



**Predicting house price using  
Machine learning**

# **INTRODUCTION**

## **AIM and IMPORTANCE**

### **Aim**

**These are the Parameters on which we will evaluate ourselves-**

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

## **Need and Motivation**

Having lived in India for so many years if there is one thing that I had been taking for granted, it's that housing and rental prices continue to rise. Since the housing crisis of 2008, housing prices have recovered remarkably well, especially in major housing markets. However, in the 4th quarter of 2016, I was surprised to read that Bombay housing prices had fallen the most in the last 4 years. In fact, median resale prices for condos and coops fell 6.3%, marking the first time there was a decline since Q1 of 2017. The decline has been partly attributed to political uncertainty domestically and abroad and the 2014 election. So, to maintain the transparency among customers and also the comparison can be made easy through this model. If customer finds the price of house at some given website higher than the price predicted by the model, so he can reject that house.

# DATASET

Here we have web scrapped the Data from 99acres.com website which is one of the leading real estate websites operating in INDIA.

Our Data contains Bombay Houses only.

**Dataset looks as follows-**

	Price	PricePerSqft	Area_Sqm	Location	Bedrooms	Latitude	Longitude	PricePerSqM
0	13300000	16625	74.32	Kandivali (East)	2	19.210200	72.864891	178885.00
1	9000000	15666	55.74	Ramgad Nagar	1	19.167700	72.949300	168566.16
2	9000000	19148	43.66	Mahakali Caves	1	19.130609	72.873816	206032.48
3	9000000	10588	78.97	Louis Wadi	2	19.126005	72.825052	113926.88
4	100000000	20000	464.51	Barrister Nath Pai Nagar	5	19.075014	72.907571	215200.00

```
<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 Exploration**

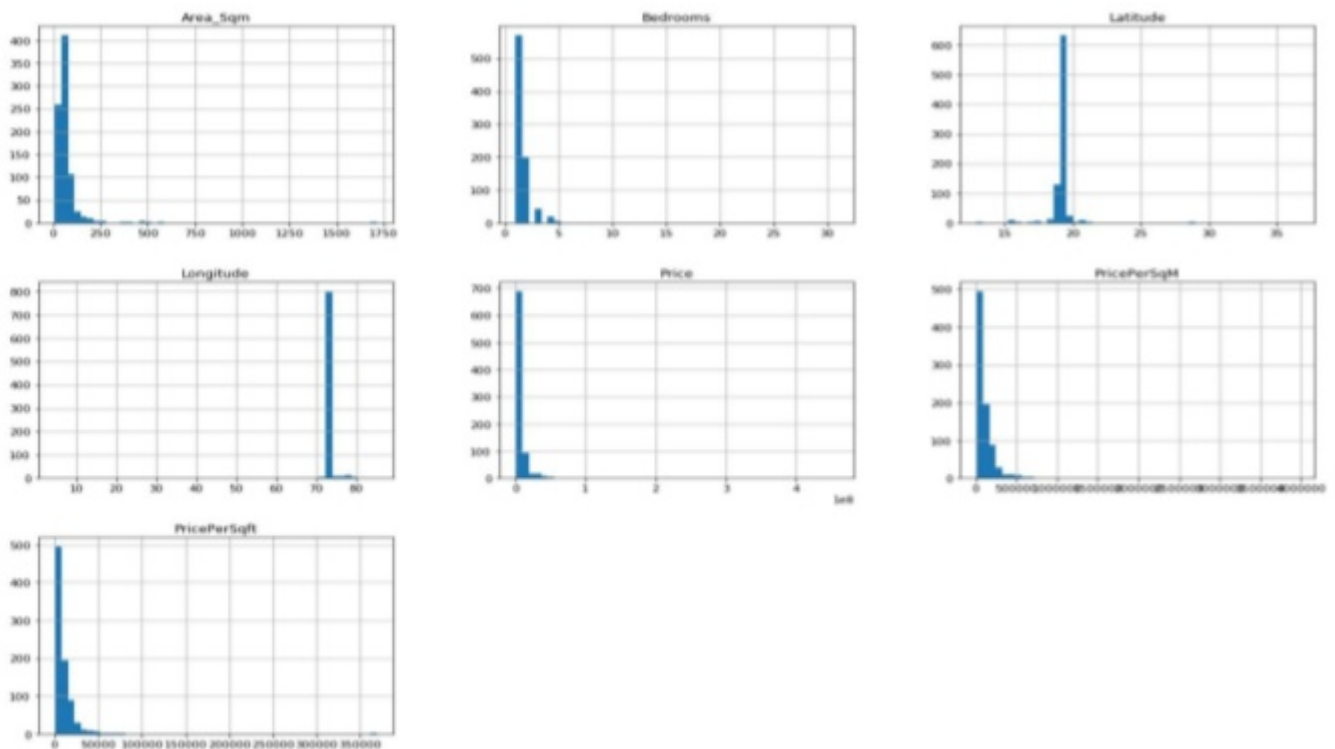
