necessary in order to normalize the data within a particular range and it also helps in speeding the model calculations.

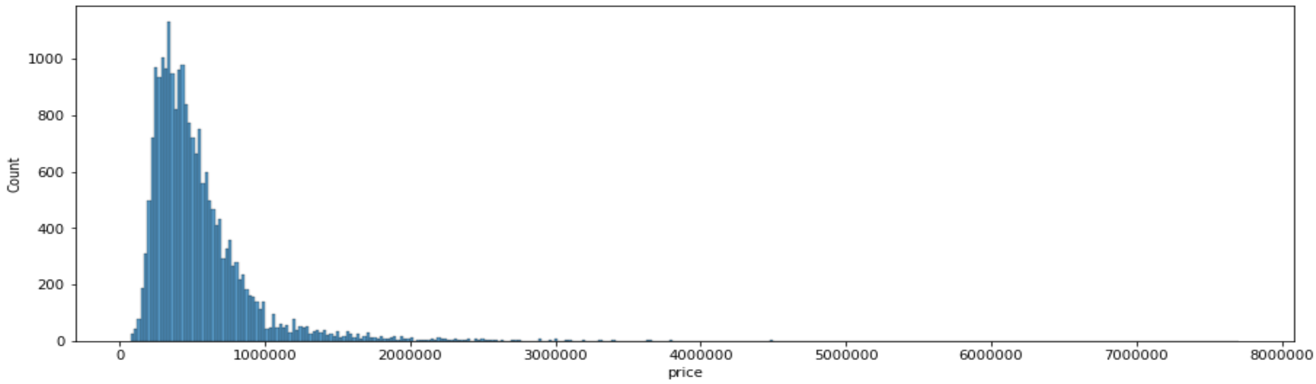
3.2 Missing Values

```
cid                  0
dayhours             0
price                0
room_bed           108
room_bath          108
living_measure      17
lot_measure         42
ceil                42
coast                1
sight               57
condition           57
quality              1
ceil_measure         1
basement             1
yr_built             1
yr_renovated         0
zipcode              0
lat                  0
long                 0
living_measure15   166
lot_measure15       29
furnished           29
total_area          29
dtype: int64
```

Almost all the columns have missing values, where living_measure15 has the highest number of missing values of 166. The total number of missing values is 689. These null or missing values can be imputed with median for numerical columns and mode for categorical columns based on their percentage.

3.3 Univariate Analysis & Bivariate Analysis

- **price**

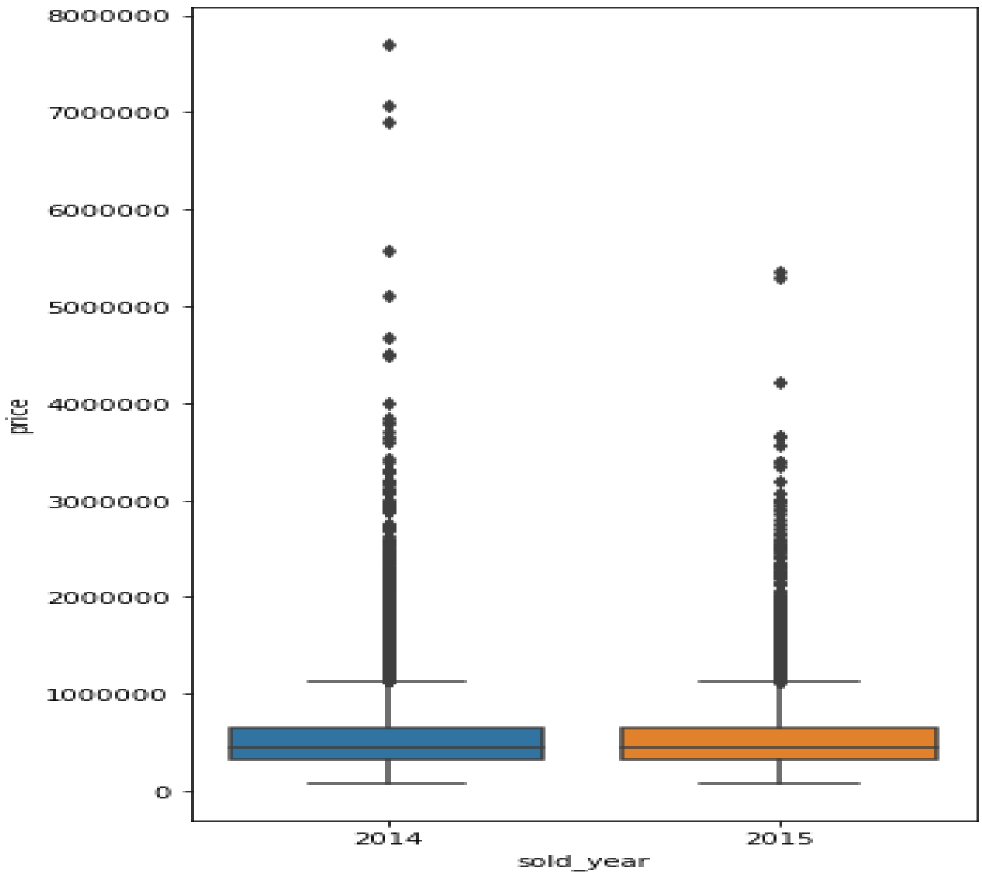


Price is the target column. From the above histogram we can see that most of the properties are priced under 10,00,000. The highest price of a property is 7,700,000 and the lowest is 75,000. It is normally distributed and median value of a house is priced at 4,50,000.

- **cid**

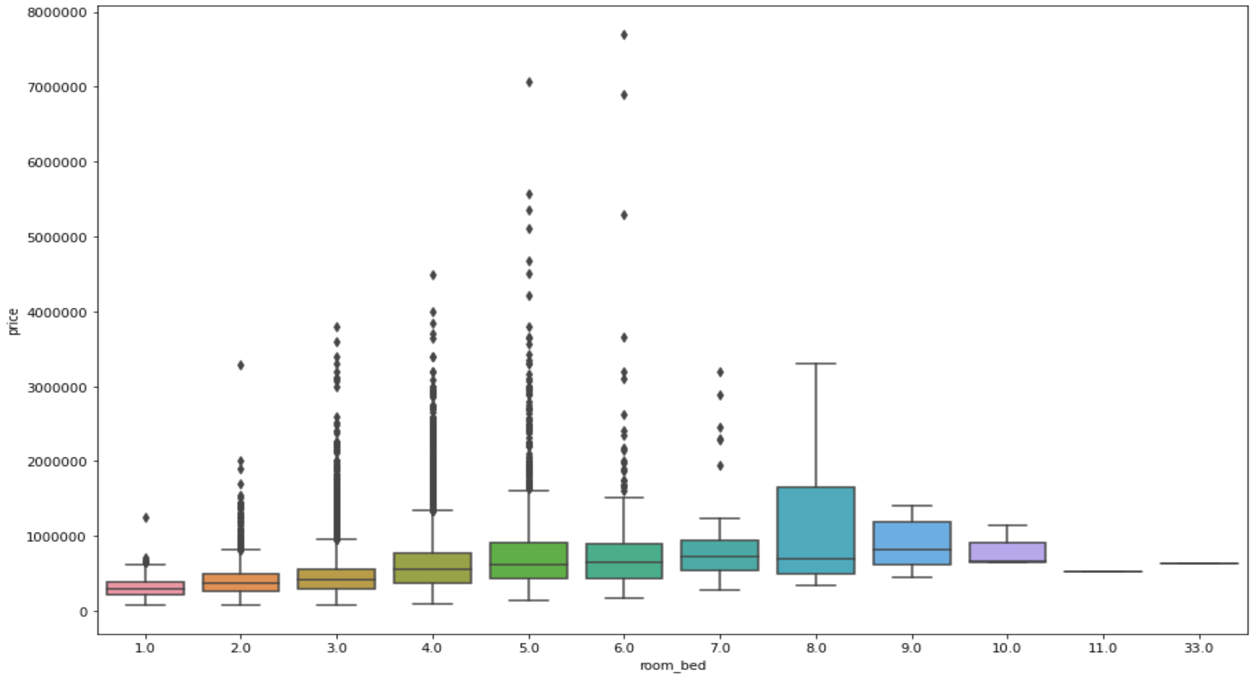
It's a notation for a house. We can drop this column since it doesn't contain any meaningful data that we can use for further analysis and to build our model.

