



House Price Prediction

By,
Jofia Joe
Vandana Vijayan
Gana Shankarappa
Sanjo Antony M



INTRODUCTION

- Demonstrates the usage of machine learning algorithms in the prediction of Real estate/House prices in Bangalore.
- Literature about research on machine learning prediction of house prices in India is extremely limited.
- Use machine learning algorithms to implement this prediction engine for real-life usage by users.
- Findings:
 - Different algorithms can drastically change accuracy.
 - A poor dataset can negatively affect the predictions.
 - Sufficient proof of what algorithm is best suitable for this task.

PROBLEM STATEMENT

- People and real estate agencies buy or sell houses, people buy to live in or as an investment and the agencies buy to run a business.
- Everyone should get exactly what they pay for.
- Over-valuation/Under-valuation in housing markets has always been an issue
- Lack of proper detection measures.
- Primary aim - use Machine Learning Techniques and curate them into ML models which can then serve the users.





GAPS IN EXISTING SYSTEM

- The present method is that the customer approaches a real estate agent to manage his/her investments and suggest suitable estates for his investments.
 - agents need to be paid a fraction of the amount just for searching a house and setting a price tag for you.
 - agents and sellers may have a secret dealing and the customer might be sold an overpriced house without his/her knowledge.
- When people first think of buying a house/Real estate they tend to go online and try to study trends and other related stuff.
 - doesn't have detailed knowledge & accurate information about what the actual price should be.
 - misinformation about the prices in the internet.
 - comparing it with multiple estates is highly time-consuming and has a potential risk of incorrect pricing.

GOAL

- ❑ The main aim of this project is to foresee house costs in Bengaluru city in view of certain elements like area, size/region, number of rooms, and number of washrooms.
- ❑ Bengaluru house price dataset is utilized to create the model.
- ❑ We have tried using a few machine learning algorithms in order to find out the best one which can give us the most accurate results .





TOOLS AND TECHNOLOGIES USED

- Python
- Numpy and Pandas for Data Cleaning
- Matplotlib for Data Visualization
- Sklearn for Model Building
- Jupyter Notebook as IDE



STEPS

- Data Collection
- Data Cleaning
- Feature Engineering
- Dimensionality Reduction
- Outlier Detection and Removal
- Model Building
- Model Testing


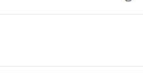

1. IMPORTING DATASET

The dataset was downloaded from here : <https://www.kaggle.com/amitabhajoy/bengaluru-house-price-data>

Bengaluru_House_Data.csv (938.02 kB)

Detail Compact Column

9 of 9 columns

area_type	availability	location	size	society	total_sqft	# bath	# balcony	price						
Super built-up Area	66%	Ready To Move	79%	Whitefield	4%	2 BHK	39%	[null]	41%	1200	6%			
Built-up Area	18%	18-Dec	2%	Sarjapur Road	3%	3 BHK	32%	GrrvaGr	1%	1100	2%			
Other (2112)	16%	Other (2432)	18%	Other (12381)	93%	Other (3811)	29%	Other (7738)	58%	Other (12256)	92%			
Super built-up Area	19-Dec			Electronic City Phase II		2 BHK		Coomee		1056		2	1	39.07
Plot Area	Ready To Move			Chikka Tirupathi		4 Bedroom		Theanmp		2600		5	3	120
Built-up Area	Ready To Move			Uttarahalli		3 BHK				1440		2	3	62
Super built-up Area	Ready To Move			Lingadheeranahalli		3 BHK		Soiewre		1521		3	1	95
Super built-up Area	Ready To Move			Kothanur		2 BHK				1200		2	1	51
Super built-up Area	Ready To Move			Whitefield		2 BHK		DuenaTa		1170		2	1	38
Super built-up Area	18-May			Old Airport Road		4 BHK		Jaades		2732		4		204
Super built-up Area	Ready To Move			Rajaji Nagar		4 BHK		Brway G		3300		4		600
Super built-up Area	Ready To Move			Marathahalli		3 BHK				1310		3	1	63.25
Plot Area	Ready To Move			Gandhi Bazar		6 Bedroom				1020		6		370
Super built-up Area	18-Feb			Whitefield		3 BHK				1800		2	2	70
Plot Area	Ready To Move			Whitefield		4 Bedroom		Prrry M		2785		5	3	295
Super built-up Area	Ready To Move			7th Phase JP Nagar		2 BHK		Shncyes		1000		2	1	38



ATTRIBUTES IN DATASET

1. Area_type
2. Availability
3. Location
4. Size
5. Society
6. Total_sqft
7. Bath
8. Balcony
9. Price

2. DATA CLEANING

a) Drop features that are not required to build our model.

