Introduction

In the filed of machine learning and data mining, the perdiction of house price is very common and chose to make a program that predict the house price based on dataset of a huge records.

The program we made was designed to be an html page that required some data from user describe the some features of the house he wants and predict the price of the house he wanted all based on dataset of 13320 record using data mining algorithms.

The model building

First data set downloaded from https://www.kaggle.com/amitabhajoy/bengaluru-house-price-data and is consisting of (['area_type', 'availability', 'location', 'size', 'society', 'total_sqft', 'bath', 'balcony', 'price'] columns and 13320 rows

Second data cleaning started with droping unnesesary columns as 'area_type', 'availability', 'balcony', 'society'.

Then the 'total_sqft' filed which contains non-specific values so we needed to get a mean average of the original value which was a rang of minimum and maximum values (e.g. 3067 – 8156), note that not all row are equal, but some have a single value and others have a rang of tow values as shown. And we builed a function to convert those rang values to a single average valu.

As the houses price is very dependent on price per the total square area so we needed this new feature insid the csv file, as the location filed contains lots of values so We need to apply dimensionality reduction technique here to reduce number of locations by declaring any location has less than 10 records to get a 240 from 1287 unique location.

After the data cleaning, the outlier removal using standard deviation of the new feature “price_per_sqft” to remove any values greater than mean minus standred deviation and less than mean plus standred deviation

Then removing any record whose bedroom area less than 300 ft as a standard policy very common in pricing houses market.

After finishing from the cleaning stage we can now build our model by using a common algorithm(GridSearchCV) who is going to compare between the score of linear_regression, lasso and decision_tree o and the result shows that linear regression was the best of them

The dataset

Dataset was downloaded from

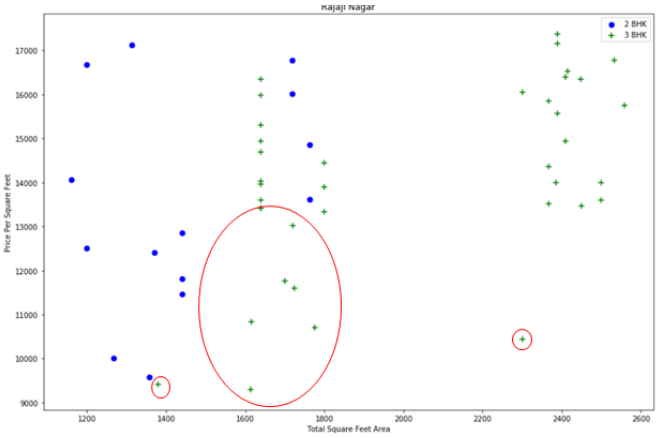
	area_type	availability	location	size	society	total_sqft	bath	balcony	price
0	Super built-up Area	19-Dec	Electronic City Phase II	2 BHK	Coomee	1056	2.0	1.0	39.07
1	Plot Area	Ready To Move	Chikka Tirupathi	4 Bedroom	Theanmp	2600	5.0	3.0	120.00
2	Built-up Area	Ready To Move	Uttarahalli	3 BHK	NaN	1440	2.0	3.0	62.00
3	Super built-up Area	Ready To Move	Lingadheeranahalli	3 BHK	Soiewre	1521	3.0	1.0	95.00
4	Super built-up Area	Ready To Move	Kothanur	2 BHK	NaN	1200	2.0	1.0	51.00

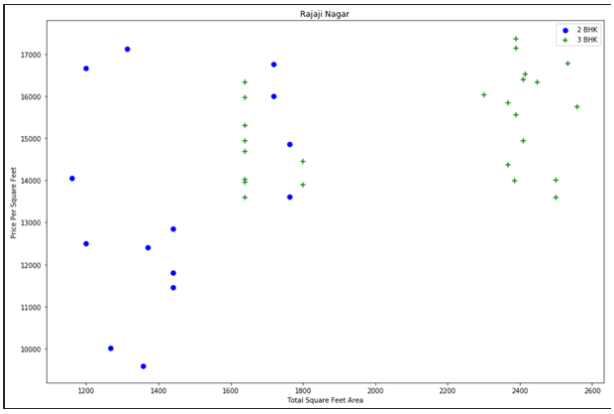
After data cleaning and feature engenering