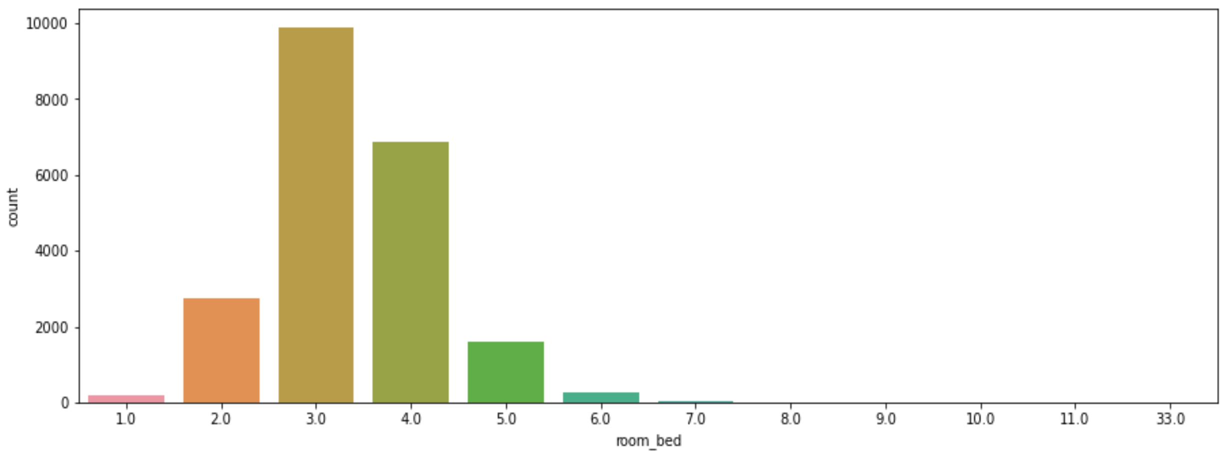
- **dayhours**



This column shows which year the house was sold. All the houses were sold in the years of 2014 and 2015, wherein 2014, 14,633 houses were sold. And in the year 2015, only 6,980 were sold. Against the target column, houses sold in the year 2014 and 2015 are equally priced. New column sold_year is added by dropping dayhour.

- **room_bed**

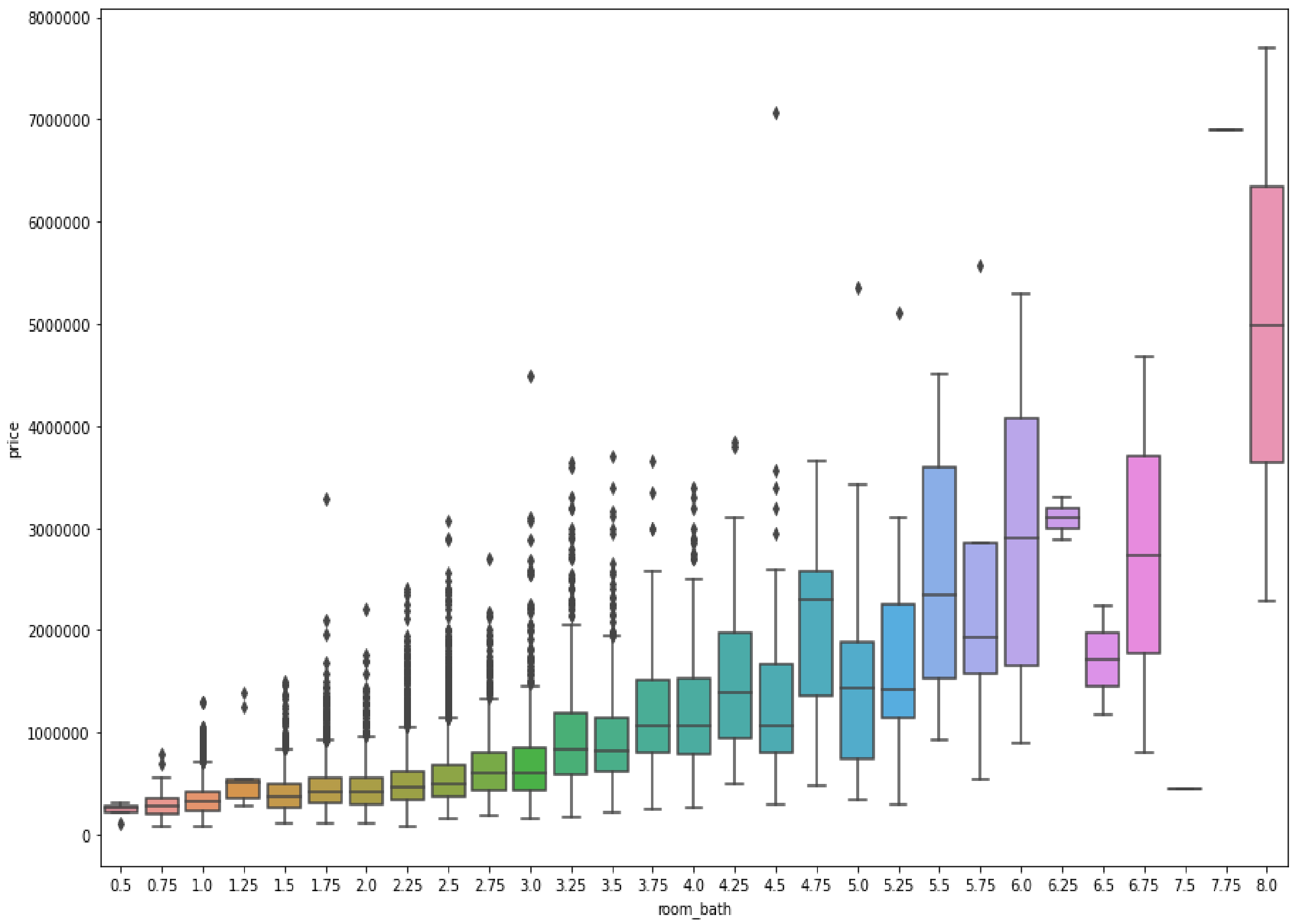


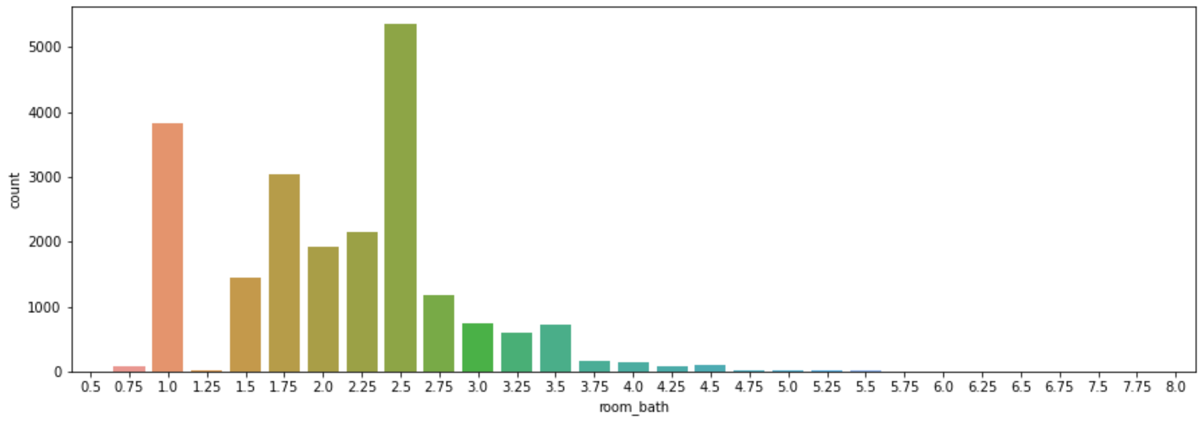


room_bed represents number of bedrooms in a house. The above count plot shows that houses with 3 bedrooms are the highest, 9,767. And houses with 33 bedrooms are the least, 1. But 33 maybe an outlier or an error, which will be treated before model building.

According to boxplot, houses with 8 bedrooms are priced the highest compared to other houses. There are 13 houses with 0 bedrooms, which is an error, since houses always comes with bedrooms. Hence, 0 is replaced as missing values and are treated with imputation.

