Project Report

Industrial Training
5th Semester B.E. Information Technology

Housing Price Prediction Model using Linear Regression

Submitted By-

Manik Modi
IT Section 1
UE188056

TABLE OF CONTENTS

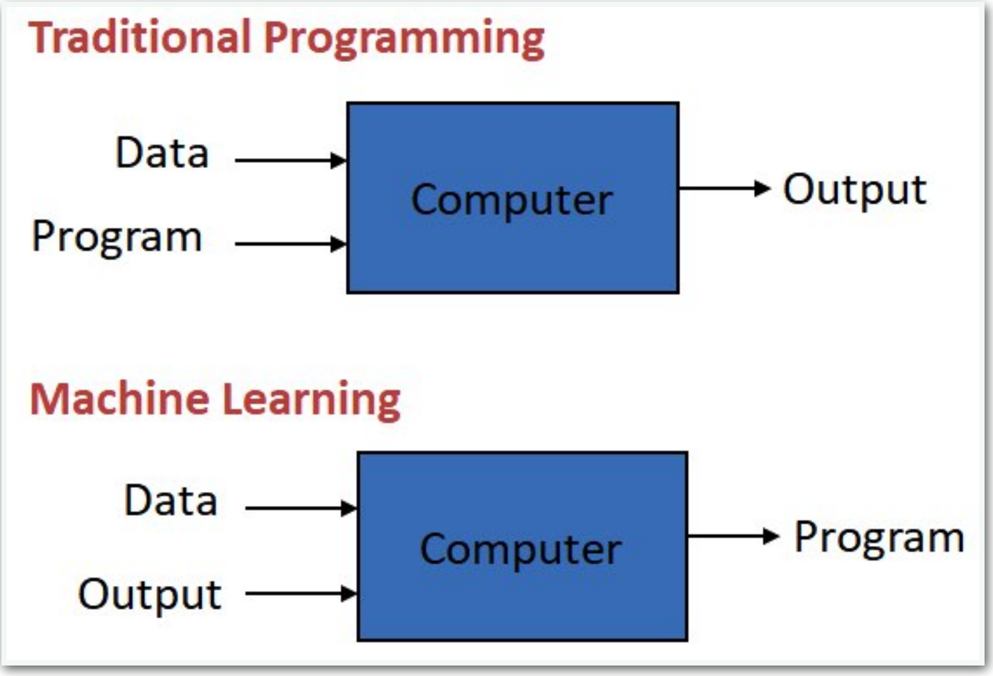
Machine Learning

Definition

"Field of study that gives computers the ability to learn without being explicitly programmed"

"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."

The complexity in traditional computer programming is in the code (programs that people write).
In **machine learning**, algorithms (programs) are in principle simple and the complexity (structure) is in the data.



That is, machine learning is the about the construction and study of systems that can learn from data.

This is very different than traditional computer programming.

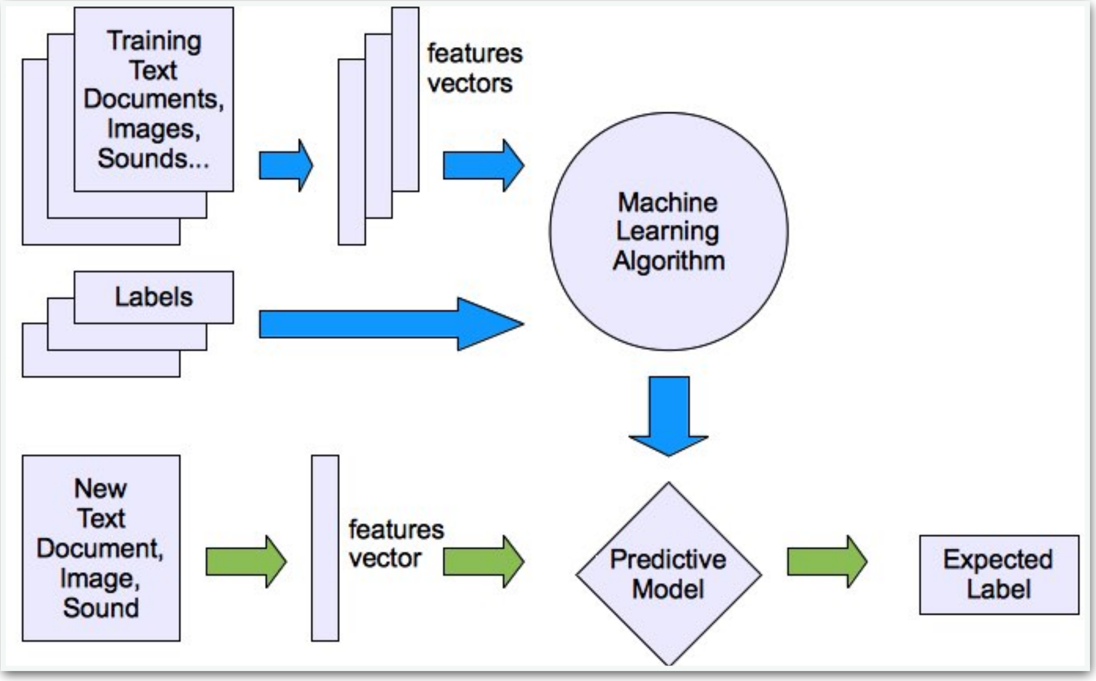
Types of Learning Algorithms

Two types of learning algorithms are commonly used:

- **Supervised learning**
  Teach the computer how to do something, then let it use it's new found knowledge to do it.

- **Unsupervised learning**
  Let the computer learn how to do something, and use this to determine structure and patterns in data.

Supervised Learning


Training Text Documents, Images, Sounds...
features vectors
Machine Learning Algorithm
Labels
New Text Document, Image, Sound
features vector
Predictive Model
Expected Label

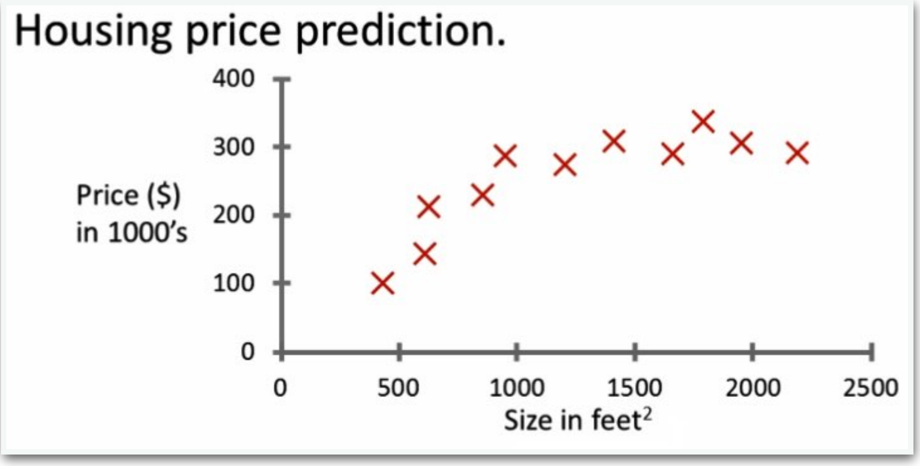
In **Supervised Learning**, we are given a data set and already know what our correct output should look like, having the idea that there is a relationship between the input and the output.

Supervised learning problems are categorised into "*regression*" and "*classification*" problems.