Data exploration is the first step in data analysis and typically involves summarizing the main characteristics of a data set, including its size, accuracy, initial patterns in the data and other attributes. It is commonly conducted by data analysts using visual analytics tools, but it can also be done in more advanced statistical software, Python. 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. An initial exploration of the data set can help answer these questions by familiarizing analysts with the data with which they are working.

We divided the data 9:1 for Training and Testing purpose respectively.



# Data Visualization

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



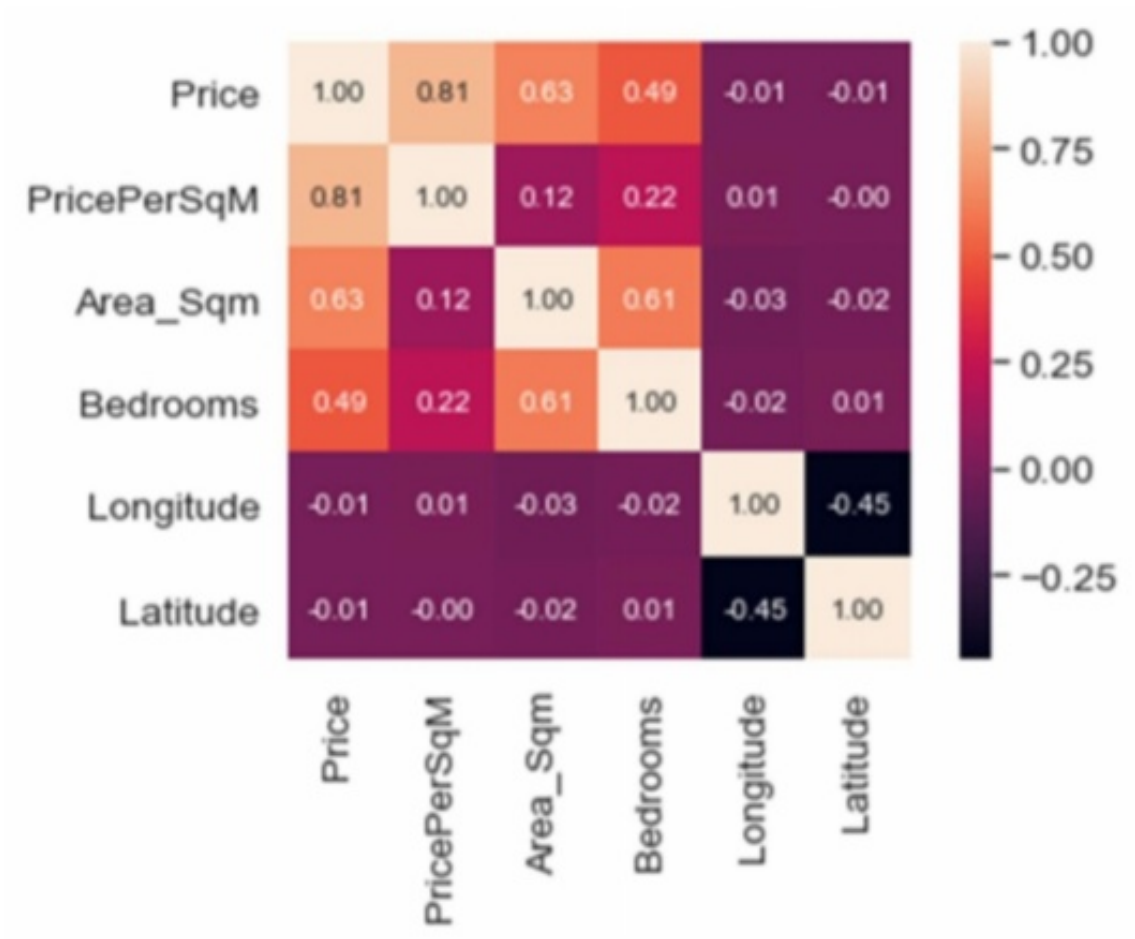
## **Data Selection**

Data selection is defined as the process of determining the appropriate data type and source, as well as suitable instruments to collect data. Data selection precedes the actual practice of data collection. This definition distinguishes data selection from selective data reporting (selectively excluding data that is not supportive of a research hypothesis) and interactive/active data selection (using collected data for monitoring activities/events, or conducting secondary data analyses). The process of selecting suitable data for a research project can impact data integrity.

The primary objective of data selection is the determination of appropriate data type, source, and instrument(s) that allow investigators to adequately answer research questions. This determination is often discipline-specific and is primarily driven by the nature of the investigation, existing literature, and accessibility to necessary data sources.

	Price	Area_Sqm	Bedrooms	Latitude	Longitude	PricePerSqM
0	13300000	74.32	2	19.210200	72.864891	178885.00
1	9000000	55.74	1	19.167700	72.949300	168566.16
2	9000000	43.66	1	19.130609	72.873816	206032.48
3	9000000	78.97	2	19.126005	72.825052	113926.88
4	100000000	464.51	5	19.075014	72.907571	215200.00

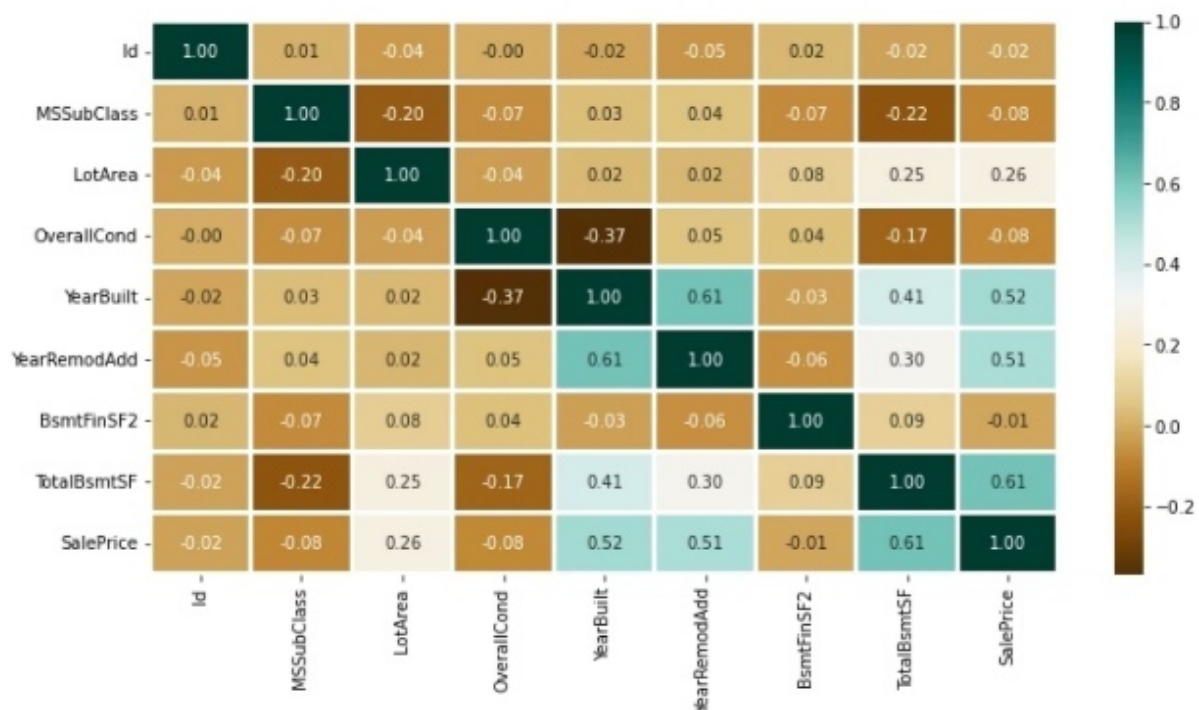
Correlation Heatmap