- **room_bath**

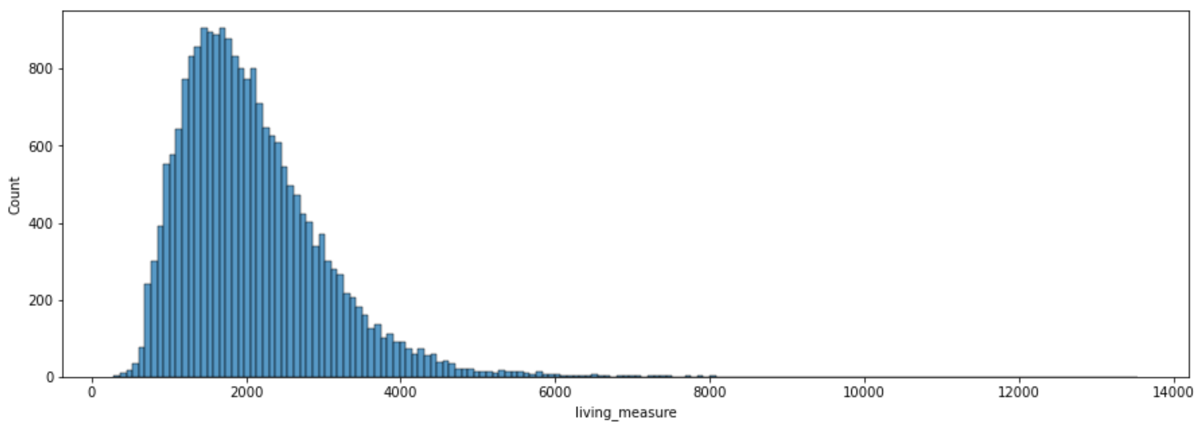




room_bath represents number of bathrooms in a house. From the above count plot it is clear that houses with 2.5 bathrooms have the highest count, 5,358. And 7.5 and 7.75 have the lowest count, 1. Houses with 8 bathrooms are priced the highest.

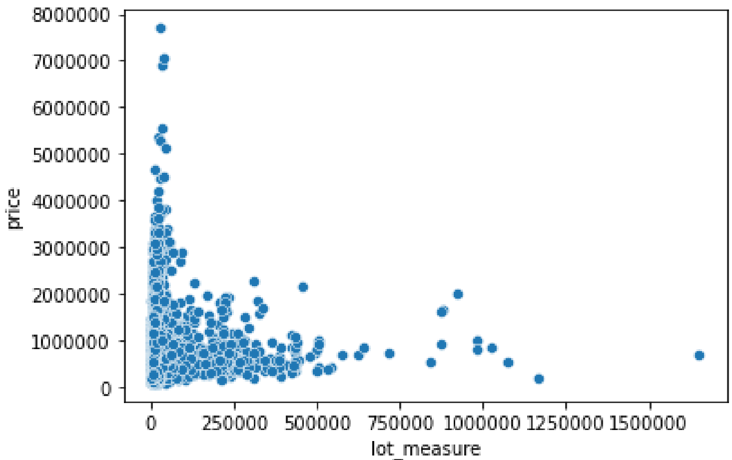
Similar to bedrooms, there 10 houses with 0 bathrooms, which is obviously an error. 0 is replaced as missing values. Outliers are present, which will be treated before building the model and missing values are imputed with median.

- **living_measure**



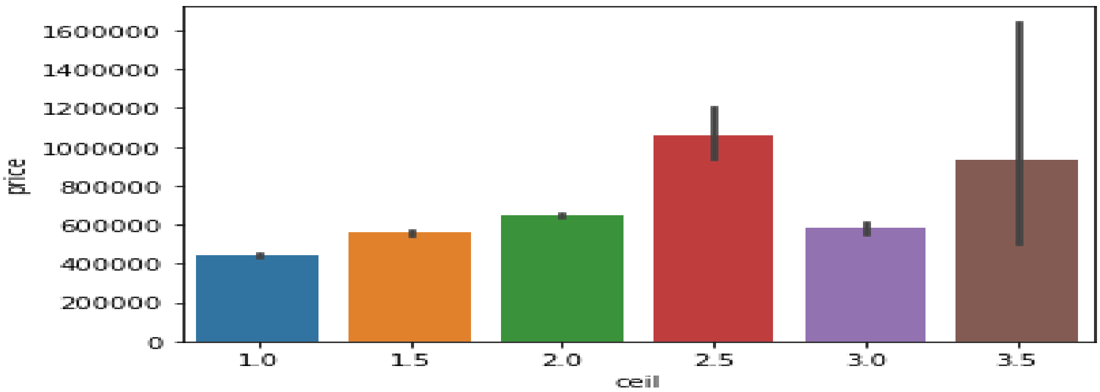
living_measure describes the square footage of the home. The above histogram shows that it is normal distribution. The median of living_measure is 1,910 square feet. 13,540 square feet is the largest house.

- **lot_measure**



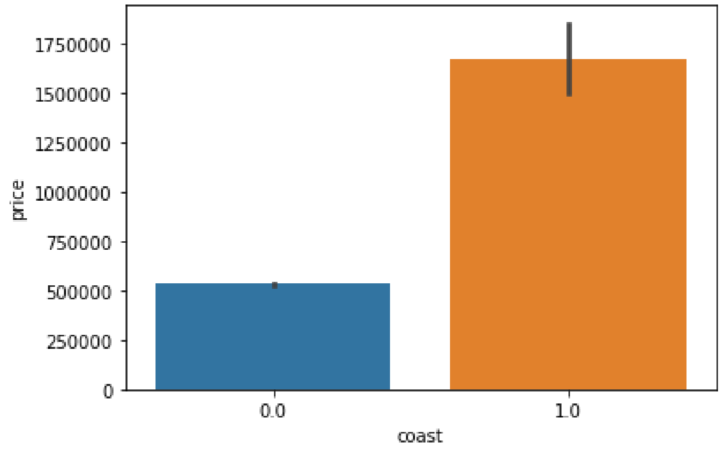
lot_measure represents square footage of the lot. In the above scatter plot, lot_measure is compared against price, where it is quite unclear to deduce any insights. The median value is 7,618 square feet and a house has 16,51,359 square feet which is the largest property.

- **ceil**



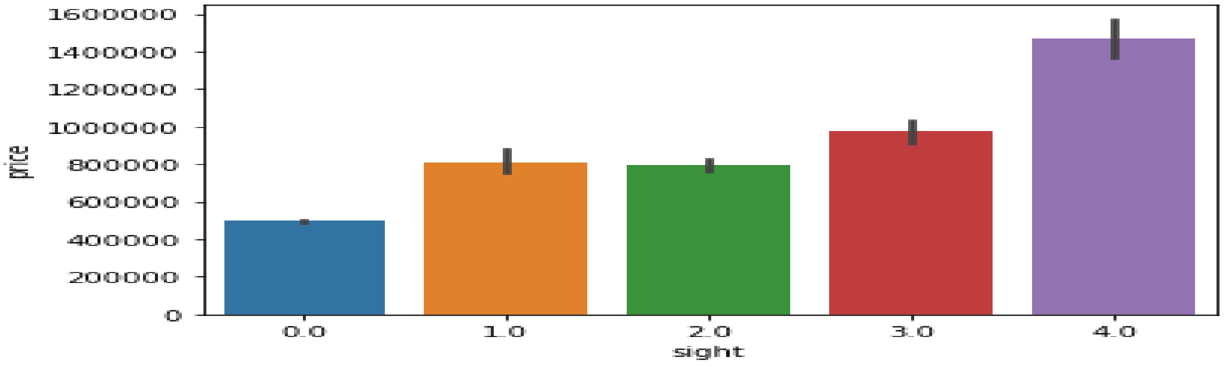
ceil represents total number of floors is a house. From the above bar graph it is clear that houses with 2.5 floors are priced the highest, while houses with 1 floor is priced the least. There are 10,647 houses with 1 floor and only 8 houses have 3.5 floors.

- **coast**



coast represents houses which has waterfront view or lakeview. There are 21,421 houses with no waterfront view and only 161 houses with waterfront view. From the above bar graph, we can say that houses with no waterfront/lake view are priced low (around 50,00,000) and houses with waterfront view are priced high.

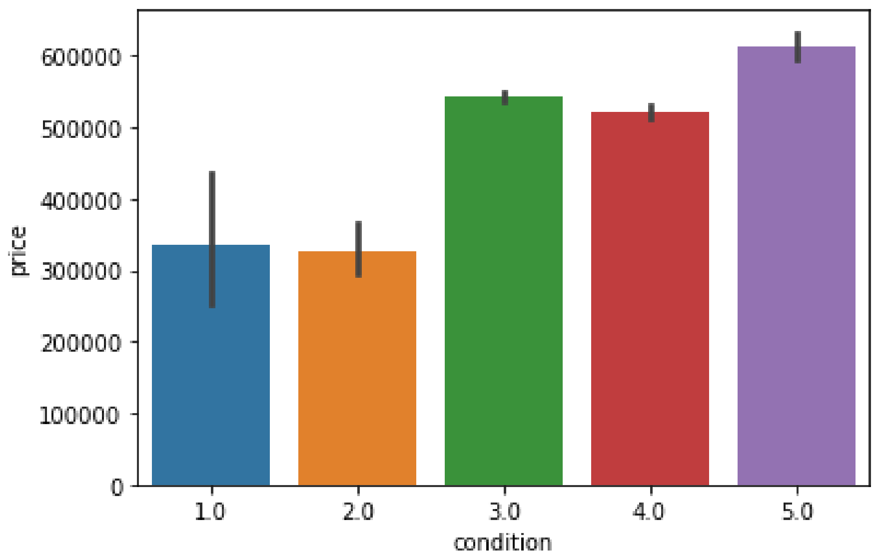
- **sight**



sight represents houses that has been viewed or visited by a customer. 19,437 properties have not been viewed even a single time. And 318 properties have been viewed 4 times, which is the highest, by the customers.