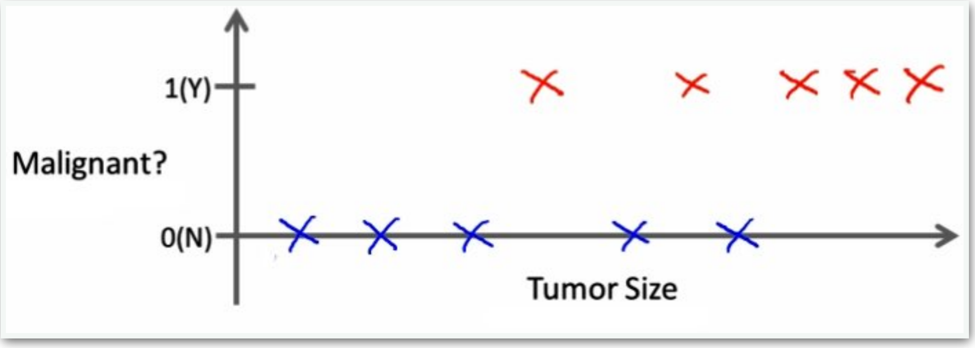
- In a **Regression** problem, we are trying to predict results within a continuous output, meaning that we are trying to map input variables to some continuous function.

- In a **Classification** problem, we are instead trying to predict results in a discrete output. In other words, we are trying to map input variables into discrete categories.

Example 1:
Given data about the size of houses on the real estate market, try to predict their price. Price as a function of size is a continuous output, so this is a **regression problem**.



Example 2:
Given a patient with a tumor, we have to predict whether the tumor is malignant or benign. Since output can be either or and not both this is a **classification problem**.



Introduction to Project

Machine learning is a branch of Artificial Intelligence which is used to analyse the data more smartly. It automates the process using certain algorithms to minimise human intervention in the process.

Linear regression is one such machine learning tool that helps you to make predictions by using the existing data (basically the relationship between the target data and set of other data).

House price forecasting is an important topic of real estate.

In this project, we will develop and evaluate the performance and the predictive power of a model trained and tested on data collected from India (Bangalore).

Linear Regression is applied to analyse historical property transactions. In our case, the house price basically depends on the parameters such as the number of bedrooms, location, size of living area etc.

Once we get a good fit, we will use this model to *predict the monetary value of a house* located at the Bangalore area.

A model like this would be very helpful for a real state agent and sellers and buyers so as to have an overview of the real estate market.

Aim and Importance

In India for so many years housing and rental prices have continued to rise. Since the housing crisis of 2008, housing prices have recovered remarkably well, especially in major housing markets.

Now with the lingering impact of demonetisation, the enforcement of the Real Estate (Regulation and Development) Act (RERA), and the lack of trust in property developers in the city, housing units sold across India in 2017 dropped by 7 percent.

Buying a home, especially in a city like Bengaluru, is a tricky choice. With its millennial crowd, vibrant culture, great climate and a slew of job opportunities, it is difficult to ascertain the price of a house.

Thus, a model like this would make people aware of the periodic fluctuations in the real estate market thereby maintaining *transparency*.

If customer finds the price of a house at some given website higher than the price predicted by the model, that property can be rejected.

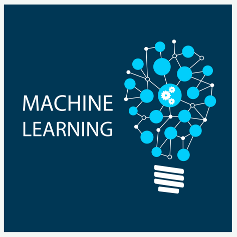
Therefore, comparisons can be made easily using this model enabling customers to make a more *informed decision*.

Hence, our aim with this project would be to :

- Create an effective price **prediction model.**
- Identify the important home price **attributes** which feed the model's predictive power.
- Validate the model's prediction **accuracy.**

Tools and Technologies Used

- **Machine learning :** Machine learning is a method of data analysis that automates analytical model building. It is a branch of artificial intelligence based on the idea that systems can learn from data, identify patterns and make decisions with minimal human intervention.



- **Python :** Python is an interpreted, high-level and general-purpose programming language which emphasises code readability with its notable use of significant whitespace.
  Libraries used in this project include:

  - **Pandas** : A fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language.

  - **NumPy** : Library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

  - **Matplotlib** : Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python.

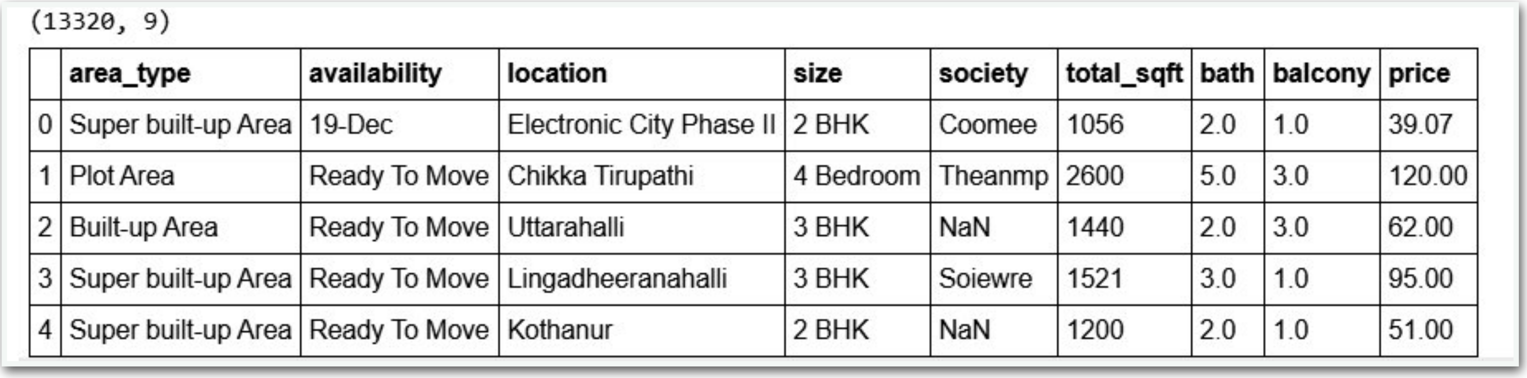
  - **Scikit Learn** : It is a machine learning library for the Python programming language. It features various classification, regression and clustering algorithms.

DATASET

The dataset used in this project comes from a Machine Learning Repository.

This data was collected in 2018 and each of the 13320 entries represents aggregate information about 9 features of homes from various properties located in Bangalore, India.

Our Data contains Bangalore houses only.
Dataset looks as follows-

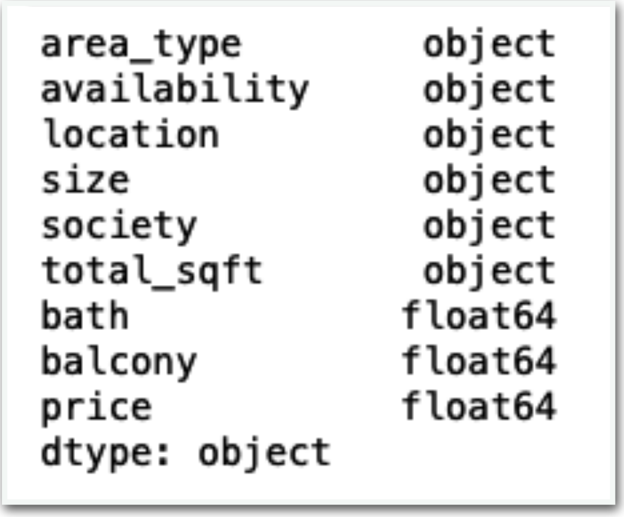


```
(13320, 9)
```

	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

The features extracted for each house are the following :

- **area_type**
- **availability**
- **location**
- **size**
- **society**
- **total_sqft**
- **bath**
- **balcony**
- **price**



```
area_type        object
availability     object
location         object
size             object
society          object
total_sqft       object
bath            float64
balcony         float64
price           float64
dtype: object
```