```
plt.figure(figsize=(12, 6))
sns.heatmap(dataset.corr(),
             cmap = 'BrBG',
             fmt = '.2f',
             linewidths = 2,
             annot = True)
```

**Output:**

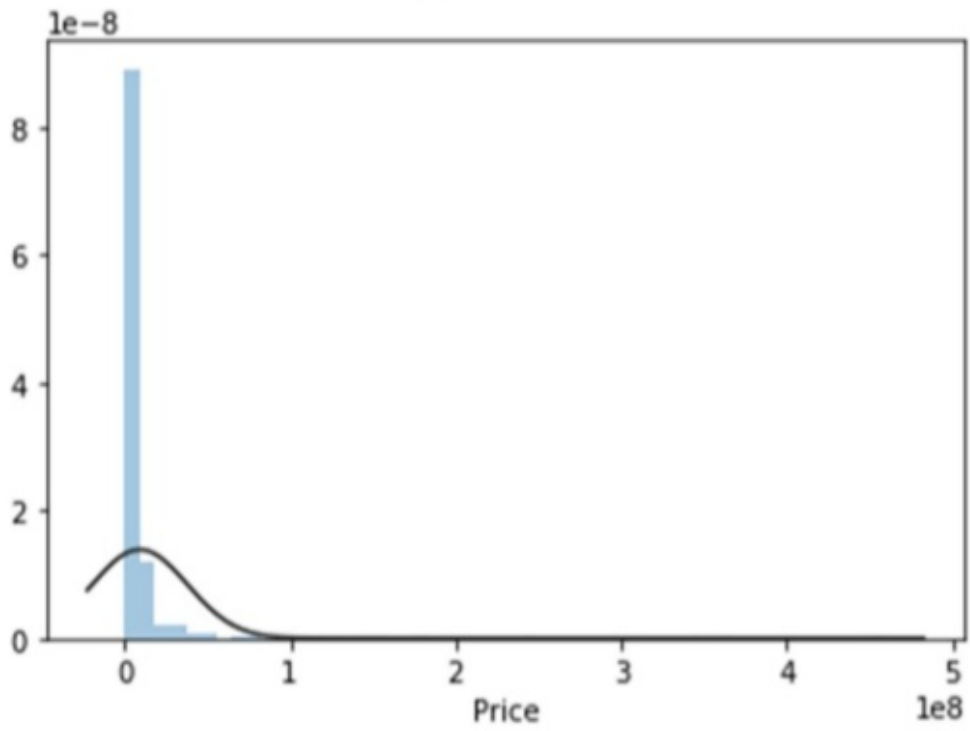


## **Data Transformation**

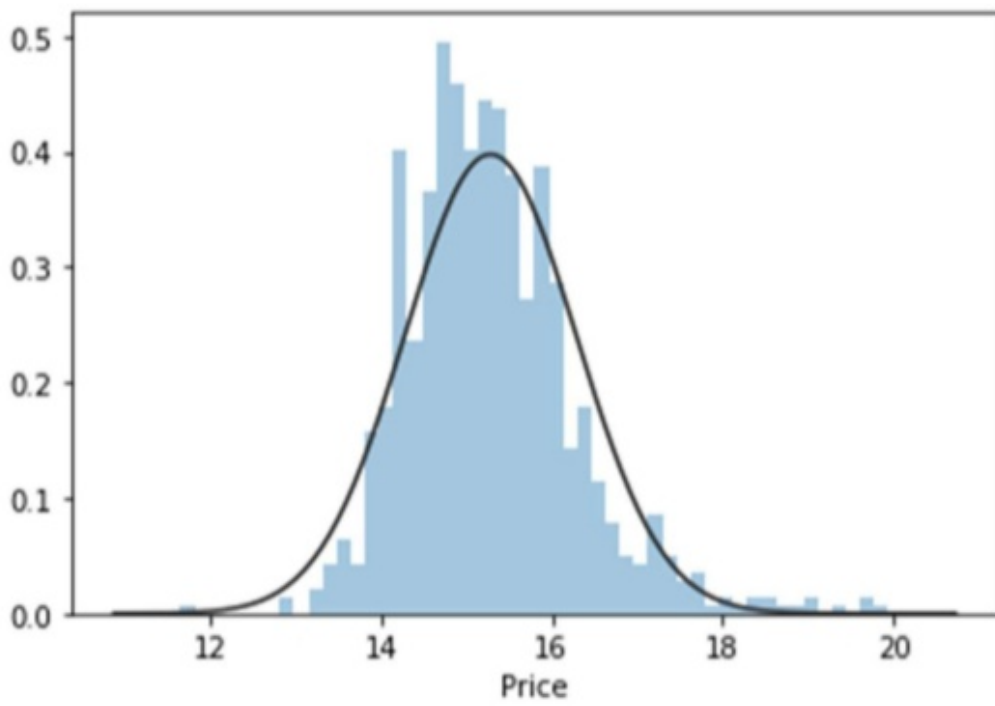
The log transformation can be used to make highly skewed distributions less skewed. This can be valuable both