While comparing sight against price, properties that have been viewed 4 times are priced the highest and properties that have not been viewed even for 1 time are priced the lowest.

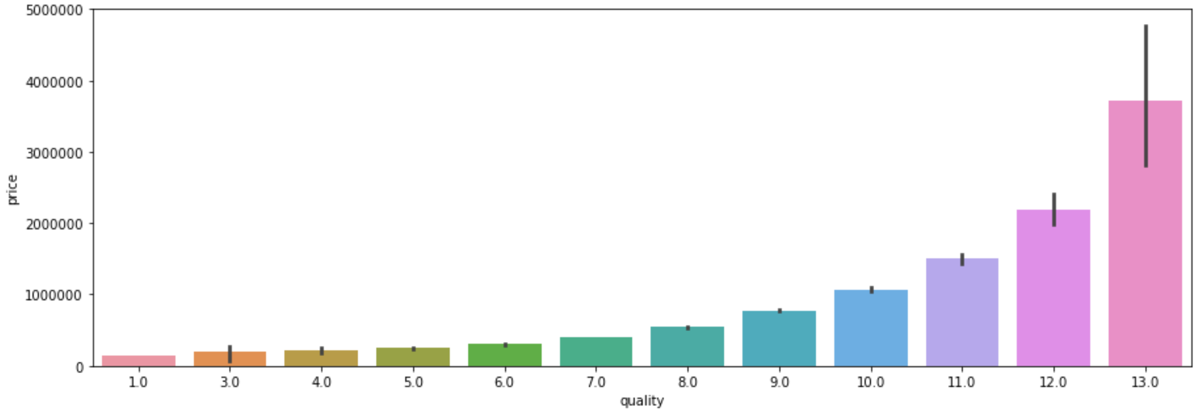
- **condition**



condition represents the condition of the property. There are only 30 properties with very low condition, 1. And 1,694 properties have very good condition, 5.

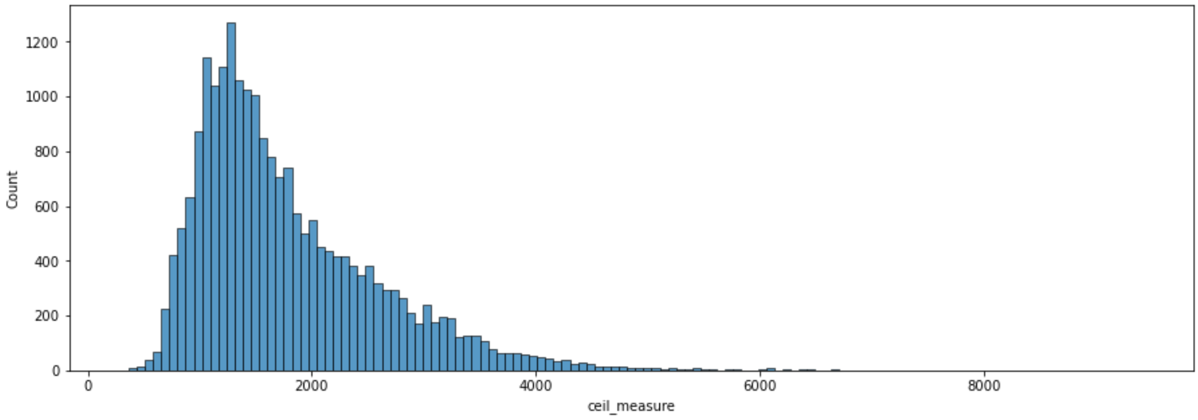
It is quite common that good condition fetches good price. Customers are always attracted to and prefer houses that are good in condition/construct. Hence, 5 rating condition houses are priced the highest and 1 rating condition houses are priced the lowest.

- **quality**



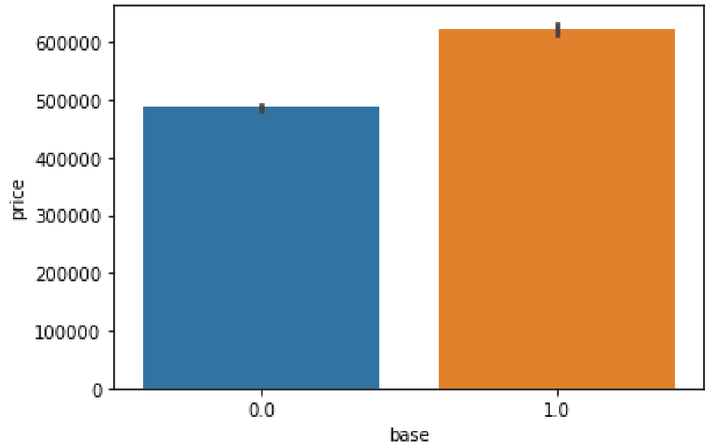
quality is similar to condition. Customers always go for houses that have best quality. From the above graph, 13 rating quality houses are the costliest and 1 rating houses are the cheapest. 13 houses have 13 rating quality and only 1 house have 1 rating quality.

- **ceil_measure**



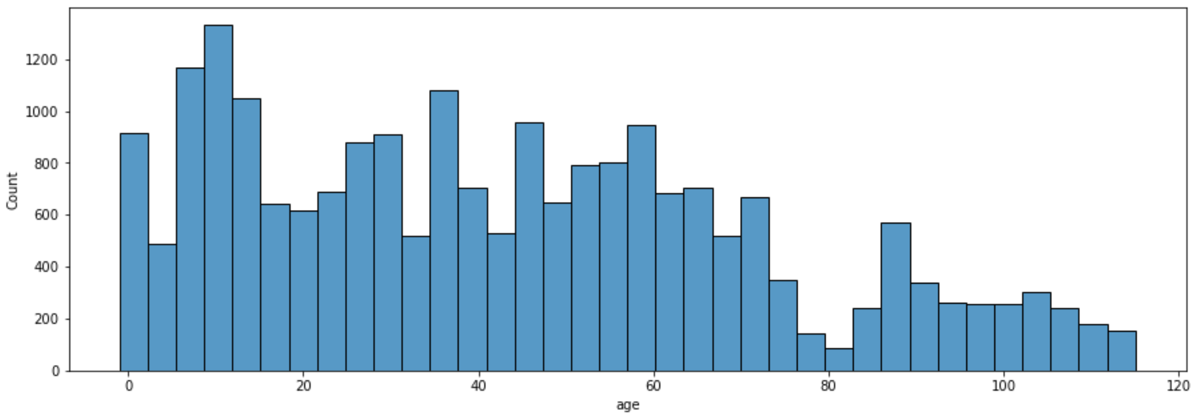
ceil_measure represents the square foot of a house apart from basement. A property with 9,410 square foot of ceiling measure is the highest and a property with 290 square foot is the lowest ceiling measure. Median is 1,560 square feet.

- **basement_measure**



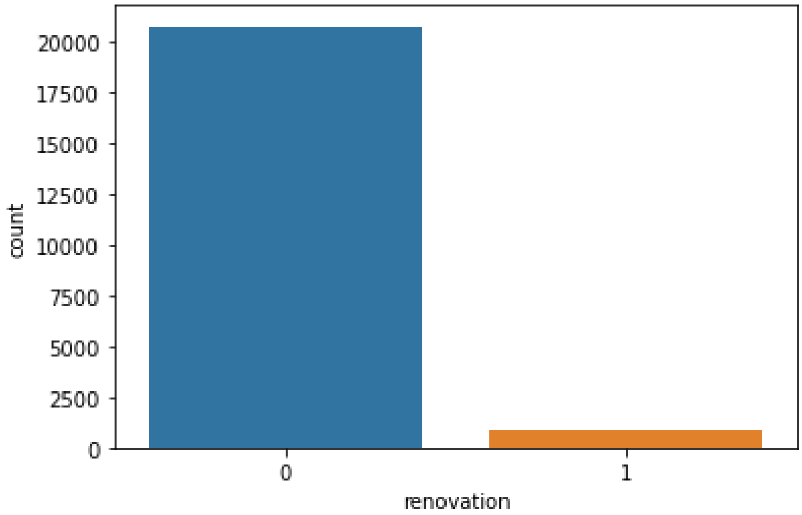
Basement represents the square footage of the basement, where 13,125 properties have 0 square foot, that is there is no basement in the house. The highest basement measure of a property is 4,820 square feet. The boxplot shows that houses that have basement are priced high when compared to houses that don't have basement. New column base is added by dropping basement.