Project Implementation

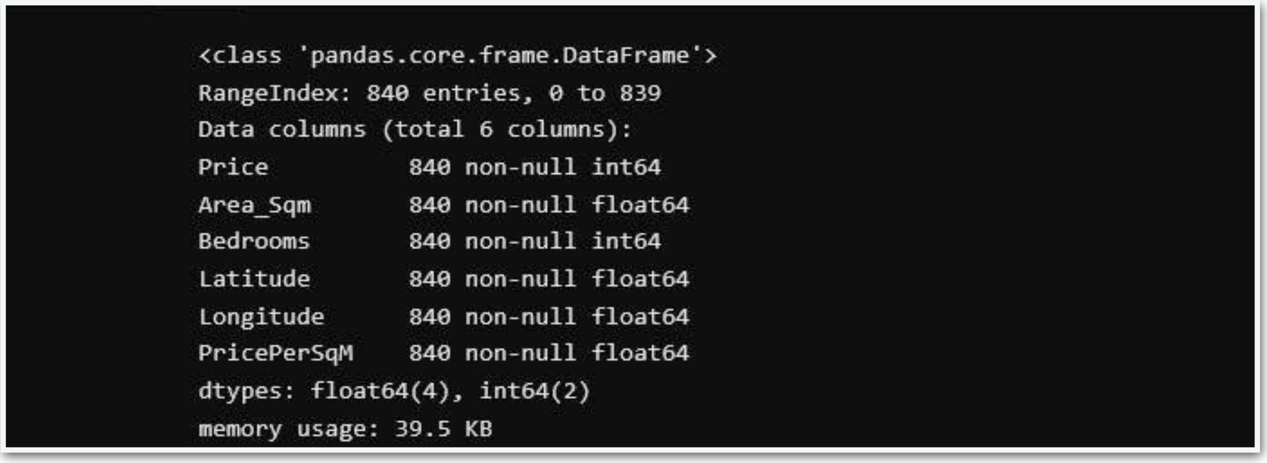
Implementation of the project model can be categorised into several important sub-processes namely:

- Data Preprocessing
- Data Cleaning
- Feature Engineering
- Dimensionality Reduction
- Outlier Removal Using Business Logic
- Model Building
- GUI Application

Data Preprocessing

Data Exploration

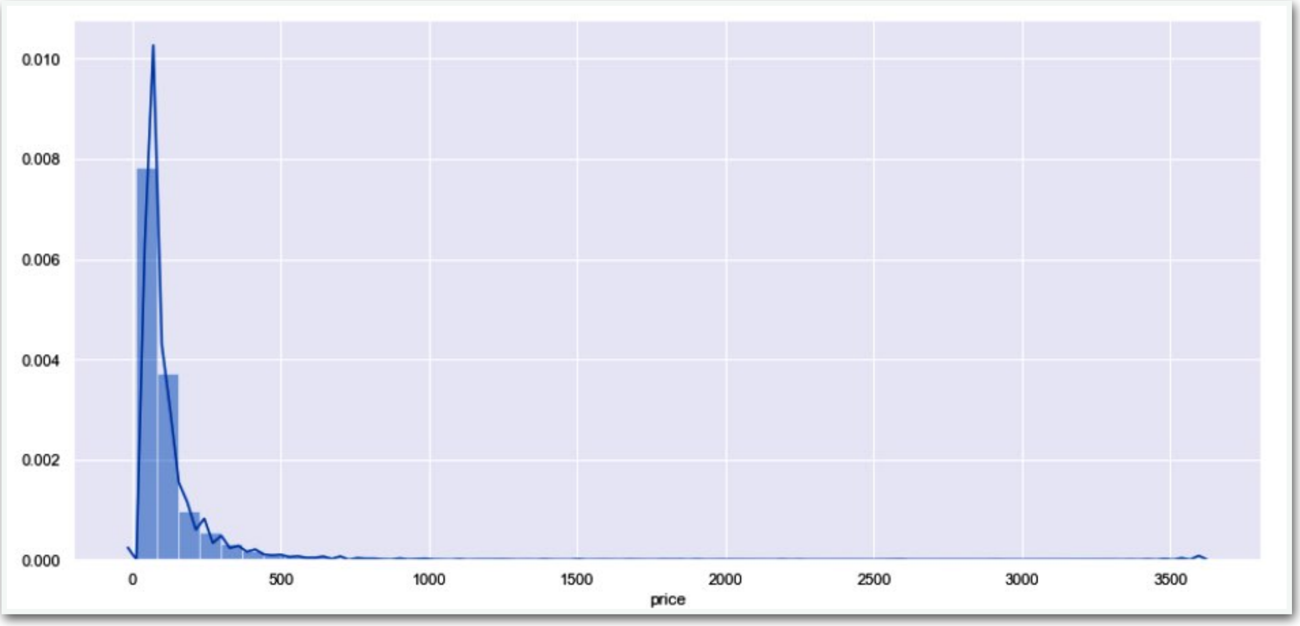
Data exploration is the first step in data analysis and typically involves summarising the main characteristics of a data set, including its size, accuracy, initial patterns in the data and other attributes. Before it can conduct analysis on data collected by multiple data sources and stored in data warehouses, an organization must know how many cases are in a data set, what variables are included, how many missing values there are and what general hypotheses the data is likely to support.
We divided the data 8:2 for Training and Testing purpose respectively.