for making patterns in the data more interpretable and for helping to meet the assumptions of inferential statistics.

It is hard to discern a pattern in the upper panel whereas the strong relationship is shown clearly in the lower panel. The comparison of the means of log-transformed data is actually a comparison of geometric means. This occurs because, as shown below, the anti-log of the arithmetic mean of log-transformed values is the geometric mean.

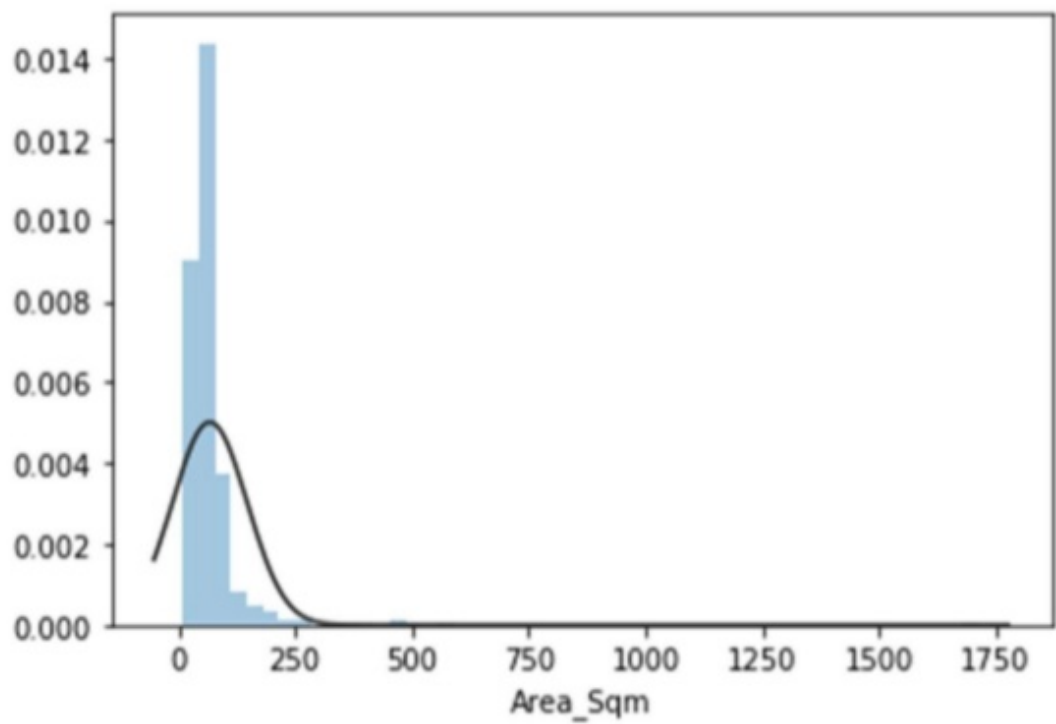
**Skewed Price**



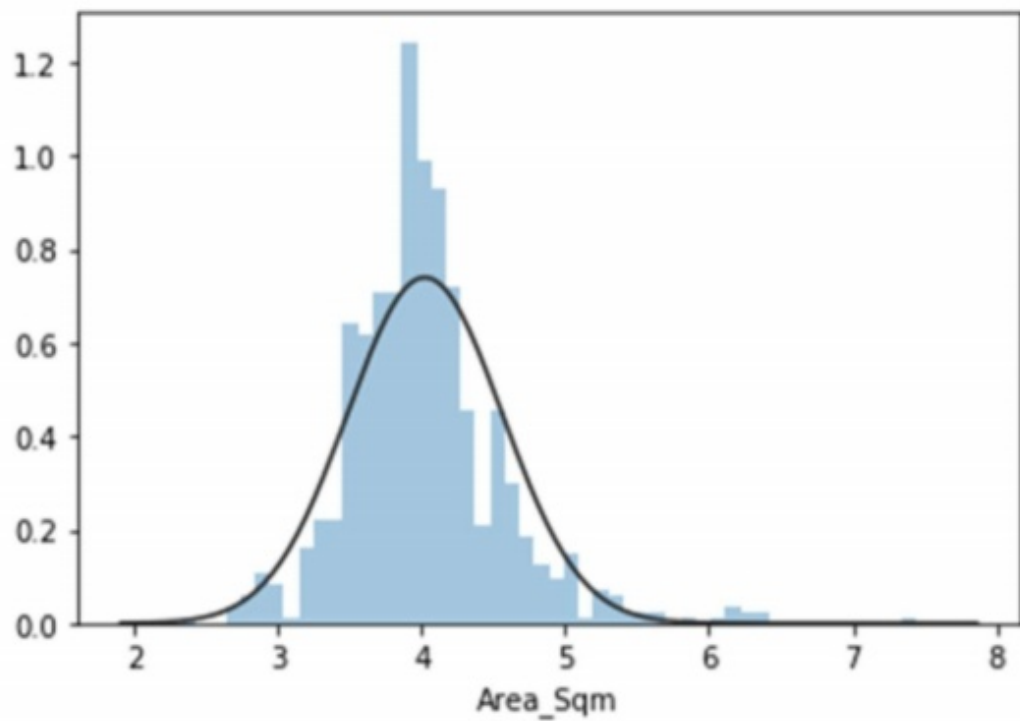
**Normal Price**



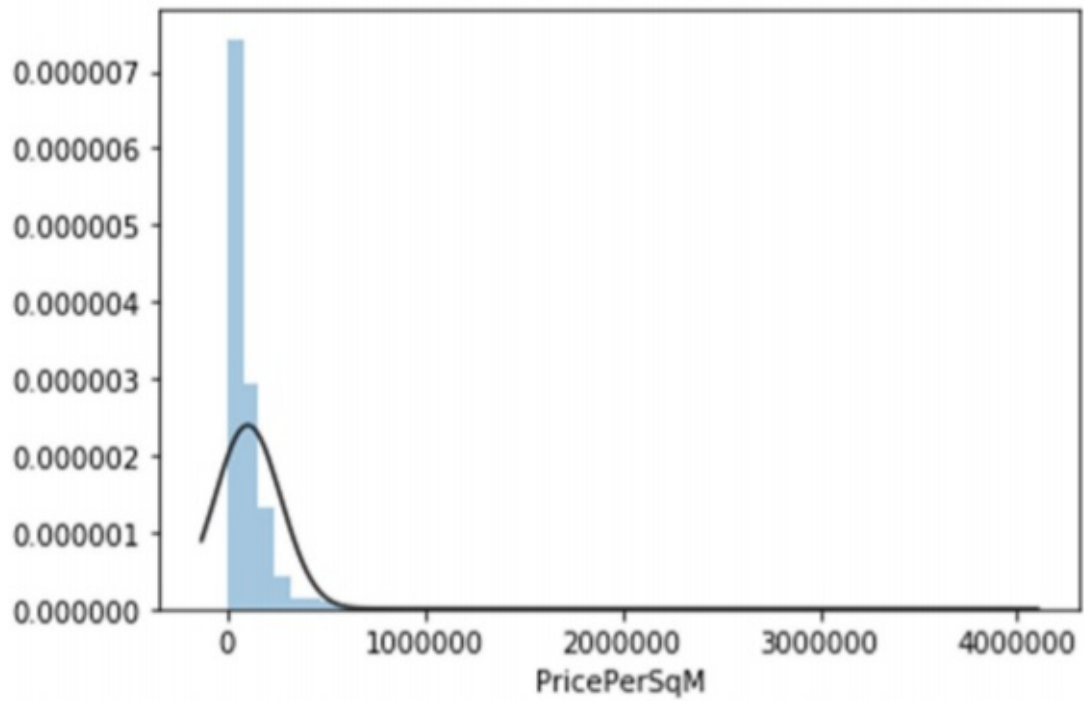
**Skewed Area**



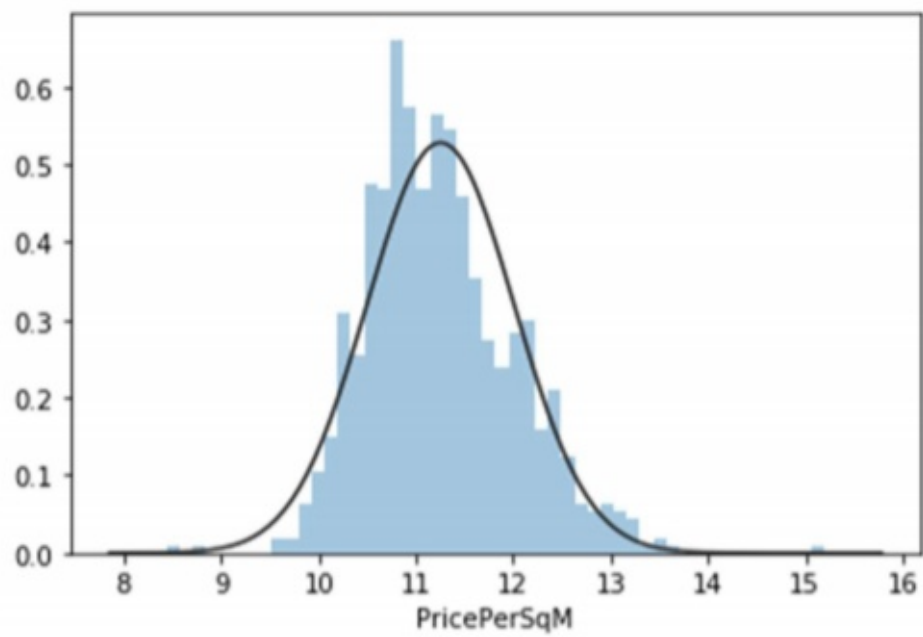
**Normal Area**



**Skewed Price/Sq.**



**Normal Price/Sq.**



**LANGUAGE AND MODELS USED**

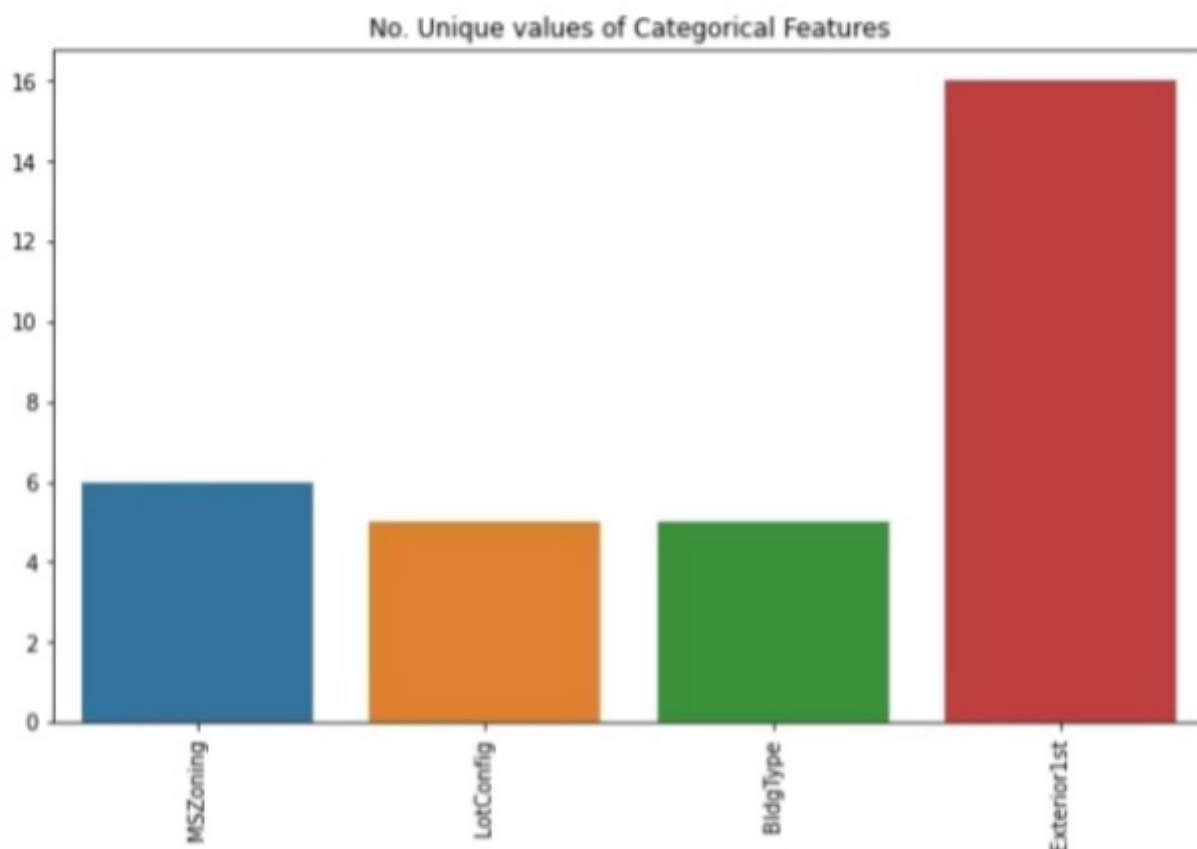