- **yr_built**

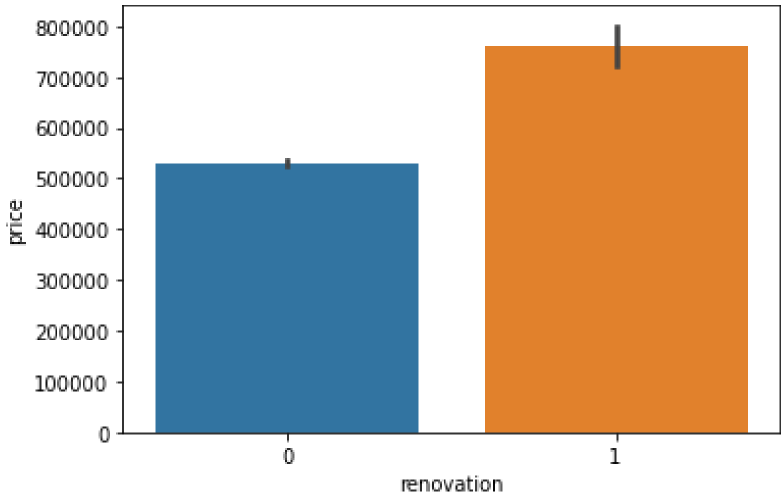


Most of the houses are built in the years between 2000 and 2015. Newer houses always fetch good price in the market. Highest number of houses were built 10 years ago. New column age is added by dropping yr_built.

- ## yr_renovated

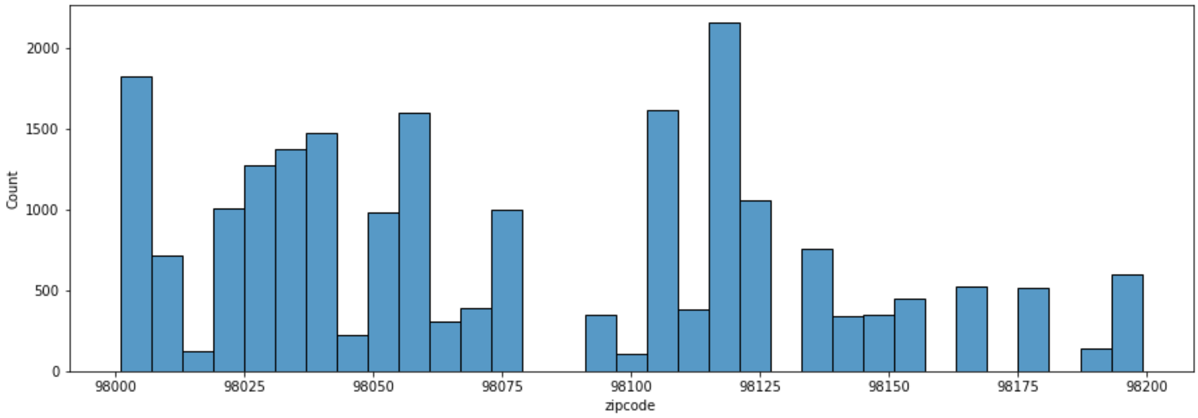


This column represents in which year the houses were renovated. We can clearly view from the above graph majority of the houses were not at all renovated (20,699 properties). New column renovation is added by dropping yr_renovated.



The above boxplot shows that houses that were renovated fetches high price in the market compared to houses that weren't renovated.

- **zipcode**

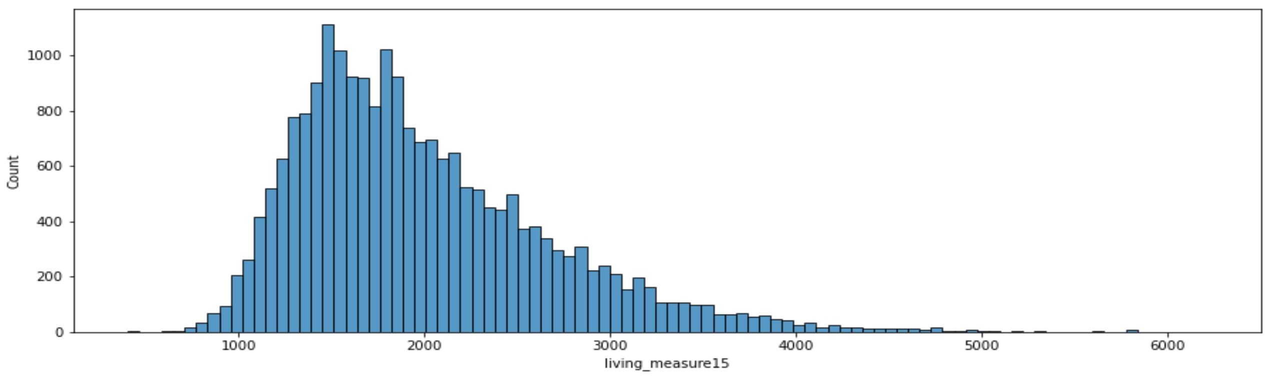


zipcode is the area where the properties are located. Highest number of properties (602) are located in the area code of 98103. And only 50 properties are located in the 98039 zipcode.

- **lat and long**

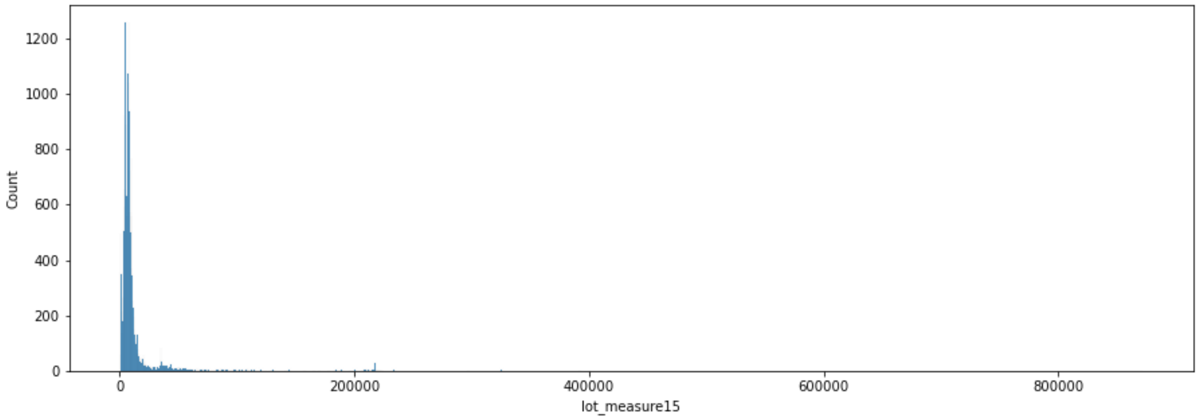
Latitude and longitude show the exact location of the property, which is quite similar to zipcode column.

- **living_measure15**



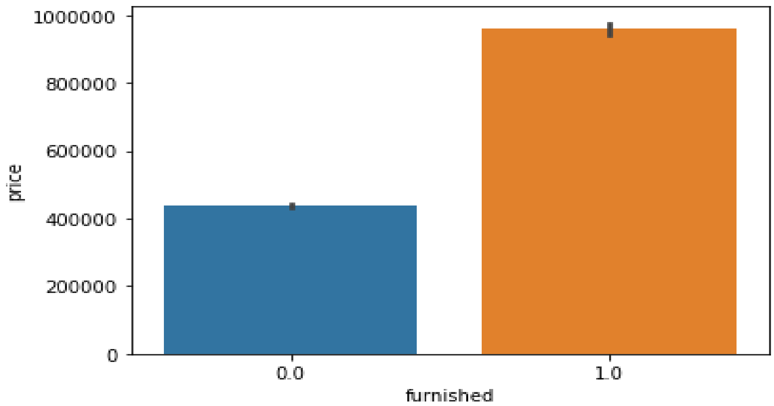
This column represents the living room area in the year 2015. Some houses are constructed quite recently, hence their living room area maybe high when compared to houses that were built in the year 1900. This also implies that some renovations are made. 197 properties have living room area of 1,540 square footage.

- **lot_measure15**



This column represents lot size area in the year 2015. 2,194 properties have 17,550 lot size, which are the highest. 7,620 lot size is the median value. This maybe due to some renovations made in the year 2015.

