**House Price Prediction**

Name: Saurabh Dharmadhikari

PGP-DSBA Online January’ 21

Capstone Project

Date: 27/11/2022

Table of Contents:

Executive Summary……………………...................................................................................................................................3

Introduction ...................................................................................................................................................................3

Data Description ............................................................................................................................................................3

Sample of the dataset ...................................................................................................................................................4

Exploratory Data Analysis..............................................................................................................................................5

Let us check the types of variables in the data frame. .................................................................................................5

Null Values ....................................................................................................................................................................6

Discrepancy in room\_bed variable. ..............................................................................................................................6

Box Plot for room\_bed and price..................................................................................................................................7

Box plot for room\_bath and price …………………………………………………………………………………………………………...................7

Sight and coast price analysis …………………………………………………………………………………………….......................................8

Data after cleaning …………………………………………………………………………..........................................................................9

Total area and price analysis …...................................................................................................................................11

Latitude and longitude analysis with price ……………………………………………………………………………………………………………11

Pair plot with cleaned data …………………………………………………………………………………………………………………………………. 12

Heat map for cleaned data …………………………………………………………………………………………………………………………………. 13

Insights gained …………………………………………………………………………………………………………………………………………………… 13

What models we are using and why …………………………………………………………………………………………………………………… 17

Model score before optimisation ….……………………………………………………………………………………………………………………. 17

Parameters used for optimisation …….…………………………………………………………………………………………….…………………. 18

Model score after optimisation ……………………………………………………………………………………………………………………….…. 18

Model score after scaling the data ……………………………………………………………………………………………………………………... 19

Feature importance …………………………………………………………………………………………………………………………….. 20

Insights/ recommendations ……………………………………………………………………………………………………………….. 21

End …………………………………………………………………………………………………………………………………………………….. 22

Executive Summary

We as an analyst are expected to do an exploratory data analysis (EDA) of the data given to us with various variables and prices of each house. These houses have been built from 1900 to 2015. We must be able to clean the data in such a way that we are able to easily visualise and understand the impact of these features provided to us on the prices of the houses. We should be able to present the data in order to gain better insights from the data given to us.

Introduction

The purpose of this whole exercise is to explore the dataset. Do the exploratory data analysis. The data consists of 21613 entries and 23 features. We have an entry of house built starting from January 1900 to 2015. Each house has may have gone under renovation for which we have data as on 2015. Thus, plot size may have also changed as per new government regulations or any renovation projects undertaken for the property. We are also provided with other features like area zip code, latitude and longitude for location and rooms in the house as explained in data description.

Data Description

1. **cid:** a notation for a house
2. **dayhours:** Date house was sold
3. **price:** Price is prediction target
4. **room\_bed:** Number of Bedrooms/House
5. **room\_bath:** Number of bathrooms/bedrooms
6. **living\_measure:** square footage of the home
7. **lot\_measure:** quare footage of the lot
8. **ceil:** Total floors (levels) in house
9. **coast:** House which has a view to a waterfront
10. **sight:** Has been viewed
11. **condition:** How good the condition is (Overall)
12. **quality:** grade given to the housing unit, based on grading system
13. **ceil\_measure:** square footage of house apart from basement
14. **basement:** square footage of the basement
15. **yr\_built:** Built Year
16. **yr\_renovated:** Year when house was renovated
17. **zipcode:** zip
18. **lat:** Latitude coordinate
19. **long:** Longitude coordinate
20. **living\_measure15:** Living room area in 2015(implies-- some renovations) This might or might not have affected the lotsize area
21. **lot\_measure15:** lotSize area in 2015(implies-- some renovations)
22. **furnished:** Based on the quality of room 23: total\_area: Measure of both living and lot
23. **total\_area:** Total area of the plot

Sample of the data set

|  | **cid** | **dayhours** | **price** | **room\_bed** | **room\_bath** | **living\_measure** | **lot\_measure** | **ceil** | **coast** | **sight** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 3876100940 | 20150427T000000 | 600000 | 4.0 | 1.75 | 3050.0 | 9440.0 | 1.0 | 0.0 | 0.0 |
| **1** | 3145600250 | 20150317T000000 | 190000 | 2.0 | 1.00 | 670.0 | 3101.0 | 1.0 | 0.0 | 0.0 |
| **2** | 7129303070 | 20140820T000000 | 735000 | 4.0 | 2.75 | 3040.0 | 2415.0 | 2.0 | 1.0 | 4.0 |
| **3** | 7338220280 | 20141010T000000 | 257000 | 3.0 | 2.50 | 1740.0 | 3721.0 | 2.0 | 0.0 | 0.0 |
| **4** | 7950300670 | 20150218T000000 | 450000 | 2.0 | 1.00 | 1120.0 | 4590.0 | 1.0 | 0.0 | 0.0 |

Above are the first 5 rows of first 10 columns.