```

unique_values = []
for col in object_cols:
    unique_values.append(dataset[
plt.figure(figsize=(10,6))
plt.title('No. Unique values of
plt.xticks(rotation=90)
sns.barplot(x=object_cols,y=uni

```

## Output:



## **Python**

Python is widely used in scientific and numeric computing:

- SciPy is a collection of packages for mathematics, science, and engineering.
- Pandas is a data analysis and modelling library.
- IPython is a powerful interactive shell that features easy editing and recording of a work session, and supports visualizations and parallel computing.
- The Software Carpentry Course teaches basic skills for scientific computing, running bootcamps and providing open-access teaching materials.

### **Libraries Used for this Project include –**

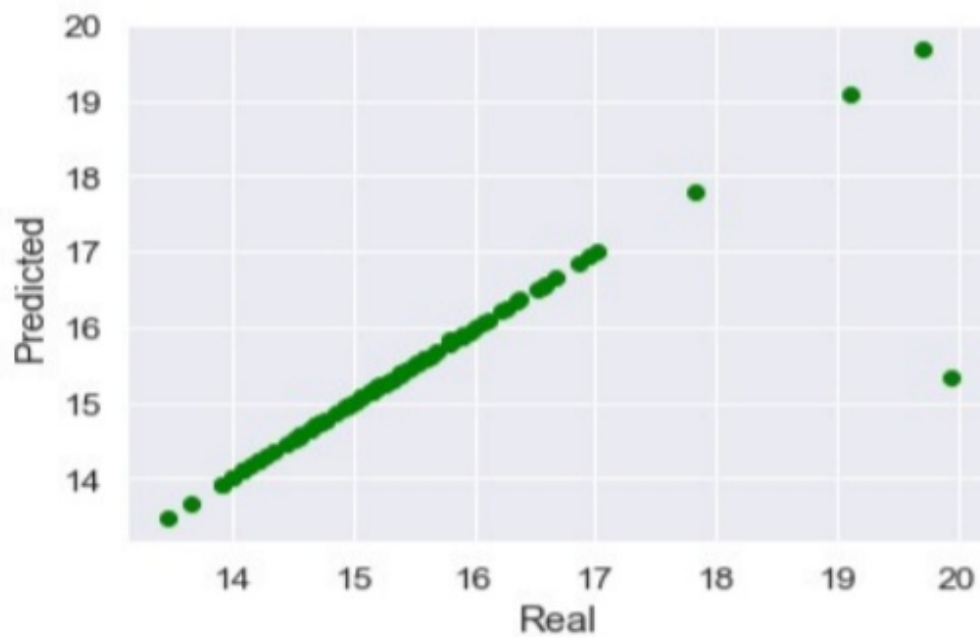
- **Pandas**
- **NumPy**
- **Matplotlib**
- **Seaborn**
- **Scikit Learn**
- **XG Boost**

# MODELS USED

## Regression Model

- 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.
- It is mostly used for finding out the relationship between variables and forecasting.

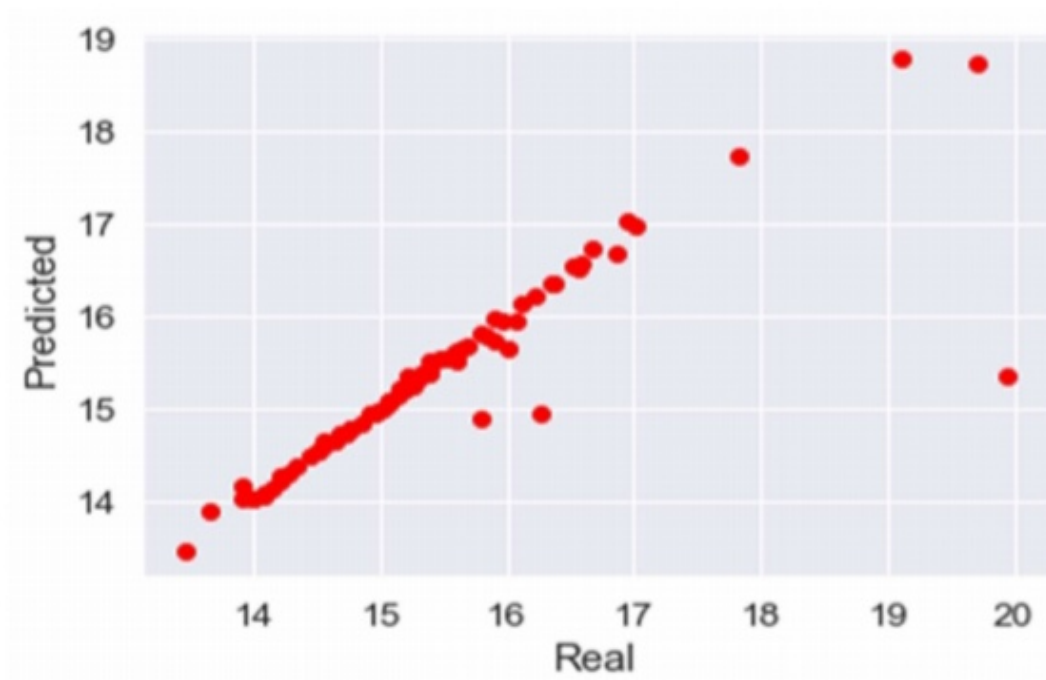