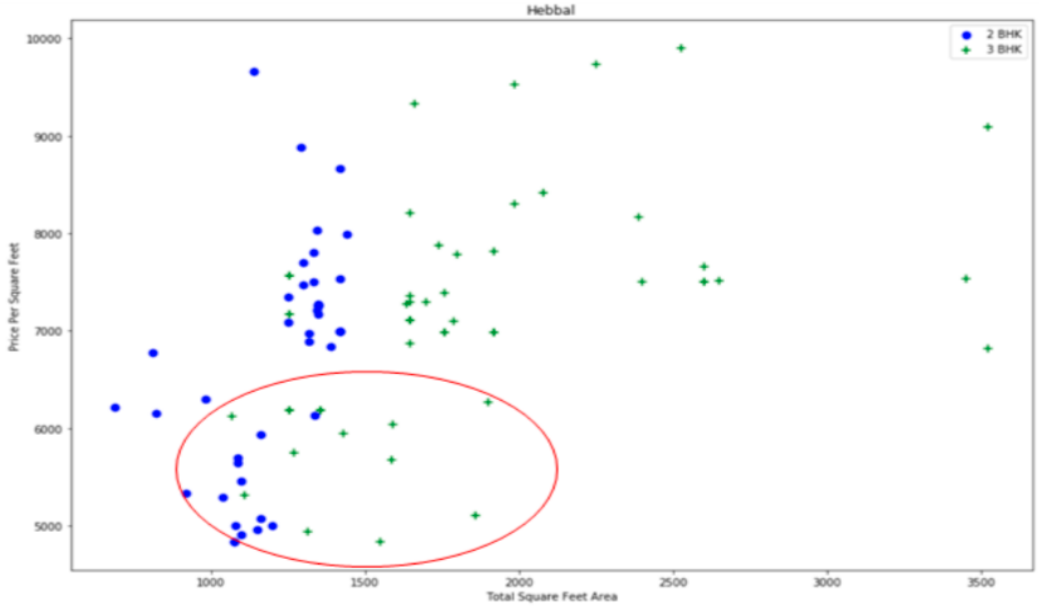
	location	size	total_sqft	bath	price	bhk	price_per_sqft
0	Electronic City Phase II	2 BHK	1056.0	2.0	39.07	2	3699.810606
1	Chikka Tirupathi	4 Bedroom	2600.0	5.0	120.00	4	4615.384615
2	Uttarahalli	3 BHK	1440.0	2.0	62.00	3	4305.555556
3	Lingadheeranahalli	3 BHK	1521.0	3.0	95.00	3	6245.890861
4	Kothanur	2 BHK	1200.0	2.0	51.00	2	4250.000000

Outlier Removal

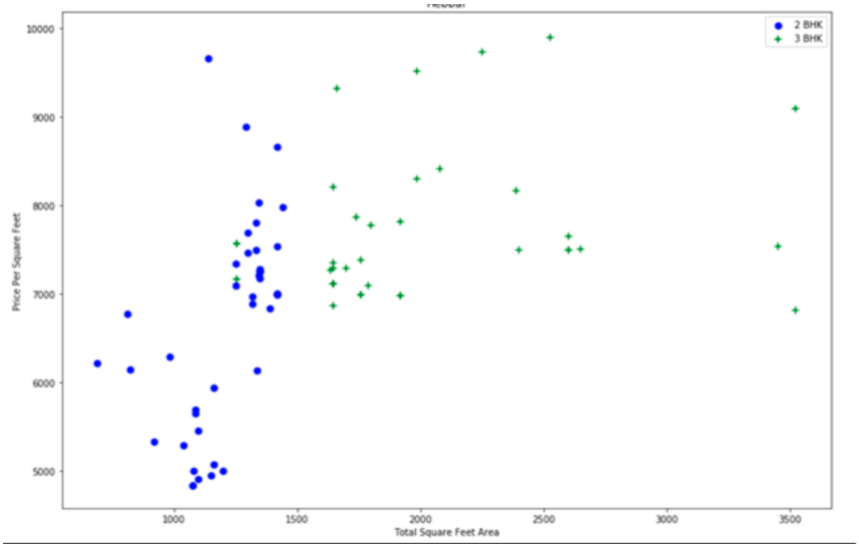
For any house with 3 bedrooms with price less than once with 2 bedrooms





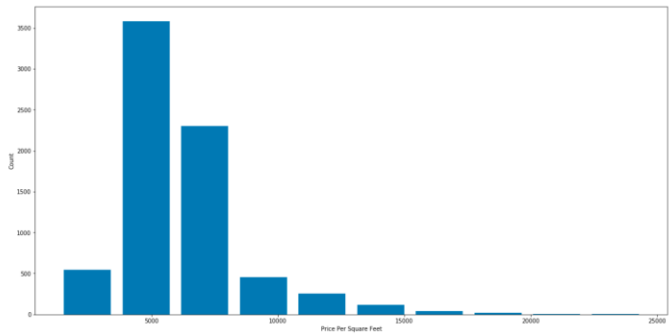


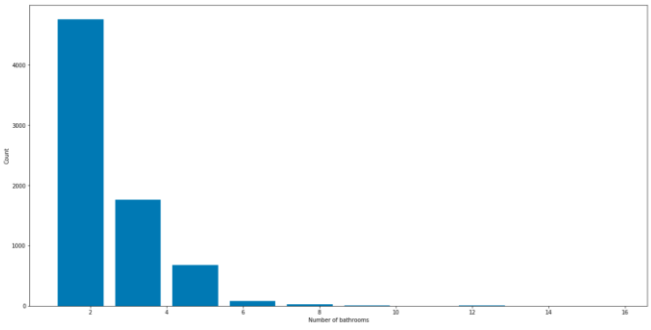
After procesing



Outlier Removal Using Bathrooms Feature

Before after


Count
Price Per Square Feet
0
500
1000
1500
2000
2500
3000
3500
5000
10000
15000
20000
25000


Count
Number of bathrooms
0
1000
2000
3000
4000
2
4
6
8
10
12
14
16