|  | **quality** | **ceil\_measure** | **basement** | **yr\_built** | **yr\_renovated** | **zipcode** | **lat** | **long** | **living\_measure15** | **lot\_measure15** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 8.0 | 1800.0 | 1250.0 | 1966.0 | 0 | 98034 | 47.7228 | -122.183 | 2020.0 | 8660.0 |
| **1** | 6.0 | 670.0 | 0.0 | 1948.0 | 0 | 98118 | 47.5546 | -122.274 | 1660.0 | 4100.0 |
| **2** | 8.0 | 3040.0 | 0.0 | 1966.0 | 0 | 98118 | 47.5188 | -122.256 | 2620.0 | 2433.0 |
| **3** | 8.0 | 1740.0 | 0.0 | 2009.0 | 0 | 98002 | 47.3363 | -122.213 | 2030.0 | 3794.0 |
| **4** | 7.0 | 1120.0 | 0.0 | 1924.0 | 0 | 98118 | 47.5663 | -122.285 | 1120.0 | 5100.0 |

Above are the first 5 rows of column 11 to column 20.

|  | **furnished** | **total\_area** |
| --- | --- | --- |
| **0** | 0.0 | 12490.0 |
| **1** | 0.0 | 3771.0 |
| **2** | 0.0 | 5455.0 |
| **3** | 0.0 | 5461.0 |
| **4** | 0.0 | 5710.0 |

Above are last two columns.

Exploratory Data Analysis:

Info on columns of data:

Let us see more information about these features provided to us.

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)

We can observe from the above chart that there are certain data entries that are missing in some columns like room\_bed, room\_bath and others. We also see that there are a total 12 features that are described as float, 4 as integer type and 7 as objects.

**Null Values:**

Let us see total values that are missing in the dataset we have so as to ascertain how we might be able to treat them.

living\_measure15 166

room\_bed 108

room\_bath 108

condition 85

ceil 72

total\_area 68

sight 57

lot\_measure 42

long 34

coast 31

lot\_measure15 29

furnished 29

living\_measure 17

yr\_built 15

ceil\_measure 1

basement 1

quality 1

dtype: int64

So, as we see there are a total number so 166 values missing in living\_measure15 and 108 each in room\_bed and room\_bath. There are other features as well where the values are missing but are not very high in the number of occurrences.

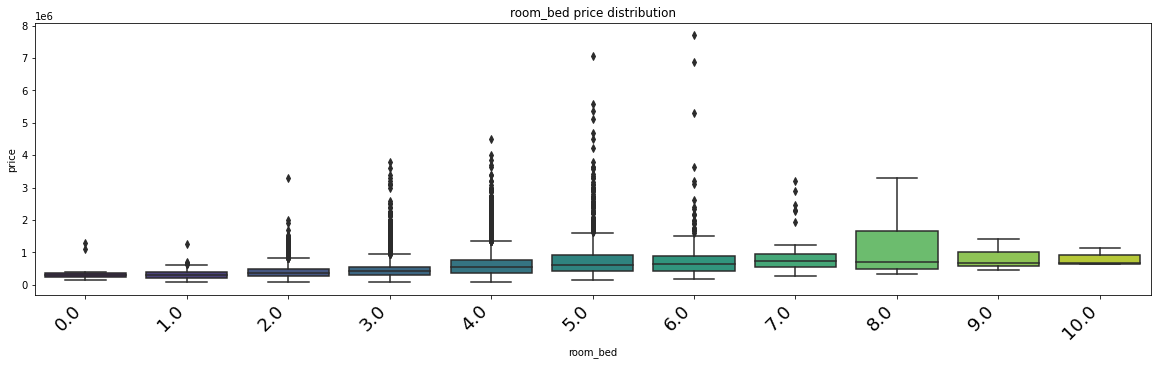
**Discrepancy in room\_bed variable:**

We will treat missing values of living \_measure with the mean or average value and for room\_bed and room\_bath we will use mode (most occurring value) so as to not disturb the variation.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **16913** | 2402100895 | 20140625T000000 | 640000 | 33.0 | 1.75 | 1620.0 | 6000.0 | 1.0 |
| **20972** | 1773100755 | 20140821T000000 | 520000 | 11.0 | 3.0 | 3000.0 | 4960.0 | 2.0 |

We also find few outliers like above at entry 16913 and 20972 with 33 and 11 number of bed rooms respectively. Which dose not seem possible after observing ceiling, lot\_measure and living\_measure. So, we will be changing the number of bedrooms for these houses to 9.