## Real Vs Predicted



## **Random Forest Regression Model**

- A Random Forest is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees and a technique called Bootstrap Aggregation, commonly known as bagging.
- Bagging, in the Random Forest method, involves training each decision tree on a different data sample where sampling is done with replacement.
- The basic idea behind this is to combine multiple decision trees in determining the final output rather than relying on individual decision trees.

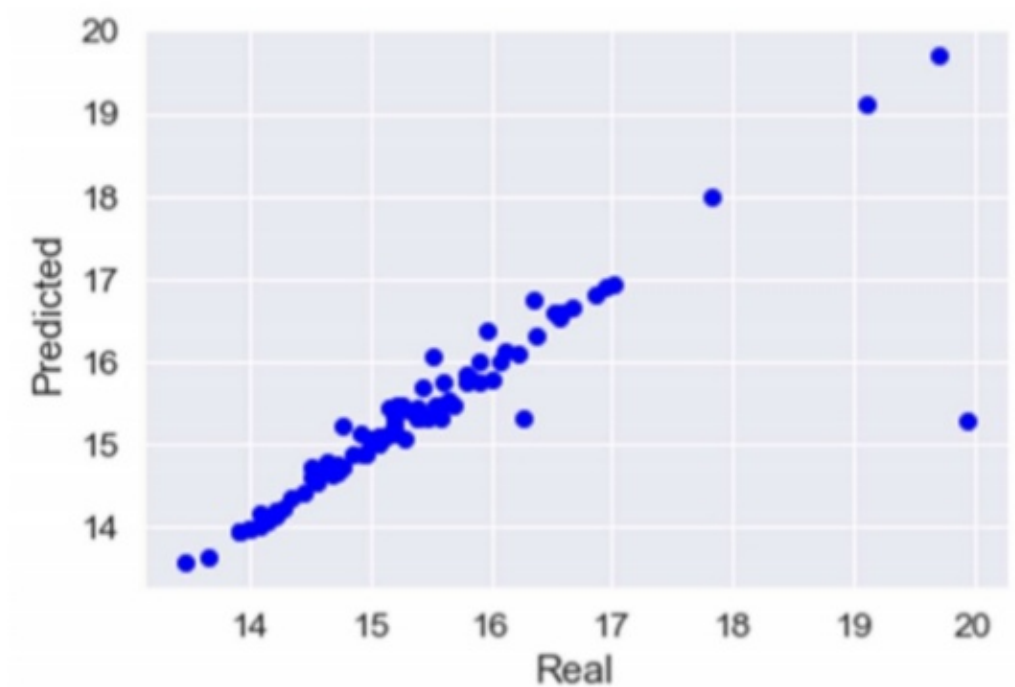
### **Real Vs Predicted**



## XG Boost Regressor Model

- XG Boost stands for eXtreme Gradient Boosting.
- The XG Boost library implements the gradient boosting decision tree algorithm.
- Boosting is an ensemble technique where new models are added to correct the errors made by existing models.
- Models are added sequentially until no further improvements can be made.

### Real Vs Predicted





## **RESULTS AND DISCUSSIONS**

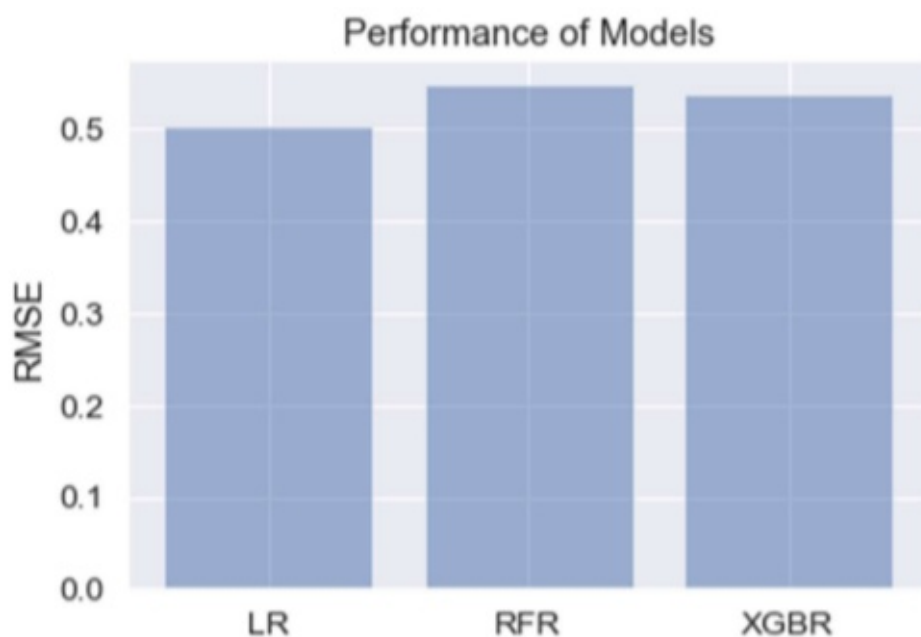
### **Best Suited Model**

So, our study showed that.....

Linear Regression displayed the best performance for this Dataset and can be used for deploying purposes.

Random Forest Regressor and XGBoost Regressor are far behind, so can't be recommended for further deployment purposes.

### **RMSE Bar Graph**