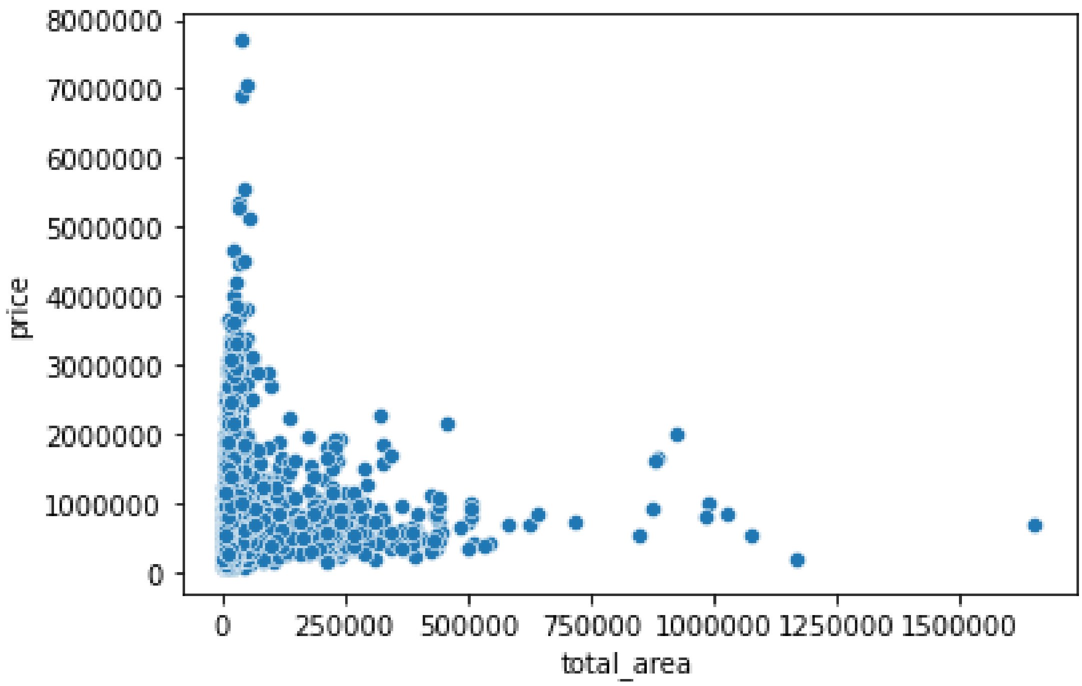
- **furnished**



furnished column is one of the most important variables. 17,338 properties have been furnished and 4,246 properties were not furnished.

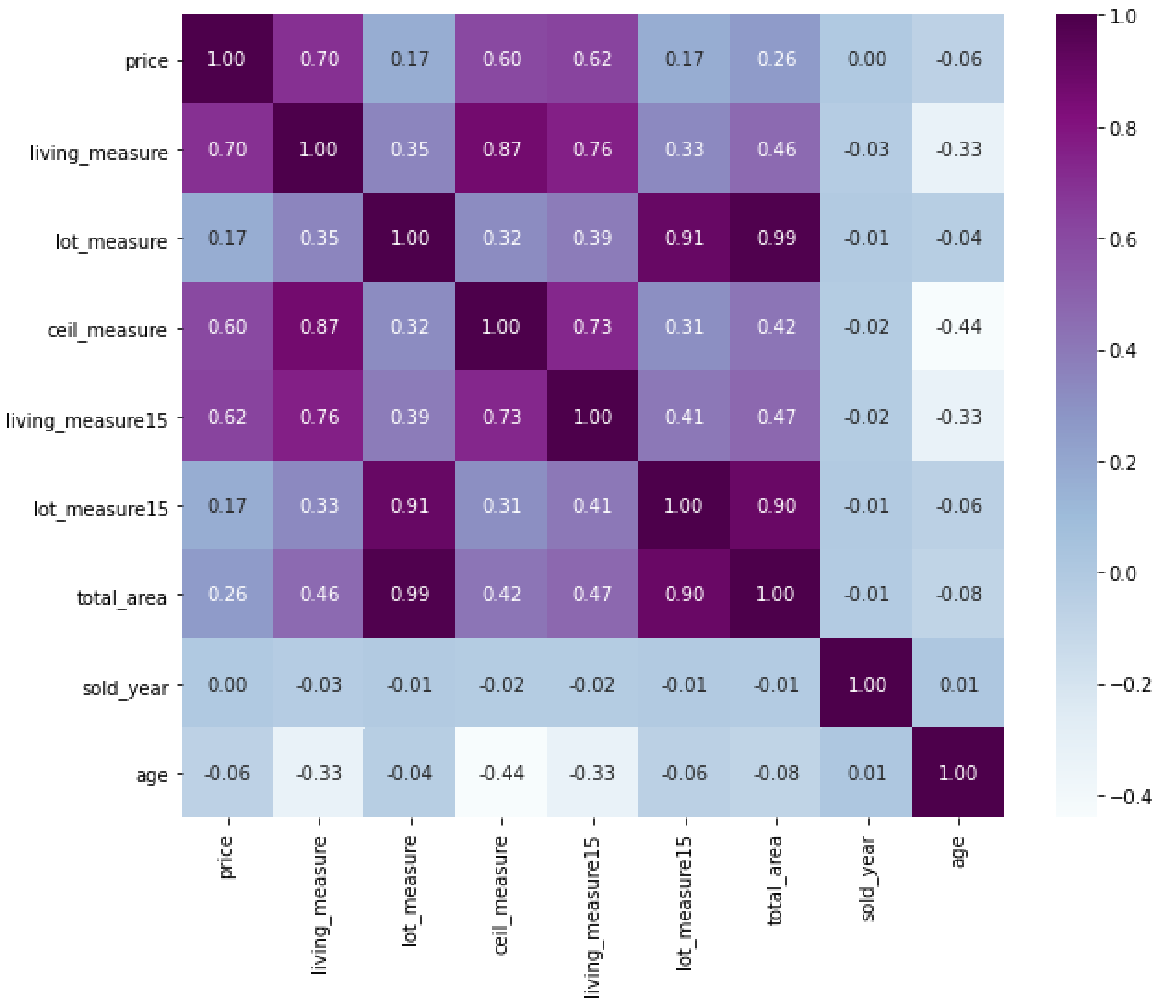
When we compare it against price, it is quite clear that houses that are furnished always fetch good price in the market, whilst the unfurnished properties don't attract the customers/buyers and fetch only low price.

- **total_area**



This column represents the total measurement of the property, i.e., both house and lot. Comparing against price, the insight is not clear whether price of the property may increase or decrease based on total area.