- Area_type
- Society
- Balcony
- Availability

b) Handle NA values.

```
In [13]: df2 = df1.drop(['area_type', 'society', 'balcony', 'availability'], axis='columns')
df2.shape
```

```
Out[13]: (13320, 5)
```

```
In [14]: df2.isna().sum()
```

```
Out[14]: location      1
size      16
total_sqft  0
bath      73
price      0
dtype: int64
```

```
In [9]: df2.head(3)
```

```
Out[9]:
```

	location	size	total_sqft	bath	price
0	Electronic City Phase II	2 BHK	1056	2.0	39.07
1	Chikka Tirupathi	4 Bedroom	2600	5.0	120.00
2	Uttarahalli	3 BHK	1440	2.0	62.00

3. FEATURE ENGINEERING

a) Add new feature (integer) for bhk (bedroom hall kitchen)

```
In [11]: df3['size'].unique()
```

```
Out[11]: array(['2 BHK', '4 Bedroom', '3 BHK', '4 BHK', '6 Bedroom', '3 Bedroom',  
              '1 BHK', '1 RK', '1 Bedroom', '8 Bedroom', '2 Bedroom',  
              '7 Bedroom', '5 BHK', '7 BHK', '6 BHK', '5 Bedroom', '11 BHK',  
              '9 BHK', '9 Bedroom', '27 BHK', '10 Bedroom', '11 Bedroom',  
              '10 BHK', '19 BHK', '16 BHK', '43 Bedroom', '14 BHK', '8 BHK',  
              '12 Bedroom', '13 BHK', '18 Bedroom'], dtype=object)
```

```
In [13]: df3.head()
```

```
Out[13]:
```

	location	size	total_sqft	bath	price	bhk
0	Electronic City Phase II	2 BHK	1056	2.0	39.07	2
1	Chikka Tirupathi	4 Bedroom	2600	5.0	120.00	4
2	Uttarahalli	3 BHK	1440	2.0	62.00	3
3	Lingadheeranahalli	3 BHK	1521	3.0	95.00	3
4	Kothanur	2 BHK	1200	2.0	51.00	2

b) Explore and transform total_sqft feature

```
In [18]: df3[~df3.total_sqft.apply(is_float)].head(5)
```

Out[18]:

	location	size	total_sqft	bath	price	bhk
30	Yelahanka	4 BHK	2100 - 2850	4.0	186.000	4
122	Hebbal	4 BHK	3067 - 8156	4.0	477.000	4
137	8th Phase JP Nagar	2 BHK	1042 - 1105	2.0	54.005	2
165	Sarjapur	2 BHK	1145 - 1340	2.0	43.490	2
188	KR Puram	2 BHK	1015 - 1540	2.0	56.800	2

Out[78]:

	location	size	total_sqft	bath	price	bhk
0	Electronic City Phase II	2 BHK	1056.0	2.0	39.07	2
1	Chikka Tirupathi	4 Bedroom	2600.0	5.0	120.00	4
2	Uttarahalli	3 BHK	1440.0	2.0	62.00	3
3	Lingadheeranahalli	3 BHK	1521.0	3.0	95.00	3
4	Kothanur	2 BHK	1200.0	2.0	51.00	2
5	Whitefield	2 BHK	1170.0	2.0	38.00	2
6	Old Airport Road	4 BHK	2732.0	4.0	204.00	4

c) Add new feature price_per_sqft

```
In [80]: df5['price_per_sqft'] = df5['price'] * 100000 / df5['total_sqft'] # price in Lakhs  
df5.head(10)
```

Out[80]:

	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
5	Whitefield	2 BHK	1170.0	2.0	38.00	2	3247.863248
6	Old Airport Road	4 BHK	2732.0	4.0	204.00	4	7467.057101
7	Rajaji Nagar	4 BHK	3300.0	4.0	600.00	4	18181.818182
8	Marathahalli	3 BHK	1310.0	3.0	63.25	3	4828.244275
9	Gandhi Bazar	6 Bedroom	1020.0	6.0	370.00	6	36274.509804

4. DIMENSIONALITY REDUCTION

Dimensionality reduction is a machine learning (ML) or statistical technique of reducing the amount of random variables in a problem by obtaining a set of principal variables.