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 840 entries, 0 to 839
Data columns (total 6 columns):
Price          840 non-null int64
Area_Sqm       840 non-null float64
Bedrooms       840 non-null int64
Latitude       840 non-null float64
Longitude      840 non-null float64
PricePerSqM    840 non-null float64
dtypes: float64(4), int64(2)
memory usage: 39.5 KB
```

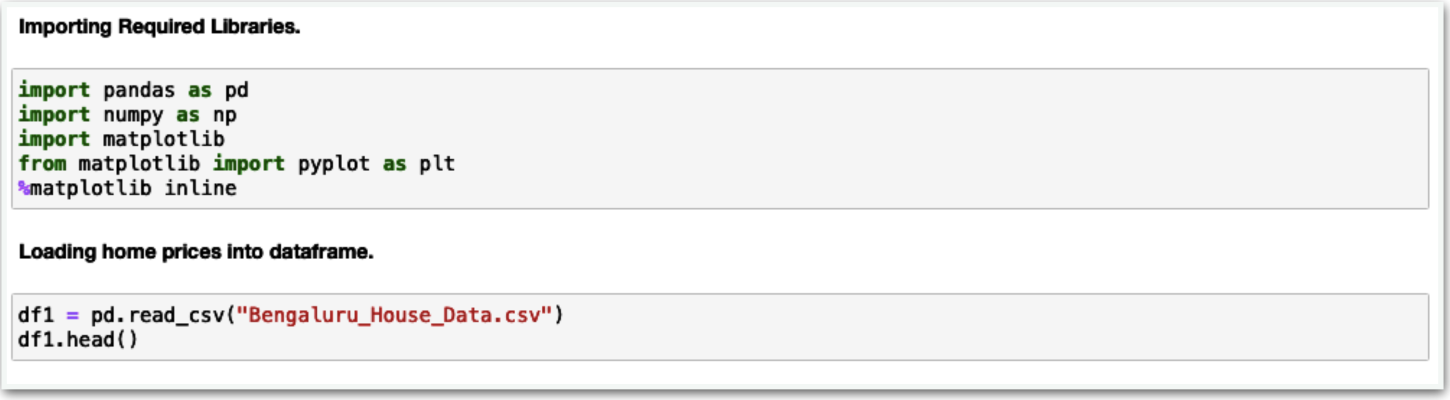
Data Visualisation

Data visualisation is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualisation tools provide an accessible way to see and understand trends, outliers, and patterns in data. In the world of Big Data, data visualisation tools and technologies are essential to analyse massive amounts of information and make data-driven decisions.



Loading Data and Importing Libraries

Import the dependencies and libraries to perform computations on the data. Also dataset (.csv) file is to be loaded.



Importing Required Libraries.

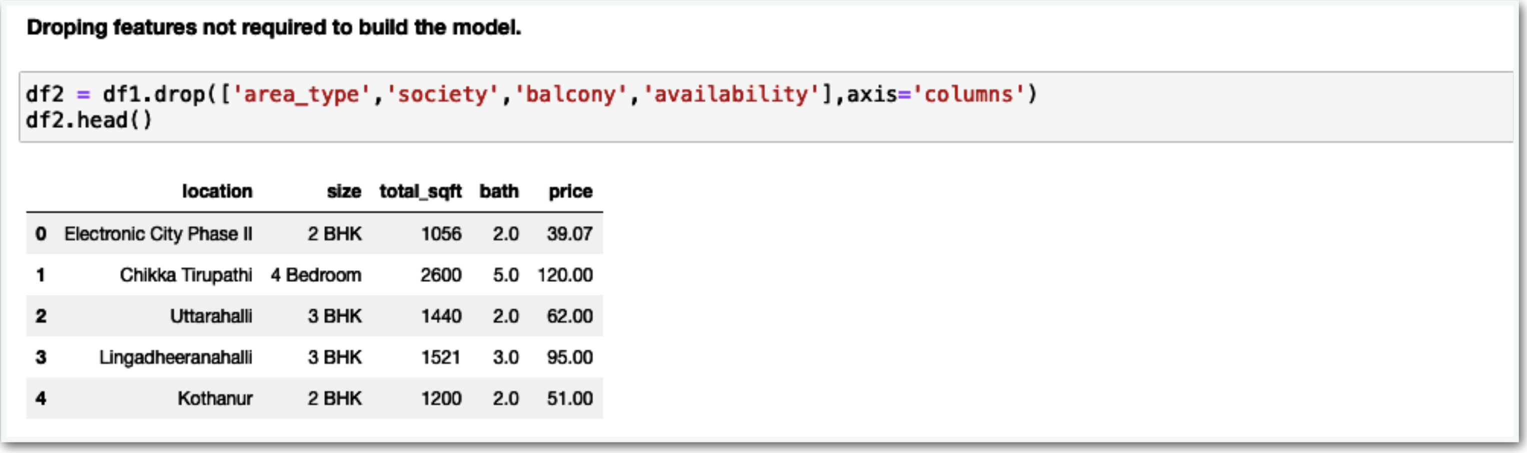
```
import pandas as pd
import numpy as np
import matplotlib
from matplotlib import pyplot as plt
%matplotlib inline
```

Loading home prices into dataframe.

```
df1 = pd.read_csv("Bengaluru_House_Data.csv")
df1.head()
```

Data Cleaning

This involves dropping features not seen to be necessary for our price prediction model since their impact on determining the final price is not much.
Dropping these features will have little to no effect on our model.



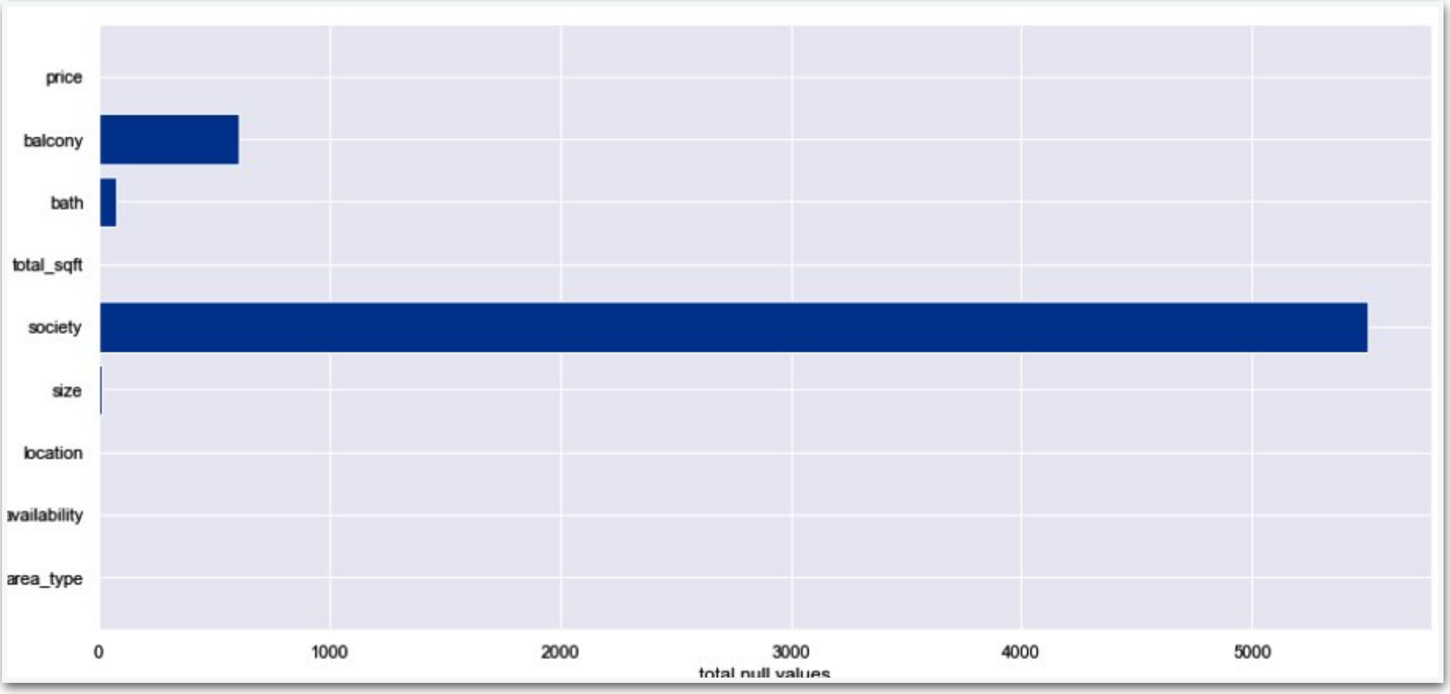
Droping features not required to build the model.

```
df2 = df1.drop(['area_type','society','balcony','availability'],axis='columns')
df2.head()
```

	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	Lingadheeranahalli	3 BHK	1521	3.0	95.00
4	Kothanur	2 BHK	1200	2.0	51.00

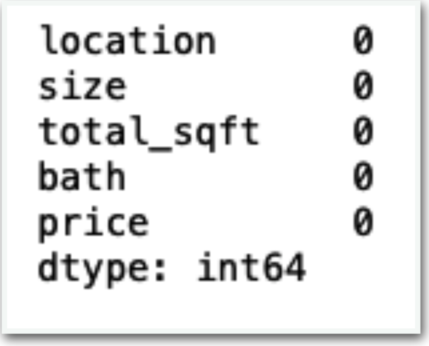
When working on analysing data, you’ll likely come across data that is missing (also called null values or NaNs).

Data cleaning is an important part of data analysis pipeline and making sure that it’s all tidy up will make the analysis much stronger.



So we just remove any rows or columns that contain missing values. Pandas does have a handy function, dropna() to help you do this.