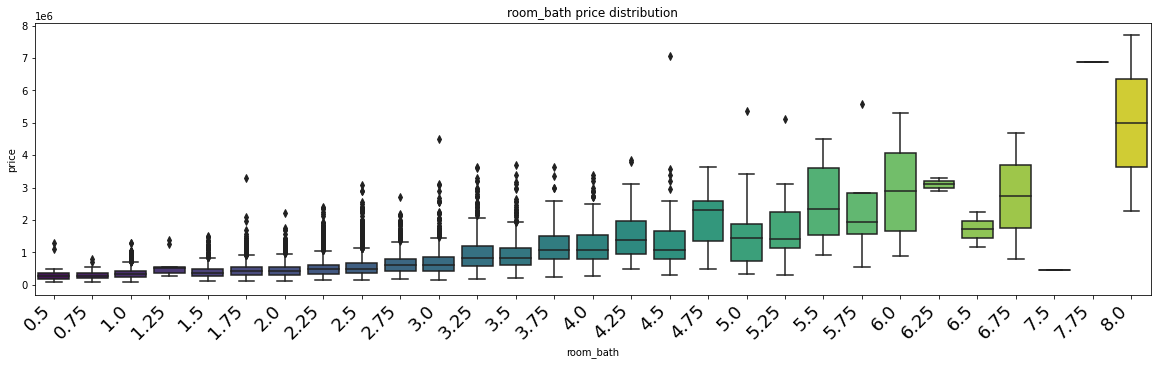
**Box Plot for room\_bed and price:**



Through above boxplot we find that that the prices have been increasing with the increase in number of bed rooms but prices also decrease when number of bedrooms are very high like houses with 9 or 10.

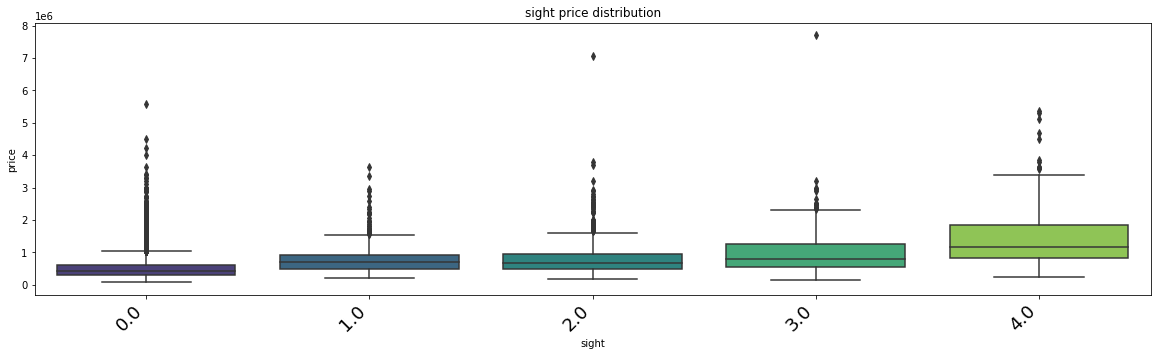
**Box plot for room\_bath and price:**

### We will also impute the number of bedrooms with 0 value to 1. Also, assuming that there is at least 1 bathroom in each property, we replace 0 bathroom with 1.

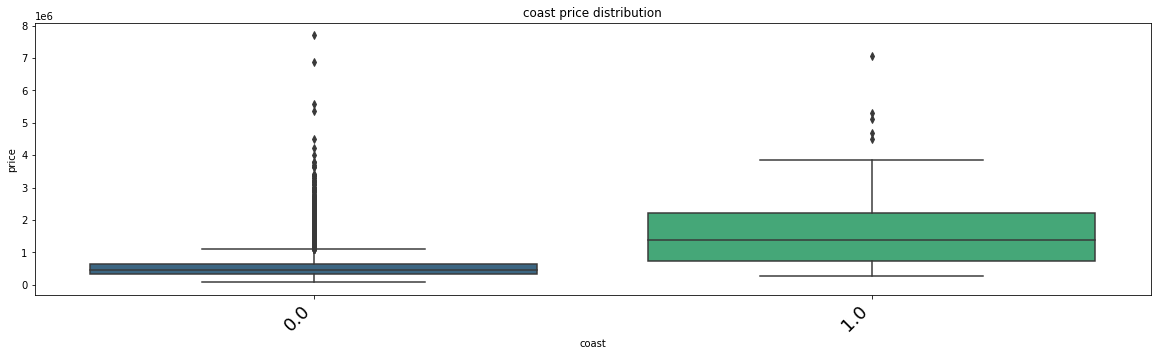


Again, with above box plot we can easily see that there is a steady increase in prices as the number of bathrooms increase.

**lot\_measure and living\_measures are the areas at the time of purchase and we see a lot has changed after that. We have recent data of area in form of living\_measure15 and lot\_measure15, thus we will remove former features.**

****

Here we see that prices of those properties increase that have been viewed a greater number of times.



We can observe here that houses having a view of the coast or with water front have a greater price than those houses that don’t.