Changed the name of the locations having less than 10 apartments in that area into “other” category.

```
In [94]: df5.loc[9:20]
```

```
Out[94]:
```

	location	size	total_sqft	bath	price	bhk	price_per_sqft
9	other	6 Bedroom	1020.0	6.0	370.0	6	36274.509804
10	Whitefield	3 BHK	1800.0	2.0	70.0	3	3888.888889
11	Whitefield	4 Bedroom	2785.0	5.0	295.0	4	10592.459605
12	7th Phase JP Nagar	2 BHK	1000.0	2.0	38.0	2	3800.000000
13	Gottigere	2 BHK	1100.0	2.0	40.0	2	3636.363636
14	Sarjapur	3 Bedroom	2250.0	3.0	148.0	3	6577.777778
15	Mysore Road	2 BHK	1175.0	2.0	73.5	2	6255.319149
16	Bisuvanahalli	3 BHK	1180.0	3.0	48.0	3	4067.796610
17	Raja Rajeshwari Nagar	3 BHK	1540.0	3.0	60.0	3	3896.103896
18	other	3 BHK	2770.0	4.0	290.0	3	10469.314079
19	other	2 BHK	1100.0	2.0	48.0	2	4363.636364
20	Kengeri	1 BHK	600.0	1.0	15.0	1	2500.000000

5. OUTLIER REMOVAL

- ❖ An outlier is an value that lies far away from all other values in a given dataset.
- ❖ Presence of outliers can lead to inconsistencies and further errors in results obtained so it is necessary to remove outliers
 - By keeping minimum threshold per bhk to be 300 sqft.

```
In [53]: df5[(df5.total_sqft/df5.bhk)<300] # anomalies to be removed
```

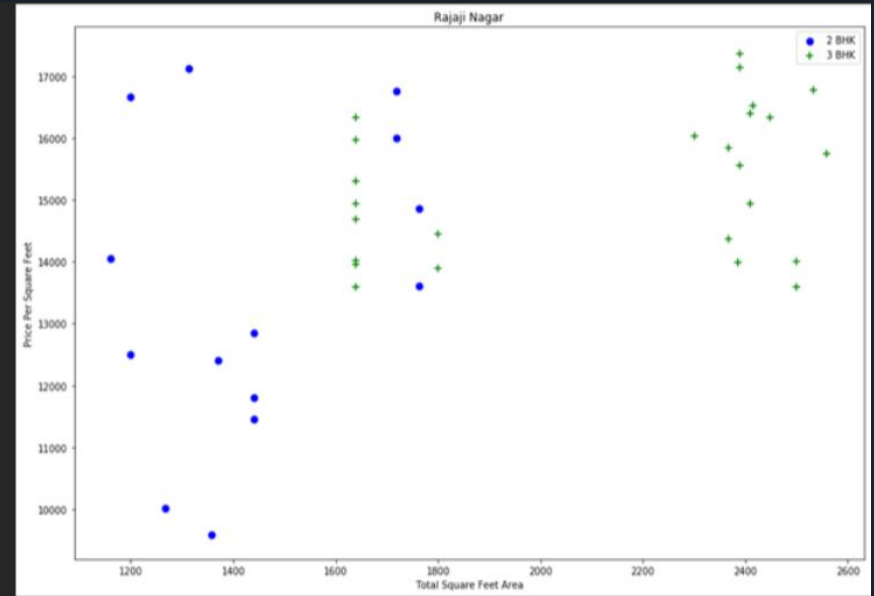
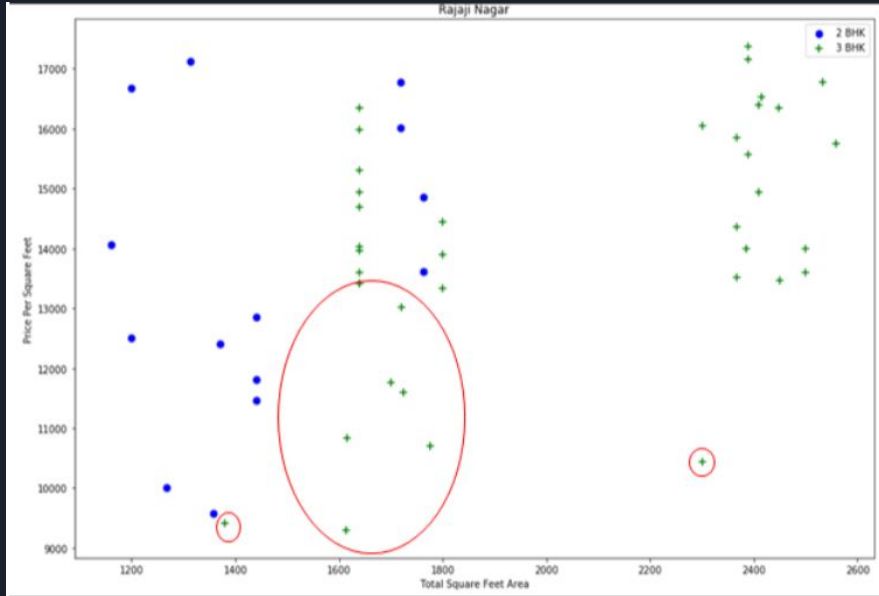
```
Out[53]:
```