```
df2.isnull().sum()



df3 = df2.dropna()
df3.isnull().sum()
```



```
location      0
size          0
total_sqft    0
bath          0
price         0
dtype: int64
```

Feature Engineering

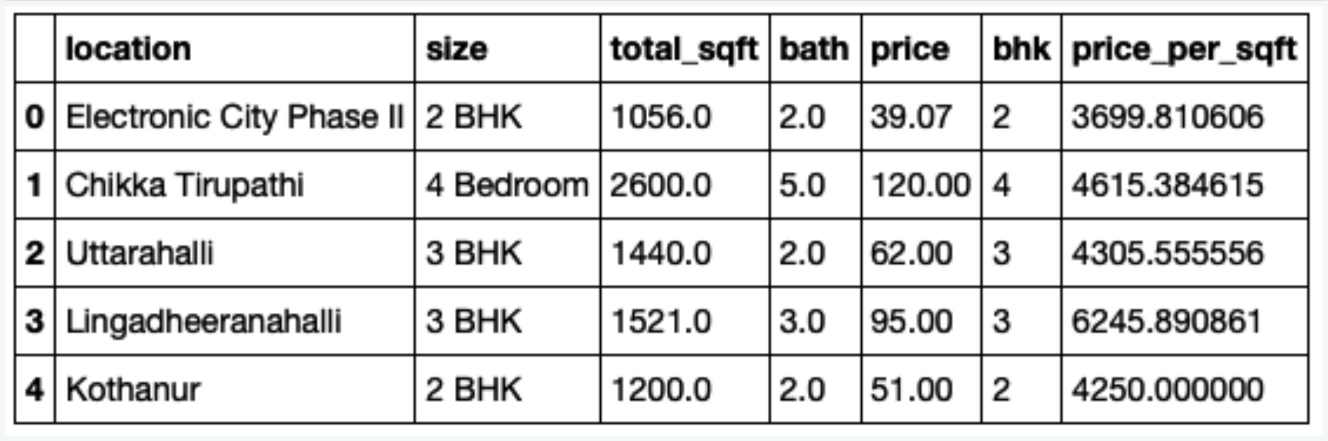
Feature engineering is the process of using domain knowledge to extract features from raw data via data mining techniques.
These features can be used to improve the performance of machine learning algorithms. Feature engineering can be considered as applied machine learning itself.

Importance of feature engineering is:

- Better features means flexibility.
- Better features means simpler models.
- Better features means better results.

Since our machine learning model can only make use of numerical data, we engineer new data columns having numeric values corresponding to columns containing character data.

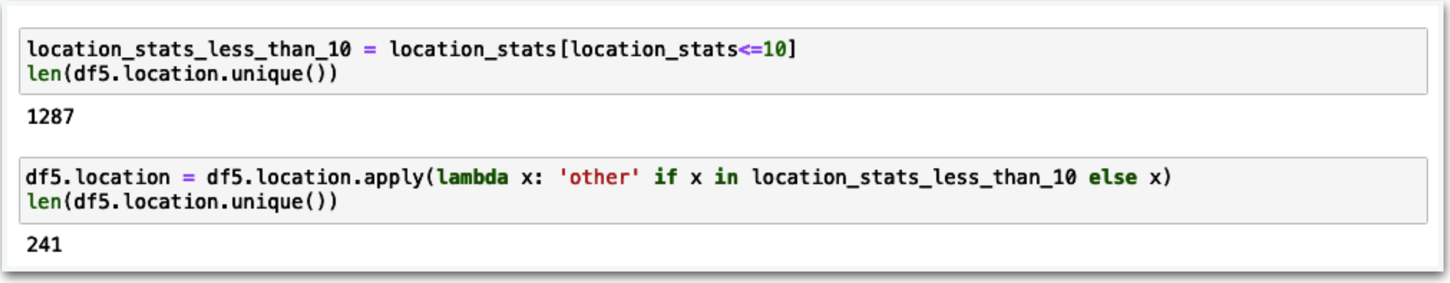
Adding a new feature bhk in place of size column and adding a feature price per square feet.



	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

Dimensionality Reduction

Dimensionality reduction, or dimension reduction, is the transformation of data from a high-dimensional space into a low-dimensional space so that the low-dimensional representation retains some meaningful properties of the original data, ideally close to its intrinsic dimension.



```
location_stats_less_than_10 = location_stats[location_stats<=10]
len(df5.location.unique())
```

```
1287
```

```
df5.location = df5.location.apply(lambda x: 'other' if x in location_stats_less_than_10 else x)
len(df5.location.unique())
```

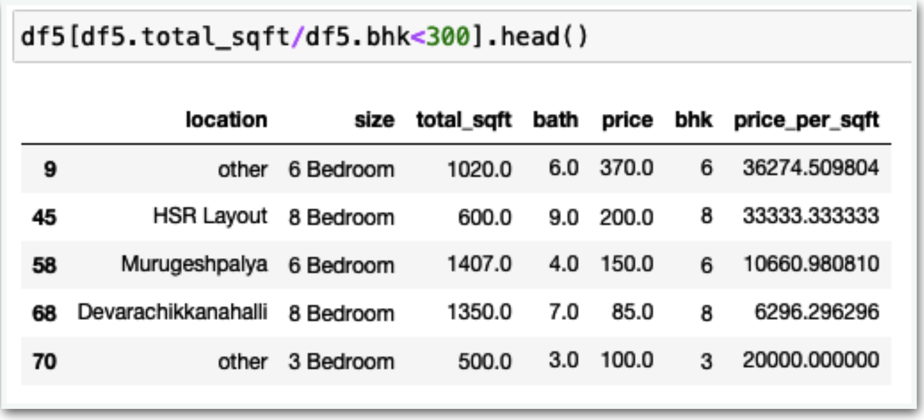
```
241
```

Any location having less than 10 data points are tagged as "other" location. This way number of categories can be reduced by huge amount.

Outlier Removal Using Business Logic

Outliers are unusual values in the dataset that distort statistical analysis and violate their assumptions. Given the problems they can cause, you might think that it’s best to remove them from your data.

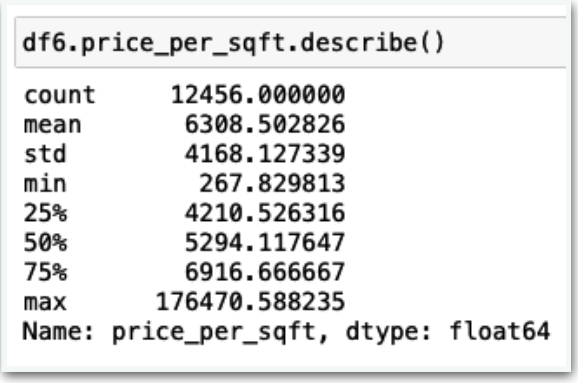
Normally **square ft per bedroom** is 300 (i.e. 2 bhk apartment is minimum 600 sqft. If you have for example 400 sqft apartment with 2 bhk then that can be removed as an outlier.