**Data after cleaning:**

Int64Index: 21375 entries, 0 to 21612

Data columns (total 17 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 price 21375 non-null int64

1 room\_bed 21375 non-null float64

2 room\_bath 21375 non-null float64

3 ceil 21375 non-null float64

4 coast 21375 non-null float64

5 sight 21375 non-null float64

6 condition 21375 non-null float64

7 quality 21375 non-null float64

8 ceil\_measure 21375 non-null float64

9 basement 21375 non-null float64

10 yr\_built 21375 non-null float64

11 yr\_renovated 21375 non-null int64

12 zipcode 21375 non-null int64

13 living\_measure15 21375 non-null int32

14 lot\_measure15 21375 non-null float64

15 furnished 21375 non-null float64

16 total\_area 21375 non-null float64

dtypes: float64(13), int32(1), int64(3)

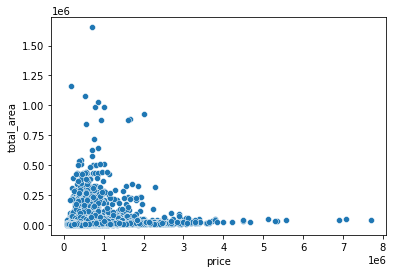
Now we are left with only 17 features and a total of 21375 after cleaning the data.

|  | **count** | **mean** | **std** | **min** | **25%** | **50%** | **75%** | **max** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **price** | 21375.0 | 540241.969965 | 367874.730779 | 75000.0 | 321500.00 | 450000.00 | 645000.0 | 7700000.0 |
| **room\_bed** | 21375.0 | 3.368982 | 0.907091 | 0.0 | 3.00 | 3.00 | 4.0 | 10.0 |
| **room\_bath** | 21375.0 | 2.116094 | 0.768488 | 0.5 | 1.75 | 2.25 | 2.5 | 8.0 |
| **ceil** | 21375.0 | 1.494620 | 0.540291 | 1.0 | 1.00 | 1.50 | 2.0 | 3.5 |
| **coast** | 21375.0 | 0.007439 | 0.085928 | 0.0 | 0.00 | 0.00 | 0.0 | 1.0 |
| **sight** | 21375.0 | 0.234433 | 0.766151 | 0.0 | 0.00 | 0.00 | 0.0 | 4.0 |
| **condition** | 21375.0 | 3.409450 | 0.650647 | 1.0 | 3.00 | 3.00 | 4.0 | 5.0 |
| **quality** | 21375.0 | 7.658012 | 1.176285 | 1.0 | 7.00 | 7.00 | 8.0 | 13.0 |
| **ceil\_measure** | 21375.0 | 1789.011556 | 828.681811 | 290.0 | 1190.00 | 1560.00 | 2210.0 | 9410.0 |
| **basement** | 21375.0 | 291.254269 | 442.448754 | 0.0 | 0.00 | 0.00 | 560.0 | 4820.0 |
| **yr\_built** | 21375.0 | 1971.009825 | 29.385505 | 1900.0 | 1951.00 | 1975.00 | 1997.0 | 2015.0 |
| **yr\_renovated** | 21375.0 | 84.035275 | 400.844781 | 0.0 | 0.00 | 0.00 | 0.0 | 2015.0 |
| **zipcode** | 21375.0 | 98077.913637 | 53.501136 | 98001.0 | 98033.00 | 98065.00 | 98118.0 | 98199.0 |
| **living\_measure15** | 21375.0 | 1987.036211 | 685.370798 | 399.0 | 1490.00 | 1840.00 | 2360.0 | 6210.0 |
| **lot\_measure15** | 21375.0 | 12762.807766 | 27287.123829 | 651.0 | 5100.00 | 7620.00 | 10083.0 | 871200.0 |
| **furnished** | 21375.0 | 0.197099 | 0.397817 | 0.0 | 0.00 | 0.00 | 0.0 | 1.0 |
| **total\_area** | 21375.0 | 17190.137216 | 41614.463353 | 1423.0 | 7036.50 | 9580.00 | 12997.0 | 1652659.0 |

We can see all the key features and their description.

We find that prices of a house start at 7500 and costliest house is for 7700000, with an average of 450000.

Also, minimum total area of a house is only 1423 square foot and maximum is 1652659.