	location	size	total_sqft	bath	price	bhk	price_per_sqft
9	other	6 Bedroom	1020.0	6.0	370.0	6	36274.509804
45	HSR Layout	8 Bedroom	600.0	9.0	200.0	8	33333.333333
58	Murugeshpalya	6 Bedroom	1407.0	4.0	150.0	6	10660.980810
68	Devarachikkanahalli	8 Bedroom	1350.0	7.0	85.0	8	6296.296296
70	other	3 Bedroom	500.0	3.0	100.0	3	20000.000000
...
13277	other	7 Bedroom	1400.0	7.0	218.0	7	15571.428571
13279	other	6 Bedroom	1200.0	5.0	130.0	6	10833.333333
13281	Margondanahalli	5 Bedroom	1375.0	5.0	125.0	5	9090.909091
13303	Vidyaranyapura	5 Bedroom	774.0	5.0	70.0	5	9043.927649
13311	Ramamurthy Nagar	7 Bedroom	1500.0	9.0	250.0	7	16666.666667

744 rows × 7 columns

5. OUTLIER REMOVAL

- Based on price per sqft.
 - Removed properties where for same location, the price of (for example) 3 bedroom apartment is less than 2 bedroom apartment (with same square ft area).



5. OUTLIER REMOVAL

- Using bathroom feature.
 - Here I am considering, if you have 4 bedroom home and even if you have bathroom in all 4 rooms plus one guest bathroom, you will have total bath = total bed + 1 max. Anything above that is an outlier or a data error and can be removed.

```
In [155]: df8[df8.bath>=df8.bhk+2]
```

```
Out[155]:
```

	location	size	total_sqft	bath	price	bhk	price_per_sqft
36	2nd Stage Nagarbhavi	6 Bedroom	3000.0	8.0	451.0	6	15033.333333
37	2nd Stage Nagarbhavi	6 Bedroom	2400.0	8.0	450.0	6	18750.000000
530	Arekere	4 BHK	2710.0	6.0	142.0	4	5239.852399
580	BTM 2nd Stage	3 Bedroom	1260.0	5.0	185.0	3	14682.539683
813	Bannerghatta	4 BHK	3012.0	6.0	250.0	4	8300.132802
...
9915	other	4 BHK	6652.0	6.0	510.0	4	7666.867108
10036	other	2 BHK	600.0	4.0	70.0	2	11666.666667
10089	other	3 Bedroom	5656.0	5.0	499.0	3	8822.489392
10202	other	4 BHK	6652.0	6.0	660.0	4	9921.828022
10209	other	4 Bedroom	6688.0	6.0	700.0	4	10466.507177

78 rows × 7 columns



Columns - “size” and “price_per_sqft” are dropped and the dataset is ready to be trained.

After removing all outlier dataset has 7251 rows and 5 columns.

```
In [96]: df10.head(10)
```

```
Out[96]:
```

	location	total_sqft	bath	price	bhk
0	1st Block Jayanagar	2850.0	4.0	428.0	4
1	1st Block Jayanagar	1630.0	3.0	194.0	3
2	1st Block Jayanagar	1875.0	2.0	235.0	3
3	1st Block Jayanagar	1200.0	2.0	130.0	3
4	1st Block Jayanagar	1235.0	2.0	148.0	2
5	1st Block Jayanagar	2750.0	4.0	413.0	4
6	1st Block Jayanagar	2450.0	4.0	368.0	4
8	1st Phase JP Nagar	1875.0	3.0	167.0	3
9	1st Phase JP Nagar	1500.0	5.0	85.0	5
10	1st Phase JP Nagar	2065.0	4.0	210.0	3