```
df5[df5.total_sqft/df5.bhk<300].head()
```

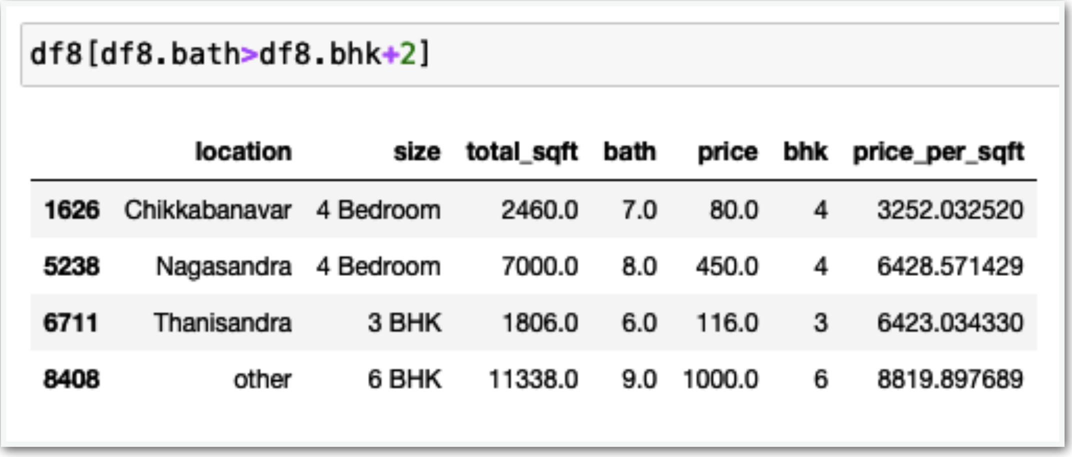
	location	size	total_sqft	bath	price	bhk	price_per_sqft
9	other	6 Bedroom	1020.0	6.0	370.0	6	36274.509804
45	HSR Layout	8 Bedroom	600.0	9.0	200.0	8	33333.333333
58	Murugeshpalya	6 Bedroom	1407.0	4.0	150.0	6	10660.980810
68	Devarachikkanahalli	8 Bedroom	1350.0	7.0	85.0	8	6296.296296
70	other	3 Bedroom	500.0	3.0	100.0	3	20000.000000

Here, min **price per sqft** is 267 whereas max is 176470, showing a wide variation in property prices. These outliers are removed per location using mean and one standard deviation.



```
df6.price_per_sqft.describe()
```

```
count     12456.000000
mean       6308.502826
std        4168.127339
min         267.829813
25%        4210.526316
50%        5294.117647
75%        6916.666667
max      176470.588235
Name: price_per_sqft, dtype: float64
```



```
df8[df8.bath>df8.bhk+2]
```

	location	size	total_sqft	bath	price	bhk	price_per_sqft
1626	Chikkabanavar	4 Bedroom	2460.0	7.0	80.0	4	3252.032520
5238	Nagasandra	4 Bedroom	7000.0	8.0	450.0	4	6428.571429
6711	Thanisandra	3 BHK	1806.0	6.0	116.0	3	6423.034330
8408	other	6 BHK	11338.0	9.0	1000.0	6	8819.897689

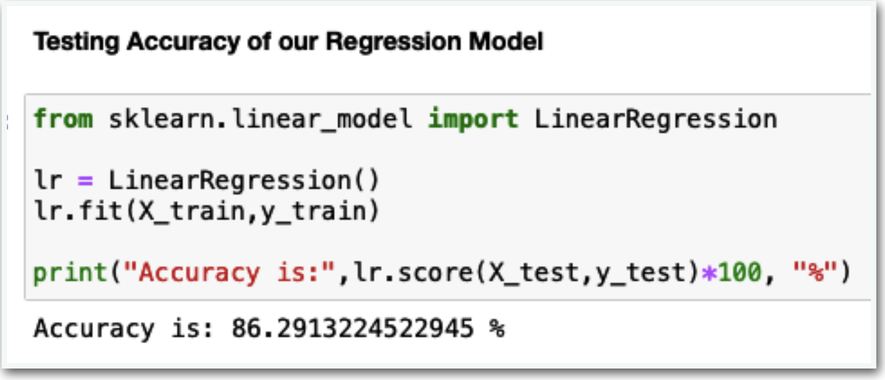
Even for a 4 bhk, it will at-most have **bathroom** in all 4 rooms plus one guest bathroom. Anything above that is an outlier which is removed.

Model Building

Using Scikit Learn machine learning library for the Python programming language we initialize the **linear regression** model.

Furthermore the dataset is split into 80% as training and 20% as testing data.

Finally the model is trained with our training data.



Testing Accuracy of our Regression Model

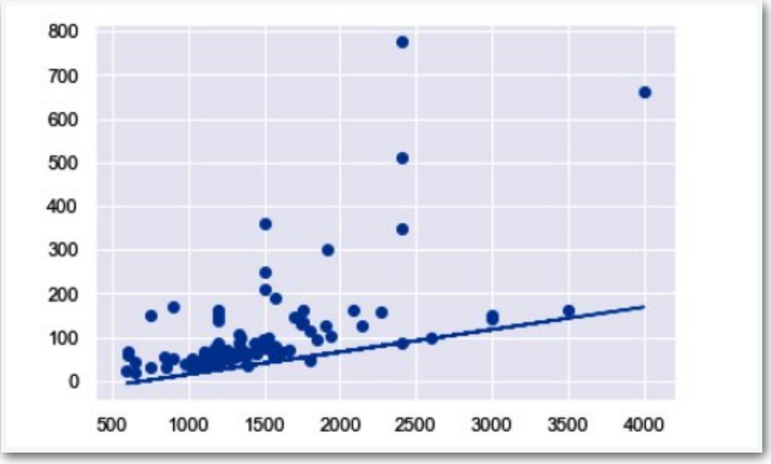
```
from sklearn.linear_model import LinearRegression

lr = LinearRegression()
lr.fit(X_train,y_train)

print("Accuracy is:",lr.score(X_test,y_test)*100, "%")
```

```
Accuracy is: 86.2913224522945 %
```

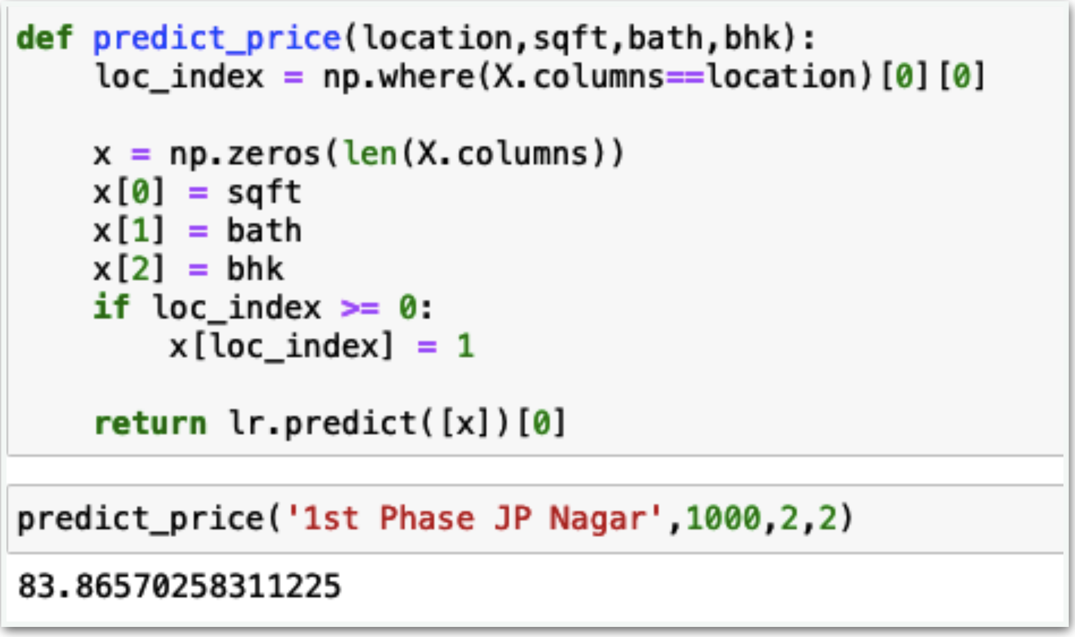
Upon testing our model returns a prediction accuracy of 86% which will work well enough.

- **Linear Regression** is a machine learning algorithm based on supervised learning.

- It performs a regression task. Regression models a target prediction value based on independent variables.



The final price prediction function takes location, area in square feet, number of bedrooms (bhk) and number of bathrooms as inputs and returns the predicted price in lakhs (rupees).