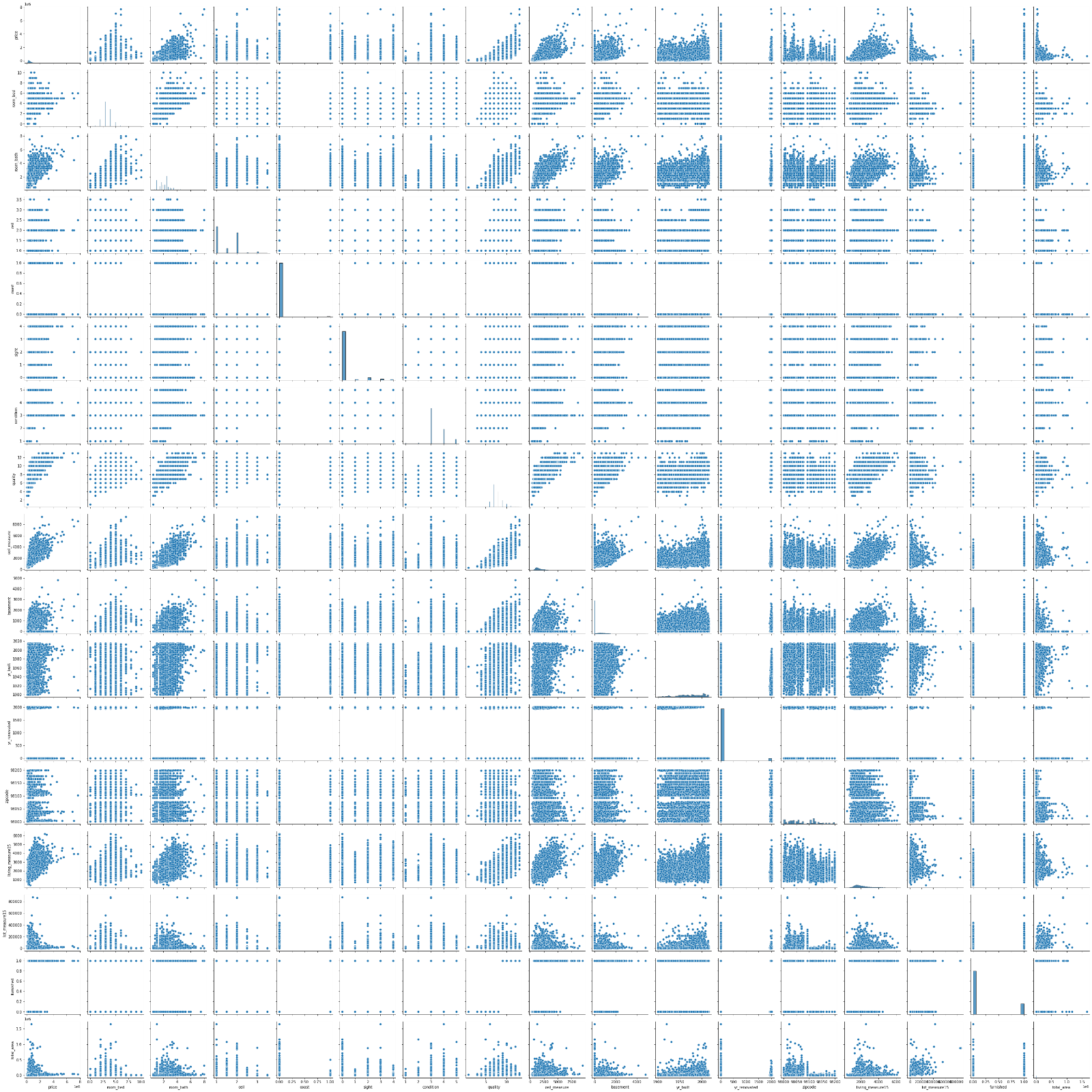


We find that as there is a positive relation between total area of a house and its price. That means whenever total area of a house increases value of that house also increases.



We see that certain longitude and latitude positions have higher price than others. This could be very usefull.

**Pairplot:**



We can see that there is mostly a positive relation between different variables.

Heat-map for cleaned data



We can observe that Correlations between different variable through this heatmap. Darker the box in the map more negatively the variables interact with each other. If the box is lighter in shade that means variables are positively interacting.

Insights Gained:

1. As discussed earlier we see that there is a direct relationship between area of the house and prices.
2. Number of bed rooms also effect the prices, till bedrooms increase till 7 prices increase but thereafter they start to decline.
3. Also, bathrooms have a direct relationship.
4. Houses in certain longitude and latitude also have a higher value.
5. Sight has a direct relationship.
6. Coast has a direct relationship with price.

Let us again see our data frame now:

|  | **price** | **room\_bed** | **room\_bath** | **ceil** | **coast** | **sight** | **condition** | **quality** | **ceil\_measure** | **basement** | **yr\_built** | **yr\_renovated** | **zipcode** | **living\_measure15** | **lot\_measure15** | **furnished** | **total\_area** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 600000 | 4.0 | 1.75 | 1.0 | 0.0 | 0.0 | 3.0 | 8.0 | 1800.0 | 1250.0 | 1966.0 | 0 | 98034 | 2020 | 8660.0 | 0.0 | 12490.0 |
| **1** | 190000 | 2.0 | 1.00 | 1.0 | 0.0 | 0.0 | 4.0 | 6.0 | 670.0 | 0.0 | 1948.0 | 0 | 98118 | 1660 | 4100.0 | 0.0 | 3771.0 |
| **2** | 735000 | 4.0 | 2.75 | 2.0 | 1.0 | 4.0 | 3.0 | 8.0 | 3040.0 | 0.0 | 1966.0 | 0 | 98118 | 2620 | 2433.0 | 0.0 | 5455.0 |

We will remove the built\_year as it is not necessary now because after furnishing (furnished) and renovating the house (yr\_renovated), the quality of house must also have increased and thus it will not be a very good determent of price of the house.