6. ONE HOT ENCODING FOR LOCATION

9/1:

	location	total_sqft	bath	price	bhk	1st Block Jayanagar	1st Phase JP Nagar	2nd Phase Judicial Layout	2nd Stage Nagarbhavi	5th Block Hbr Layout	...	Vijayanagar	Vishveshwarya Layout	Vishwapriya Layout	Vittasandra	Whitefiel
0	1st Block Jayanagar	2850.0	4.0	428.0	4	1	0	0	0	0	...	0	0	0	0	0
1	1st Block Jayanagar	1630.0	3.0	194.0	3	1	0	0	0	0	...	0	0	0	0	0
2	1st Block Jayanagar	1875.0	2.0	235.0	3	1	0	0	0	0	...	0	0	0	0	0
3	1st Block Jayanagar	1200.0	2.0	130.0	3	1	0	0	0	0	...	0	0	0	0	0
4	1st Block Jayanagar	1235.0	2.0	148.0	2	1	0	0	0	0	...	0	0	0	0	0
5	1st Block Jayanagar	2750.0	4.0	413.0	4	1	0	0	0	0	...	0	0	0	0	0
6	1st Block Jayanagar	2450.0	4.0	368.0	4	1	0	0	0	0	...	0	0	0	0	0
8	1st Phase JP Nagar	1875.0	3.0	167.0	3	0	1	0	0	0	...	0	0	0	0	0
9	1st Phase JP Nagar	1500.0	5.0	85.0	5	0	1	0	0	0	...	0	0	0	0	0
10	1st Phase JP Nagar	2065.0	4.0	210.0	3	0	1	0	0	0	...	0	0	0	0	0

7. MODEL BUILDING

- We used 3 machine learning algorithms to predict the prices of houses namely Linear Regression, Decision Tree and Lasso
- However Linear Regression gave the best results. Hence it is used.

```
In [68]: from sklearn.linear_model import LinearRegression
lr_model = LinearRegression()
lr_model.fit(X_train,y_train)
lr_model.score(X_test,y_test)
```

```
Out[68]: 0.8452277697874389
```

```
In [70]: from sklearn.tree import DecisionTreeRegressor
dt_model = DecisionTreeRegressor()
dt_model.fit(X_train,y_train)
dt_model.score(X_test,y_test)
```

```
Out[70]: 0.7258540996078431
```

```
In [71]: from sklearn.linear_model import Lasso
l_model = Lasso()
l_model.fit(X_train,y_train)
l_model.score(X_test,y_test)
```

```
Out[71]: 0.7237775279429011
```



8. TEST THE MODEL

The model is used to predict prices for few properties.

`predict_price(location, sqft, bath, bhk)`

```
In [181]: predict_price('1st Phase JP Nagar',1000, 2, 2)
```

```
Out[181]: 83.49904677209898
```

```
In [182]: predict_price('1st Phase JP Nagar',1000, 3, 3)
```

```
Out[182]: 86.80519395236702
```

```
In [183]: predict_price('Indira Nagar',1000, 2, 2)
```

```
Out[183]: 181.27815484006317
```

```
In [184]: predict_price('Indira Nagar',1000, 3, 3)
```

```
Out[184]: 184.5843020203312
```

```
In [185]: predict_price('Whitefield',1000,2,2)
```

```
Out[185]: 53.358388097755665
```

```
In [186]: predict_price('Whitefield',1000,3,3)
```

```
Out[186]: 56.664535278023706
```



RESULTS

Although we used different algorithms for house price prediction Linear Regression was found to be the best algorithm as linear regression was able to give the best model score of 84.5% .

Linear Regression	84.5%
Decision Tree Classifier	71.7%
Lasso	72.3%



APPLICATIONS

- Using this proposed model, we want people to buy houses and real estate at their rightful prices.
- Ensure that they don't get tricked by sketchy agents who just are after their money.
- Help Big companies by giving accurate predictions for them to set the pricing.
- Save them from a lot of hassle and save a lot of precious time and money.
- Correct real estate prices are the essence of the market and we want to ensure that by using this model.
- Likewise, house price predictions are also beneficial for property investors to know the pattern of lodging costs in a specific area.



A blue parallelogram and a light green parallelogram are positioned in the upper-left corner of the slide. The blue shape is partially behind the green one. Both shapes are oriented diagonally, with their longer sides running from the top-left towards the bottom-right.

THANK YOU