```
def predict_price(location,sqft,bath,bhk):
    loc_index = np.where(X.columns==location)[0][0]

    x = np.zeros(len(X.columns))
    x[0] = sqft
    x[1] = bath
    x[2] = bhk
    if loc_index >= 0:
        x[loc_index] = 1

    return lr.predict([x])[0]
```

```
predict_price('1st Phase JP Nagar',1000,2,2)
```

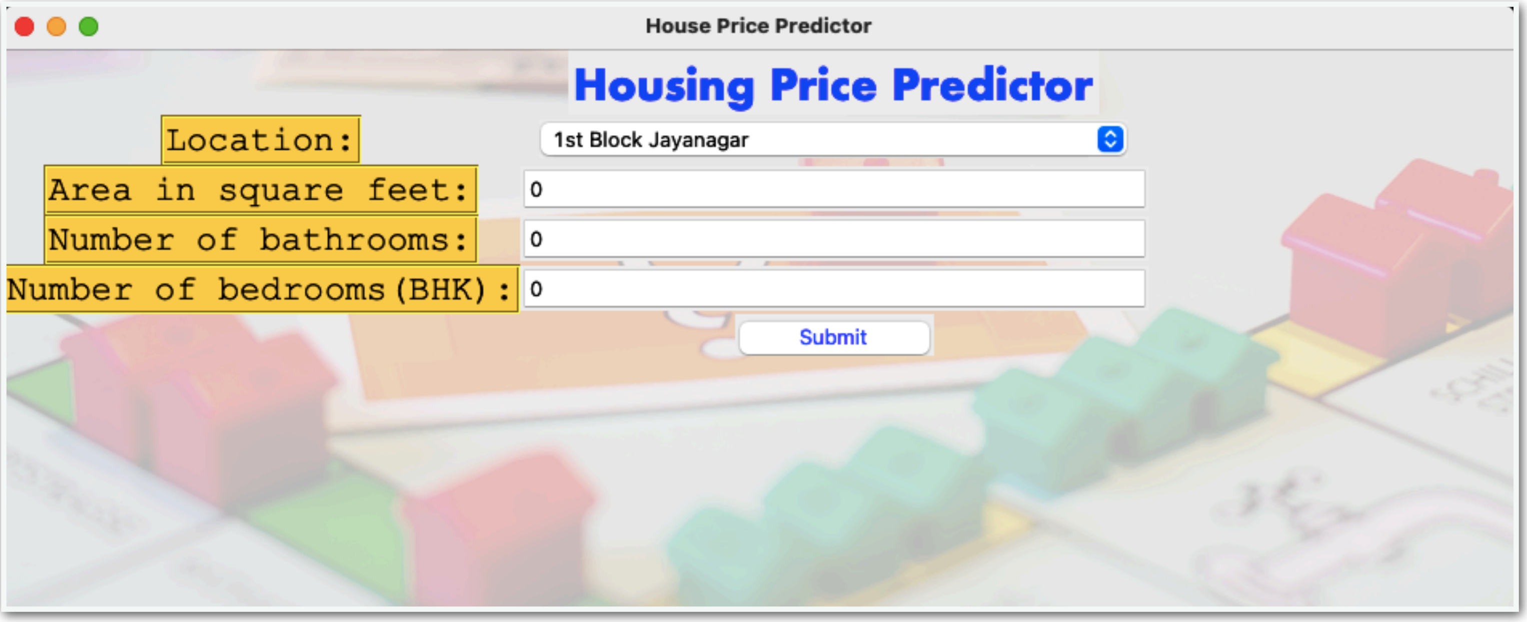
```
83.86570258311225
```

GUI Application

Here the linear regression model is deployed to a **Graphical User Interface (GUI)** in order to make it more appealing and presentable.

The GUI application is made in python itself using the Tkinter library.

Tkinter is a Python binding to the Tk GUI toolkit. It is the standard Python interface to the Tk GUI toolkit, and is Python's de facto standard GUI.



House Price Predictor

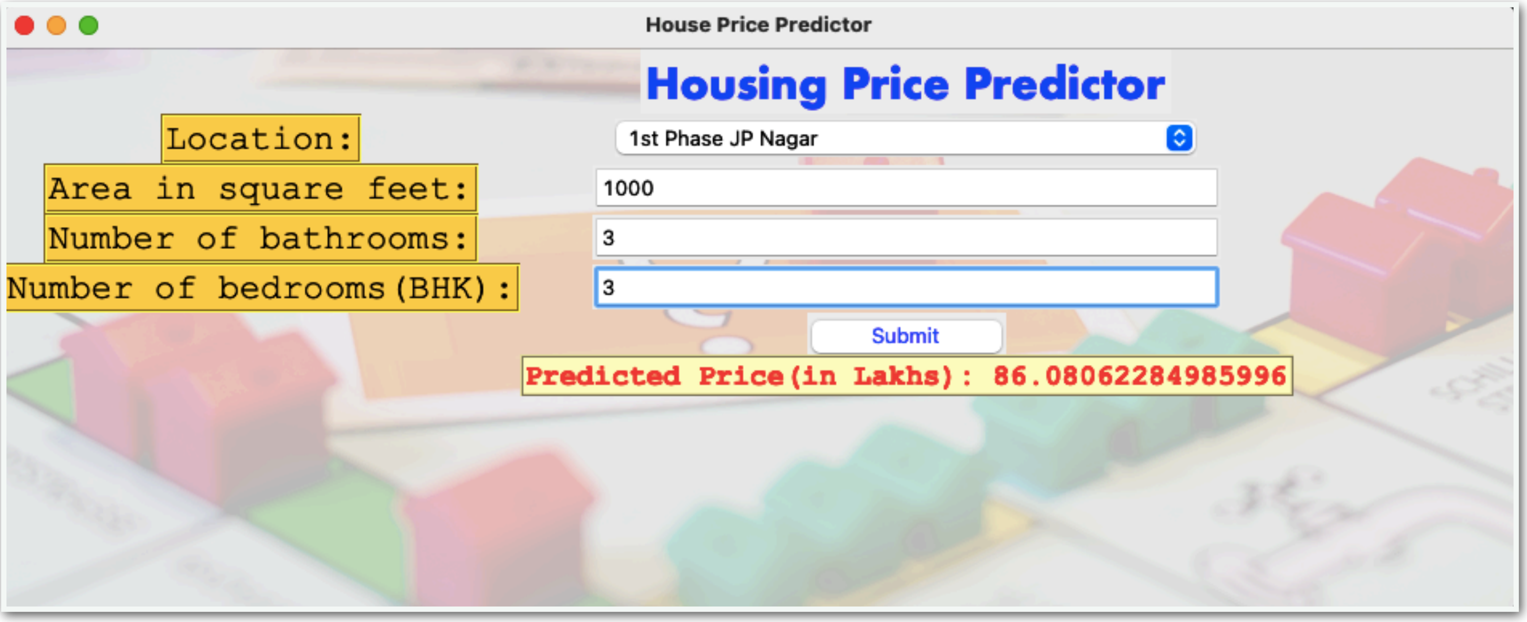
Housing Price Predictor

Location: 1st Block Jayanagar

Area in square feet: 0

Number of bathrooms: 0

Number of bedrooms(BHK): 0



House Price Predictor

Housing Price Predictor

Location: 1st Phase JP Nagar

Area in square feet: 1000

Number of bathrooms: 3

Number of bedrooms(BHK): 3

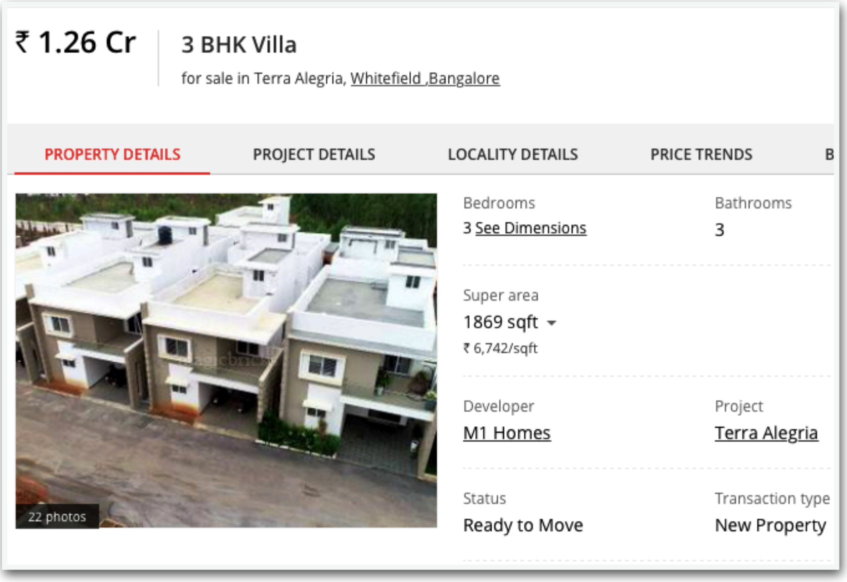
Predicted Price(in Lakhs): 86.08062284985996

Result

So, our Aim is achieved as we have successfully ticked all our parameters that we sought to complete.

It is seen that **Location** is the most effective attribute in predicting the house price and that the **Linear Regression** is the most effective model for our Dataset with accuracy score of 86%.

Comparing the price of a property found on a real estate website (*magic bricks*) to the price predicted by our Linear Regression Model.