We must also covert zipcode into categories where the prices are more and places where prices are less in order to get a better estimate of the prices of the houses nearby.

We have divided zipcode areas in 5 categories each with 14 zip codes.

Let us see how are data set now looks with new improvements.

|  | **price** | **room\_bed** | **room\_bath** | **ceil** | **coast** | **sight** | **condition** | **quality** | **ceil\_measure** | **basement** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 600000 | 4.0 | 1.75 | 1.0 | 0.0 | 0.0 | 3.0 | 8.0 | 1800.0 | 1250.0 |
| **1** | 190000 | 2.0 | 1.00 | 1.0 | 0.0 | 0.0 | 4.0 | 6.0 | 670.0 | 0.0 |
| **2** | 735000 | 4.0 | 2.75 | 2.0 | 1.0 | 4.0 | 3.0 | 8.0 | 3040.0 | 0.0 |
| **3** | 257000 | 3.0 | 2.50 | 2.0 | 0.0 | 0.0 | 3.0 | 8.0 | 1740.0 | 0.0 |
| **4** | 450000 | 2.0 | 1.00 | 1.0 | 0.0 | 0.0 | 3.0 | 7.0 | 1120.0 | 0.0 |

|  | **yr\_built** | **yr\_renovated** | **zipcode** | **living\_measure15** | **lot\_measure15** | **furnished** |
| --- | --- | --- | --- | --- | --- | --- |
| **0** | 1966.0 | 0 | 3 | 2020 | 8660.0 | 0.0 |
| **1** | 1948.0 | 0 | 4 | 1660 | 4100.0 | 0.0 |
| **2** | 1966.0 | 0 | 4 | 2620 | 2433.0 | 0.0 |
| **3** | 2009.0 | 0 | 5 | 2030 | 3794.0 | 0.0 |
| **4** | 1924.0 | 0 | 4 | 1120 | 5100.0 | 0.0 |

Data columns (total 16 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 price 21375 non-null int64

1 room\_bed 21375 non-null float64

2 room\_bath 21375 non-null float64

3 ceil 21375 non-null float64

4 coast 21375 non-null float64

5 sight 21375 non-null float64

6 condition 21375 non-null float64

7 quality 21375 non-null float64

8 ceil\_measure 21375 non-null float64

9 basement 21375 non-null float64

10 yr\_built 21375 non-null float64

11 yr\_renovated 21375 non-null int64

12 zipcode 21375 non-null int64

13 living\_measure15 21375 non-null int32

14 lot\_measure15 21375 non-null float64

15 furnished 21375 non-null float64

dtypes: float64(12), int32(1), int64(3)

We are now only left with 16 columns and 21375 entries. With 12 columns having float values, 4 with int value.

Let us divide our data into test and train:

|  | **room\_bed** | **room\_bath** | **ceil** | **coast** | **sight** | **condition** | **quality** | **ceil\_measure** | **basement** | **yr\_built** | **yr\_renovated** | **zipcode** | **living\_measure15** | **lot\_measure15** | **furnished** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **13874** | 3.0 | 4.5 | 2.0 | 1.0 | 4.0 | 3.0 | 10.0 | 3740.0 | 1490.0 | 2005.0 | 0 | 1 | 3670 | 17826.0 | 1.0 |
| **4139** | 4.0 | 1.5 | 1.5 | 0.0 | 0.0 | 5.0 | 7.0 | 1080.0 | 200.0 | 1920.0 | 0 | 4 | 1470 | 5934.0 | 0.0 |
| **7933** | 3.0 | 2.5 | 2.0 | 0.0 | 0.0 | 3.0 | 9.0 | 2550.0 | 0.0 | 1989.0 | 0 | 5 | 2410 | 9250.0 | 1.0 |
| **9426** | 2.0 | 1.0 | 1.0 | 0.0 | 0.0 | 4.0 | 7.0 | 780.0 | 0.0 | 1946.0 | 1989 | 2 | 2200 | 67518.0 | 0.0 |
| **11512** | 2.0 | 1.0 | 1.0 | 0.0 | 0.0 | 3.0 | 5.0 | 900.0 | 0.0 | 1918.0 | 0 | 5 | 2060 | 6533.0 | 0.0 |

Above is a chart showing first 5 rows of our train data.