```

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

dataset = pd.read_excel("HouseP
# Printing first 5 records of t
print(dataset.head(5))

```

## Output:

	MSSubClass	MSZoning	LotArea	LotConfig	BldgType	OverallCond	YearBuilt
0	60	RL	8450	Inside	1Fam	5	2003
1	20	RL	9600	FR2	1Fam	8	1976
2	60	RL	11250	Inside	1Fam	5	2001
3	70	RL	9550	Corner	1Fam	5	1915
4	60	RL	14260	FR2	1Fam	5	2000

	YearRemodAdd	Exterior1st	BsmtFinSF2	TotalBsmtSF	SalePrice
0	2003	VinylSd	0.0	856.0	208500.0
1	1976	MetalSd	0.0	1262.0	181500.0
2	2002	VinylSd	0.0	920.0	223500.0
3	1970	Wd Sdng	0.0	756.0	140000.0
4	2000	VinylSd	0.0	1145.0	250000.0

## Deployment App

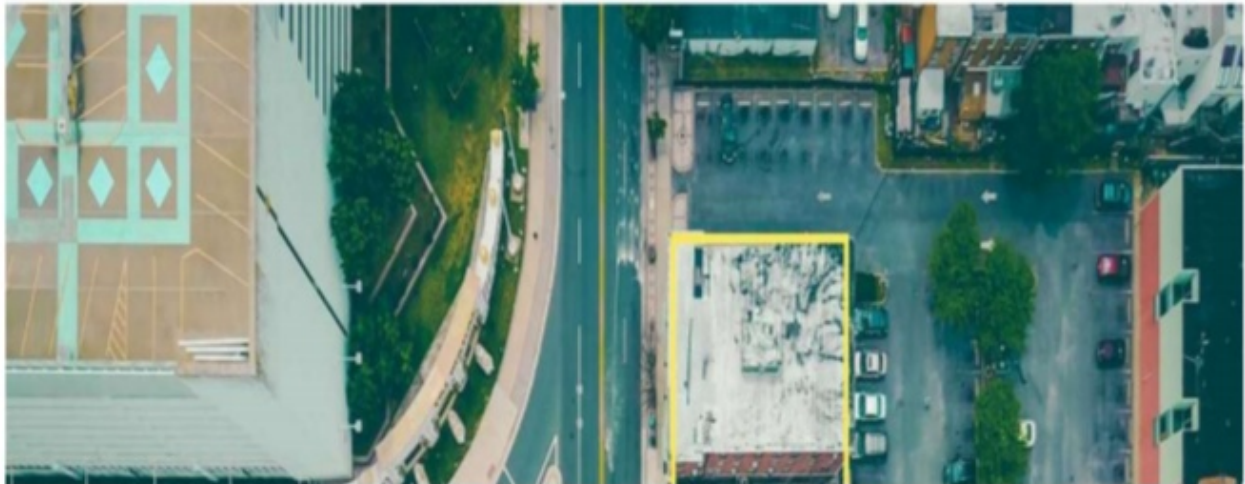
The Model is deployed through Python Web App Flask in collaboration with HTML and CSS.

Now anyone can Predict

Lets get to know future house prices

<input type="text" value="Area in SqM"/>	<input type="text" value="Bedrooms"/>	<input type="text" value="PricePerSqM"/>	<input type="text" value="Latitude"/>	<input type="text" value="Longitude"/>	<input type="button" value="Predict"/>
--	---------------------------------------	--	---------------------------------------	--	--

{{ prediction\_text }}



## **Conclusion**

So, our Aim is achieved as we have successfully ticked all our parameters as mentioned in our Aim Column. It is seen that circle rate is the most effective attribute in predicting the house price and that the Linear Regression is the most effective model for our Dataset with RMSE score of **0.5025658262899986**.