₹ 1.26 Cr

3 BHK Villa

for sale in Terra Alegria, Whitefield ,Bangalore

Bedrooms: 3 See Dimensions

Bathrooms: 3

Super area: 1869 sqft

₹ 6,742/sqft

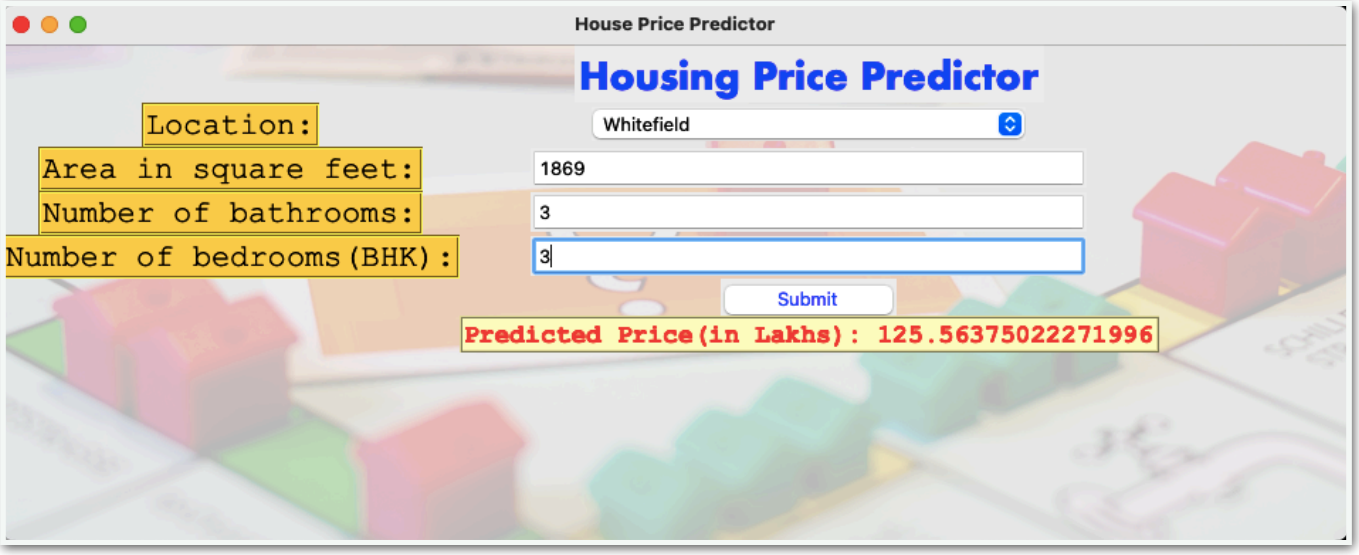
Developer: M1 Homes

Project: Terra Alegria

Status: Ready to Move

Transaction type: New Property

Actual Price - 1.26 crore
Predicted Price - 1.25 crore



House Price Predictor

Housing Price Predictor

Location: Whitefield

Area in square feet: 1869

Number of bathrooms: 3

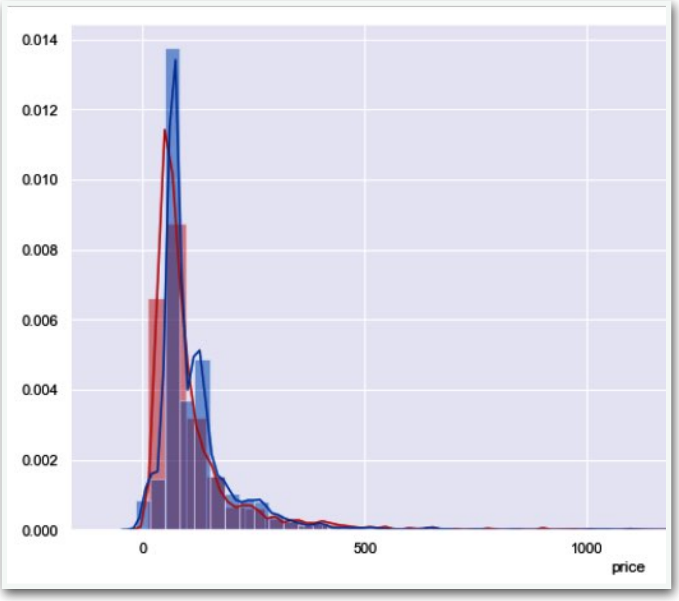
Number of bedrooms(BHK): 3

Predicted Price(in Lakhs): 125.56375022271996

Conclusion and Future Scope

In today's real estate world, it has become tough to store such huge data and extract them for one's own requirement.
Also, the extraction, analysis and cleaning of data has to be done is such a way such that it proves to be useful.

This system makes optimal use of the **Linear Regression Algorithm**. It makes use of such data in the most efficient way.



RED — TRUE VALUES
BLUE — PREDICTED VALUE

This model could help to fulfil customers by increasing the accuracy of estate choice and reducing the risk of investing in an estate.

A lot of features can be added to make the system more widely acceptable such as:

- One of the major future scopes is adding estate database of more cities which will provide the user to explore more estates and reach an accurate decision.
- More factors like recession that affect the house prices could be added.
- In-depth details of every property could be added to provide ample details of a desired estate.

This will help the system to run on a larger level and thus scaling it for widespread use.