|  | **room\_bed** | **room\_bath** | **ceil** | **coast** | **sight** | **condition** | **quality** | **ceil\_measure** | **basement** | **yr\_built** | **yr\_renovated** | **zipcode** | **living\_measure15** | **lot\_measure15** | **furnished** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **20467** | 3.0 | 2.50 | 2.0 | 0.0 | 0.0 | 3.0 | 7.0 | 2230.0 | 0.0 | 1954.0 | 0 | 2 | 1310 | 4000.0 | 0.0 |
| **19514** | 3.0 | 2.50 | 2.0 | 0.0 | 0.0 | 3.0 | 7.0 | 1640.0 | 0.0 | 1992.0 | 0 | 2 | 1580 | 28399.0 | 0.0 |
| **5511** | 3.0 | 1.75 | 3.0 | 0.0 | 0.0 | 3.0 | 7.0 | 1350.0 | 0.0 | 1999.0 | 0 | 2 | 1520 | 1652.0 | 0.0 |
| **10359** | 3.0 | 2.50 | 2.0 | 0.0 | 0.0 | 3.0 | 8.0 | 1060.0 | 490.0 | 2006.0 | 0 | 2 | 1550 | 1301.0 | 0.0 |
| **18460** | 5.0 | 3.00 | 1.0 | 0.0 | 0.0 | 4.0 | 8.0 | 1370.0 | 950.0 | 1986.0 | 0 | 2 | 2090 | 7554.0 | 0.0 |

Above is a chart showing first 5 rows of our test data.

**We have used 4 models to predict house prices.**

1. Linear Regression: We have used linear regression because this model is widely used to predict variables based on the values of other variable and is established as a good predicting model for continuous data.
2. Decision Tree: This model is easy to understand.
3. Random Forest: We have used this model because it will use random feature subsets and averages the predictions on all the different decision trees created. Giving us better results.
4. Artificial Neural network: It works artificially like a human brain and creates artificial networks and make a decision. It learns complex relationship and is a non-linear form of learning.

We will be trying to see which of these models are best suited to predict prices of house properties.

Now we are going to run our models on our test set:

|  | **Train RMSE** | **Test RMSE** | **Training Score** | **Test Scores** |
| --- | --- | --- | --- | --- |
| **Linear Regression Model** | 196588.780640 | 170983.137274 | 0.733705 | 0.739797 |
| **Decision Tree Model** | 9294.560051 | 223136.400141 | 0.999405 | 0.556854 |
| **Random Forest Model** | 59758.265912 | 151868.221878 | 0.975394 | 0.794723 |
| **Artificial Neural Network Model** | 239916.746474 | 218853.051171 | 0.603388 | 0.573704 |

We can see that the best results generated are by Random Forest Model. With a test score of 79% accuracy but in training score we see that it is performing very well perhaps we should try and optimise our model.

Also, we get almost 99.99% accuracy in training score of decision tree but we only get an accuracy of 55.6% for test data.

It is safe to say that the models created are overfit because they perform well in training data but poor in test data.

This has to be optimised so that we get a balanced outcome of our predictions.

Optimising are models using best parameters

Parameters used for Decision Tree: {'max\_depth': 10, 'min\_samples\_leaf': 3

,'min\_samples\_split': 10}

Parameters used for Random Forest: {'max\_depth': 10,

'max\_features': 6,

'min\_samples\_leaf': 3,

'min\_samples\_split': 10,

'n\_estimators': 50}

Parameters used for Artificial Neural network: {'activation': 'relu', 'hidden\_layer\_sizes': 10, 'solver': 'adam'}

After finding best parameters for our model we fit and see the results thus generated:

|  | **Train RMSE** | **Test RMSE** | **Training Score** | **Test Scores** |
| --- | --- | --- | --- | --- |
| **Linear Regression Model** | 196588.780640 | 170983.137274 | 0.733705 | 0.739797 |
| **Decision Tree Model** | 134106.946595 | 192501.728747 | 0.876078 | 0.670182 |
| **Random Forest Model** | 130372.528905 | 150598.690068 | 0.882884 | 0.798141 |
| **Artificial Neural Network Model** | 281821.925707 | 244979.118902 | 0.452739 | 0.465849 |

We can now see that the models developed after optimising generalise better than the previous models developed as we used different parameters.

Hence after observing above chart, we can see that random forest model is giving us the best results with a test score of almost 80%.

Hence it will be best to use random forest model to predict on this data set.

**Scaling the data**

Let us also scale the data and run our models and see if it gets us better results.

We are using Standard Scaler to scale our data.

|  | **Train RMSE** | **Test RMSE** | **Training Score** | **Test Scores** |
| --- | --- | --- | --- | --- |
| **Linear Regression Model** | 196588.780640 | 170983.137274 | 0.733705 | 0.739797 |
| **Decision Tree Model** | 9294.560051 | 223136.400141 | 0.999405 | 0.556854 |
| **Random Forest Model** | 59758.265912 | 151868.221878 | 0.975394 | 0.794723 |
| **Artificial Neural Network Model** | 258761.769763 | 250177.731821 | 0.538634 | 0.442939 |

Above we see our model results after scaling the data we find linear regression that should be most effected with this still does not perform better let us optimise the data and check the results again.

|  | **Train RMSE** | **Test RMSE** | **Training Score** | **Test Scores** |
| --- | --- | --- | --- | --- |
| **Linear Regression Model** | 196588.780640 | 170983.137274 | 0.733705 | 0.739797 |
| **Decision Tree Model** | 134106.946595 | 192501.728747 | 0.876078 | 0.670182 |
| **Random Forest Model** | 130372.528905 | 150598.690068 | 0.882884 | 0.798141 |
| **Artificial Neural Network Model** | 301187.590836 | 258468.772429 | 0.374943 | 0.405404 |