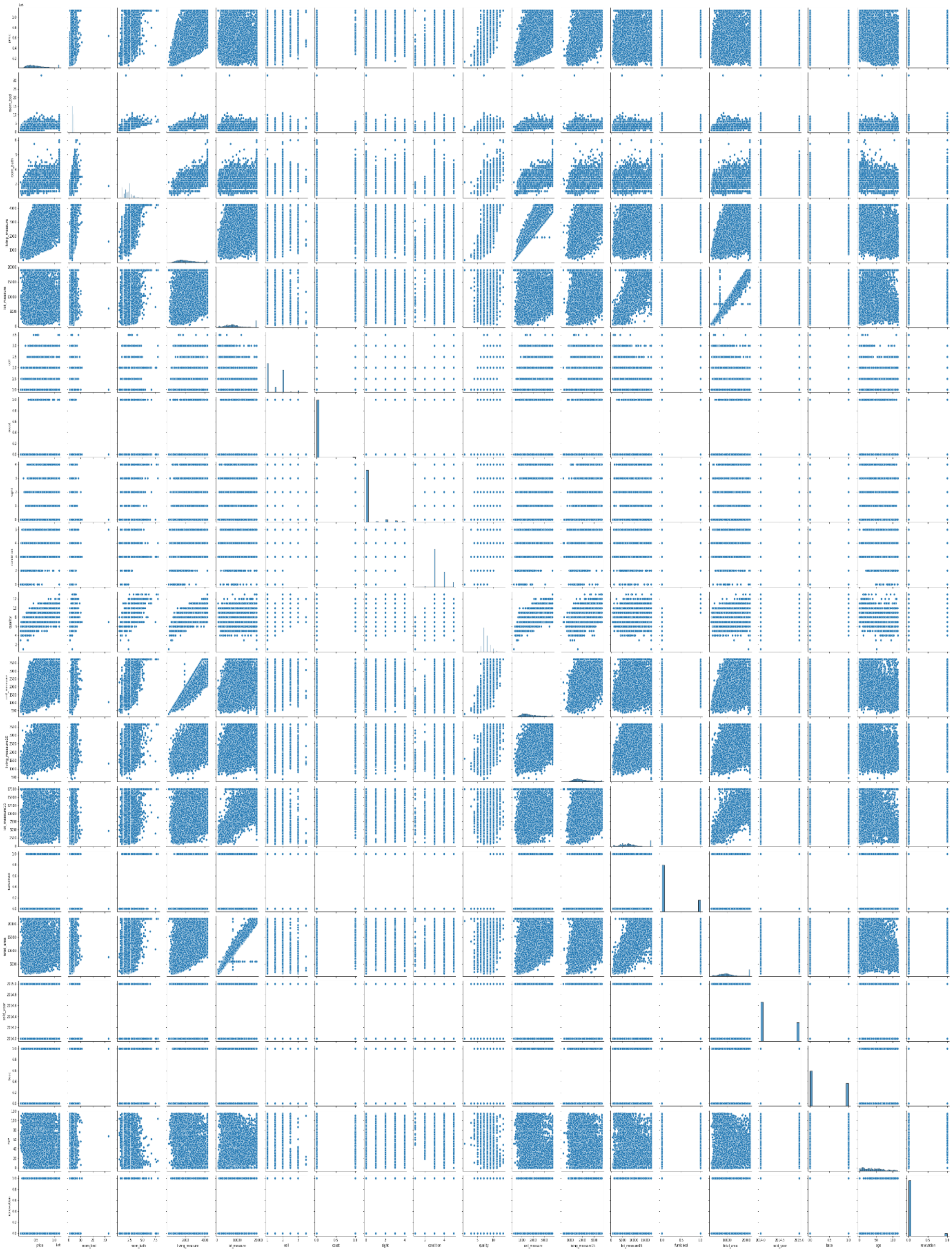
3.4 Multivariate Analysis



Correlation ranges from -1 to +1. Values close to 0 means, there is no linear trend between the two variables. Values close to 1 shows correlation and how much positively correlated, i.e., as one increases so does the other and the closer to 1 the stronger the relationship.

lot_measure15 and total_area are highly correlated against lot_measure. Similarly ceil_measure and living_measure15 are highly correlated against living_measure variable.

Pairplot



4. Data Cleaning and Pre-processing (Outliers)

```
cid                  0
dayhours             0
price                0
room_bed           108
room_bath          108
living_measure      17
lot_measure         42
ceil                42
coast                1
sight               57
condition           57
quality              1
ceil_measure         1
basement             1
yr_built             1
yr_renovated         0
zipcode              0
lat                  0
long                 0
living_measure15   166
lot_measure15       29
furnished           29
total_area          29
dtype: int64
```

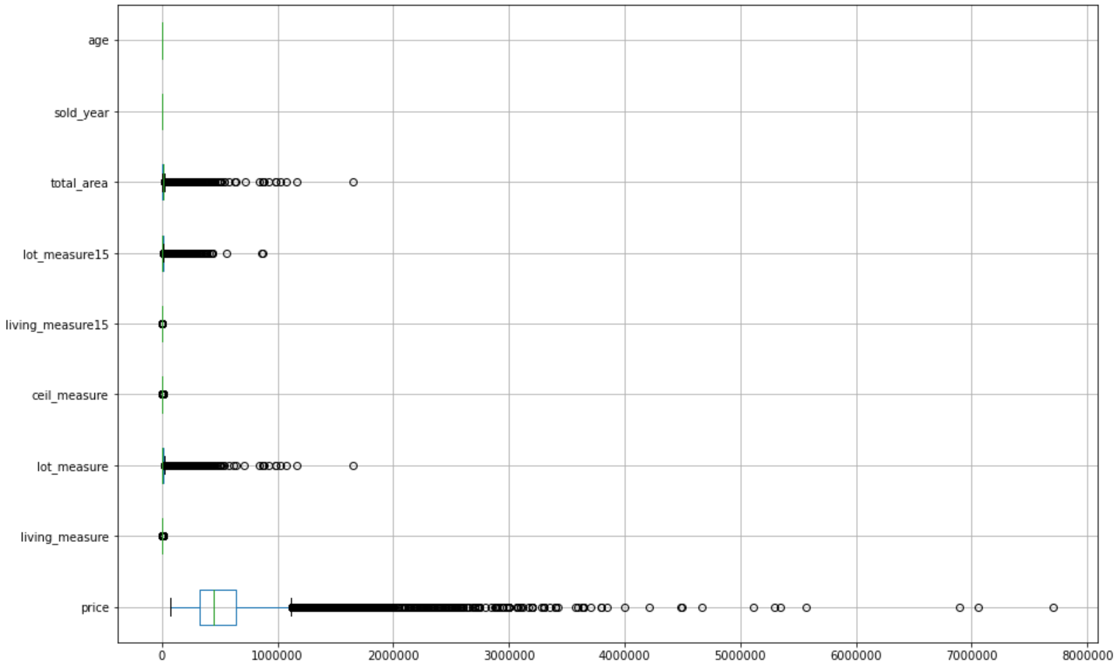
```
cid                 0
dayhours            0
price               0
room_bed            0
room_bath           0
living_measure      0
lot_measure         0
ceil                0
coast               0
sight               0
condition           0
quality             0
ceil_measure        0
basement            0
yr_built            0
yr_renovated        0
zipcode             0
lat                 0
long                0
living_measure15    0
lot_measure15       0
furnished           0
total_area          0
sold_year           0
base                0
age                 0
renovation          0
dtype: int64
```

- Ignoring missing values could lead to poor decisions which results in incorrect implementation of the data. Hence missing value treatment is mandatory before building the model.

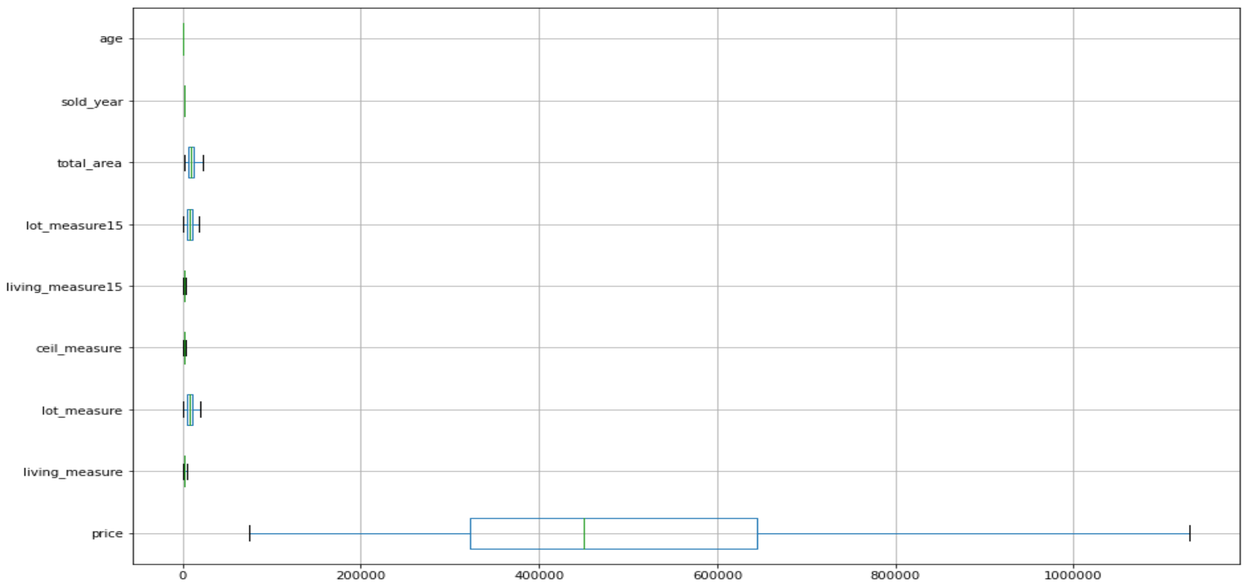
- Firstly, special characters in the dataset are converted to missing values. And for categorical variables, the missing values are imputed/replaced with mode. For numerical variables, the missing values are imputed with median.

- Outliers are data that differs significantly from other observations. It indicates bad data or abnormal data. It could be a result of human error too. The central tendency, Mean, is affected by outliers. Dataset free from outliers produce high accuracy.

- Outlier detection is done with the help of boxplot



- Here outlier treatment is done by flooring/capping method. Capping is replacing all upper range values exceeding 75th percentile by upper control limit value. Flooring is replacing all lower range values falling below 25th percentile by lower control limit value.



4.1 Variable inclusion and removal

Variables sold_year, base, age, renovation are added by pre-processing the columns dayhours, basement, yr_built, yr_renovated. By comparing the additional variables against the Target variable, the insight has become quite clear for us to deduce.

Variables such as cid, dayhours, basement, yr_built, yr_renovated, zipcode, lat and long are removed since we can't able to deduce any insights and model building will become simplified.