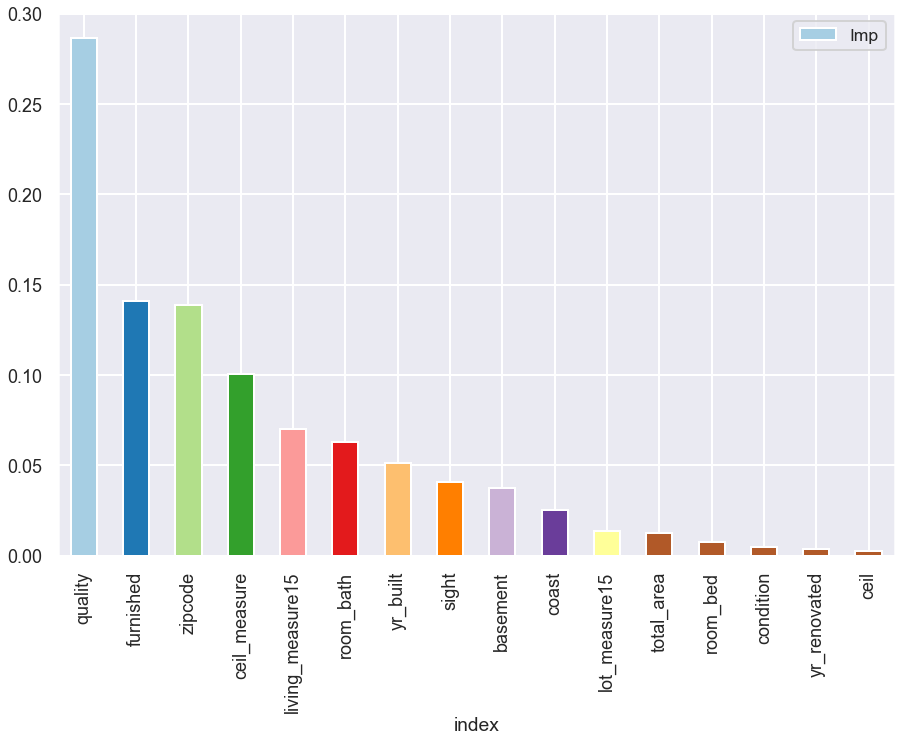
Above we see results after optimising and we find that results have not improved in comparison with unscaled data.

Thus, we can use unscaled data in our Random Forest model to give us best results.

Also, we should find out which feature is the most important feature in order to predict process of houses.

|  |  | **Imp** | **index** |
| --- | --- | --- | --- |
| **quality** |  | 0.286408 | quality |
| **furnished** |  | 0.141010 | furnished |
| **zipcode** |  | 0.138928 | zipcode |
| **ceil\_measure** |  | 0.100440 | ceil\_measure |
| **living\_measure15** |  | 0.070141 | living\_measure15 |
| **room\_bath** |  | 0.063091 | room\_bath |
| **yr\_built** |  | 0.051494 | yr\_built |
| **sight** |  | 0.040969 | sight |
| **basement** |  | 0.037372 | basement |
| **coast** |  | 0.025239 | coast |
| **lot\_measure15** |  | 0.013517 | lot\_measure15 |
| **total\_area** |  | 0.012449 | total\_area |
| **room\_bed** |  | 0.007623 | room\_bed |
| **condition** |  | 0.004761 | condition |
| **yr\_renovated** |  | 0.003934 | yr\_renovated |
| **ceil** |  | 0.002624 | ceil |

We can find from the above chart that the most important feature for predicting house prices is quality of the property and then followed by zip code (locality of the property) and living measure (living area) respectively.The least important feature turns out to be ceil (levels of property or number of floors) and yr\_renovated.



We can see the same with the help of a bar plot above to understand it better.

**Insight:**

Recommendations for seller:

* Seller should invest in quality of the house to get more price for the house as it has a very big impact on the price of the house.
* He can also invest in furnishing the house to increase the price of the property.
* He should also not try to increase the floors because it will take a lot of investment but the process will not increase as much.
* He should try bring more people to view the property as we see that it has an impact on the price of the property, probably because there are more bidders for the house.

Recommendations for buyer:

* A buyer should check for a different location or property in another zip code to have the same level of quality and furnishing in a house.
* Buyer may also compromise on number of bath rooms to get a lower price for the property.

Recommendations to bank:

* Banks can see the approximation value of the property and decide whether to give loan to a particular client or not, thus helping in reducing the risk.
* Banks may give a higher loan amount against a property with high quality.

Recommendations for advertising a property:

* Advertising firms may put up hoardings in zip code areas with similar property pricing. For example, zip codes like 98039, 98004 have high property prices thus similar clients may be residing in these areas and will be interested in properties similar to what they have.
* Thus, this will help them to target their client audience.