4.2 Variable Transformation

For some columns, variable transformation is done in order to fit the regression models. Columns such as room_bed, room_bath, ceil, coast, sight, condition, quality, base, furnished, renovation has categorical features.

Hence these variables are converted to categorical and dummy encoding is applied before model building.

4.3 Scaling

Scaling is necessary before model building. Since some of the variable values are far from each other, we are going to scale the dataset using Standard Scaler. This also comes under data pre-processing techniques. This converts the data points to be within a particular range, which is easy for algorithmic calculations.

5. Model building

The target variable – price is continuous. Hence, we're implementing Regression Algorithms. Splitting the target variable into train and test. 70% train and 30% test.

price	living_measure	lot_measure	ceil_measure	living_measure15	lot_measure15	total_area	sold_year	age	room_bed_2.0	...	quality_7.0
0.35	1.18	0.15	0.04	0.07	0.08	0.32	1.45	0.19	0	...	0
-1.29	-1.65	-1.11	-1.44	-0.49	-0.96	-1.29	1.45	0.81	1	...	0
0.89	1.17	-1.25	1.66	1.00	-1.35	-0.98	-0.69	0.16	0	...	0
-1.02	-0.38	-0.99	-0.04	0.09	-1.03	-0.98	-0.69	-1.30	0	...	0
-0.25	-1.12	-0.82	-0.85	-1.32	-0.73	-0.93	1.45	1.62	1	...	1

The regression algorithms that we applied to find the best fit model are:

- Linear Regression
- Random Forest
- KNN
- Decision Tree

For better predictions, we have used Adaboosting and Gradient Boosting ensembling techniques.

Model	Train Score	Test Score	Adj. R Train	Adj. R Test	MSE Train	MSE Test	RMSE Train	RMSE Test	MAE Train	MAE Test
Linear Regression	0.69	0.69	0.69	0.68	0.31	0.31	0.56	0.55	0.43	0.43
Random Forest	0.96	0.73	0.96	0.73	0.04	0.26	0.19	0.51	0.14	0.38
KNN	0.70	0.53	0.69	0.53	0.31	0.46	0.56	0.68	0.42	0.51
DT	1.00	0.47	1.00	0.47	0.00	0.51	0.01	0.72	0.00	0.52
ADB	0.55	0.53	0.55	0.53	0.46	0.46	0.68	0.68	0.55	0.55
GB	0.74	0.72	0.73	0.71	0.27	0.27	0.52	0.52	0.40	0.40

5.1 Evaluation Parameters

MSE – Mean Squared Error – average of the squared difference between the actual and the predicted value. Lower the MSE score, better the model.

RMSE – Root Mean Squared Error – square root of the averaged squared difference between the actual and the predicted value. Lower the RMSE score, better the model.

MAE – Mean Absolute Error – absolute difference between the actual and the predicted value. Lower the MAE score, better the model.

R2 – Coefficient of Determination – statistical measure of how well the regression predictions approximate the real data points. Higher the R2 score, better the model.

6. Model Validation

Model Performance	Linear Regression	Random Forest	KNN	Decision Tree	Adaboosting	Gradient Boosting
R square for train set	69.13%	96.28%	69.52%	99.99%	53.76%	73.62%
R square for test set	68.61%	73.43%	53.11%	46.76%	52.19%	71.80%
Adjusted R square for train set	68.98%	96.27%	69.36%	99.98%	53.53%	73.49%
Adjusted R square for test set	68.24%	73.12%	52.56%	46.14%	51.63%	71.47%
MSE for train set	31.20%	3.76%	30.81%	0.02%	46.73%	26.66%
MSE for test set	30.61%	25.90%	45.72%	51.91%	46.61%	27.50%
RMSE for train set	55.85%	19.38%	55.51%	1.23%	68.36%	51.63%
RMSE for test set	55.32%	50.89%	67.61%	72.05%	68.27%	52.44%
MAE for train set	42.80%	14.39%	42.14%	0.05%	56.67%	39.79%
MAE for test set	42.59%	38.22%	51.46%	52.51%	56.37%	40.43%

Model validation is based on the accuracy of a model.

- Out of all the models, Random Forest shows better results with train score of 96.28% and test score of 73.43%.
- Random forest is also an ensemble technique, which is suitable for complex datasets, which gives high accuracy and reduces model overfitting.
- The main drawback of RF model is the slow processing speed.
- We can further improve this model by hyper-tuning with GridSearchCV

6.1 Tuning

Tuning the Random Forest model using GridSearchCV using various parameters.

```
GridSearchCV(cv=5, estimator=RandomForestRegressor(),
             param_grid={'n_estimators': [100, 150, 200, 250]})
```

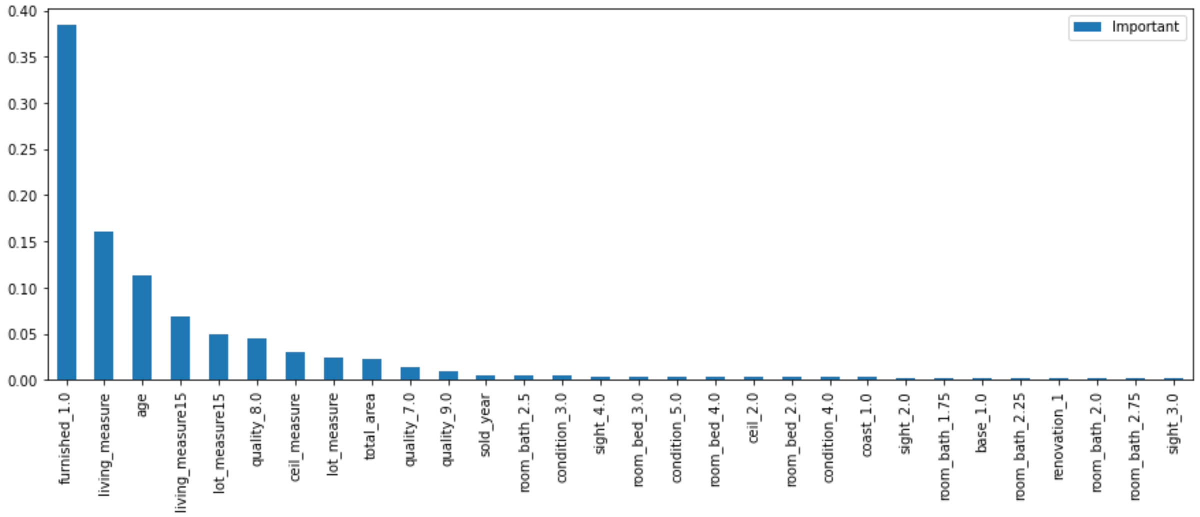
```
The best estimator across ALL searched params:
RandomForestRegressor(n_estimators=250)

The best score across ALL searched params:
0.716240785198706

The best parameters across ALL searched params:
{'n_estimators': 250}
```

Tuning the model lead to a score of 71.62%, which is much lower compared to the original R2 score of 96.28%.

7. Feature Importance



First 20 important features give accuracy of 95.59%.

First 30 important features give accuracy of 97.92%.

	Important
furnished_1.0	0.38
living_measure	0.16
age	0.11
living_measure15	0.07
lot_measure15	0.05
quality_8.0	0.05
ceil_measure	0.03
lot_measure	0.02
total_area	0.02
quality_7.0	0.01
quality_9.0	0.01
sold_year	0.01
room_bath_2.5	0.00
condition_3.0	0.00
sight_4.0	0.00

- The most important feature for pricing is furnished_1.0, i.e., the houses which are furnished are highly priced.
- The second most important feature is living_measure. The price of houses that have large square footage are high.
- The third most important feature is age of a house, i.e., yr_built. The houses which were recently built are also a factor for price increase.
- Other important features such as living_measure15, lot_measure15, quality_8.0, ceil_measure, lot_measure, total_area, quality_7.0, quality_9.0, and sold_year are to be taken into consideration for price increase of a house in the market.
- The top 30 important features are covering about 97% of variation in the Random Forest model. This is really a good coverage for just 30% of the variables.

8. Recommendation & Insights

- It is important to keep in mind about the main predictors while taking major decisions.
- Recession is a main factor that indicates unemployment, results in incomes fall and consumers lack the confidence to make a huge investment in the housing market.
- Natural calamities like flood are one of the main reasons to avoid properties surrounded by waterbodies.
- Infrastructure is one of the most important factors of price in property or real estate. Properties located near airport, railways, would fetch high price.
- IT hubs, shopping malls, entertainment zones could elevate the price of a real estate property.
- High interest rates lead to a decrease in demand for a property by home buyer. Hence when interest rates undergo a decrease, demand for properties increases, thus resulting in decreased prices.