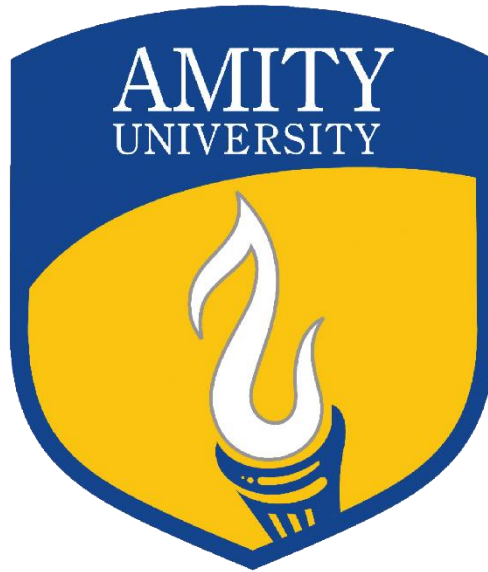


A Project Report
On
House Price Prediction using ML
Submitted to



Amity University Uttar Pradesh

By

Vaibhav Madan

Under the guidance of

Dr. Sumit Kumar

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
AMITY SCHOOL OF ENGINEERING AND TECHNOLOGY
AMITY UNIVERSITY UTTAR PRADESH

DECLARATION

I, Vaibhav Madan, student of B.Tech (5CSE-6Y) hereby declare that the project titled "**House Price Prediction using ML**" which is submitted by me to the Department of Computer Science and Engineering, **Amity School of Engineering and Technology**, Amity University Uttar Pradesh, Noida, in partial fulfilment of the requirement for the award of the degree of Bachelor of Technology in Computer Science and Engineering has not been previously formed the basis for the award of any degree, diploma or other similar title or recognition.

The author attests that permission has been obtained for the use of any copyrighted material appearing in the Dissertation / Project report other than brief excerpts requiring only proper acknowledgment in scholarly writing and all such use is acknowledged.

Vaibhav Madan

A2305220394

5CSE-6Y (2020-24)

CERTIFICATE

This is to certify that **Mr. Vaibhav Madan**, student of B. Tech in Computer Science and Engineering has conducted work presented in the project of the In-House Practical Training titled "**House Price Prediction using ML**" as a part of the Third-year program of Bachelor of Technology in Computer Science and Engineering from Amity University, Uttar Pradesh, Noida under my supervision.

Dr. Sumit Kumar

Associate Professor

Department of Computer Science and Engineering

ASET, Noida

ACKNOWLEDGEMENT

I would like to thank Prof (Dr) Abhay Bansal, Head of Department (CSE), and Amity University for allowing me to undertake this project. I would like to thank my faculty guide Dr. Sumit Kumar, who is the biggest driving force behind my successful completion of this in-house project. He has always been there to solve any query of mine and guided me in the right direction regarding the project. Without her help and inspiration, I would not have been able to complete the term paper. Also, I would like to thank my batch mates who guided me, helped me, and gave ideas and motivation at each step.

Name: Vaibhav Madan

Enrollment Number: A2305220394

Batch: 5CSE6Y (2020-24)

Abstract

With regards to estimating values, evaluating images and videos for vital information, avoiding spam emails, and forecasting prices for analysts of stocks, real estate, autos, and many other things, machine learning has played a significant part in the development of recent innovations. In the real estate market, it's critical to understand a property's price in relation to its number of floors, bathrooms, and rooms as well as exterior factors. We attempt to forecast house values in this project using the data set's information, which is crucial for both buyers and sellers. Knowing the anticipated outcome is highly useful.

We used the dataset from city of Bengaluru with factors like total area, no. of balconies, no. of bathrooms, availability, society, type of area etc. to train the model to guess the correct prices. By applying Data cleaning techniques, we optimize the data set by removing outliers and empty/null values so that our model is accurate and has fewer errors in detecting prices. We use Linear Regression in this model to get accurate results. Linear regression is a supervised learning algorithm, that can be implemented on labelled data, which can be used to give output as a value. After making the model and extracting the cleaned data set, we build a web app to utilize the model which will help users to predict house prices by computing the values for the factors affecting the prices, this application is hosted online on StreamLit to easily access it from anywhere on a browser

Table of Contents

Declaration by the Student.....	2
Certificate by the Faculty Guide.....	3
Acknowledgement.....	4
Abstract.....	5
Chapter 1:	
Introduction.....	9
1.1 Background	
1.2 Motivation	
1.3 Scope	
1.4 Objectives	
Chapter 2:	
Literature Review.....	10
Chapter 3:	
Methodology.....	11
3.1 Software Used	
3.2 Programming Language Used	
3.3 Data Set	
3.4 Libraries used in code	
3.5 Flow Chart of Project	
3.6 Implementation	

Chapter 4:

Result.....	20
-------------	----

Chapter 5:

Conclusion.....	21
-----------------	----

Chapter 6:

References.....	22
-----------------	----

Table of Figures

Fig 3.1 Notebook in Jupyter.....	11
Fig 3.2 Visual Studio code.....	12
Fig 3.3 Data Set.....	13
Fig 3.4 Size of Data set.....	13
Fig 3.5 Description of Data set.....	13
Fig 3.6 Flow Chart.....	15
Fig 3.7 Dropping unimportant factors.....	15
Fig 3.8 Finding NULL values and dropping them.....	16
Fig 3.9 Removing NULL values by replacing them with mean.....	16
Fig 3.10 Correlation Matrix.....	17
Fig 3.11 Using Quantile Flooring and Capping.....	17
Fig 3.12 Removing outliers using Mean/Median Imputation.....	18
Fig 3.13 Train/Test Splitting.....	18

Fig 3.14 R^2 Test values for train and test split.....	19
Fig 3.15 MSE test values for train and test split.....	19
Fig 4.1 Size of Cleaned Data set.....	20
Fig 4.2 Actual vs. Predicted line plot.....	20
Fig 4.3 Web App 1.....	21
Fig 4.4 Web App 2.....	21

1. Introduction

1.1 Background

At this stage in life, purchasing a home may be a very tedious endeavor. You must research the costs of various neighbourhoods and look at numerous houses that may fit your budget and your ideal location. You must accurately forecast the price of the home you wish to purchase. The average price of real estate in seven Indian cities increased by 38%. Bengaluru's average home price increased by 49%, from Rs. 3,345 per square foot to Rs. 4,975 per square foot [2].

1.2 Motivation

The goal of this project is to develop a machine learning model that aids users in predicting the house price using the provided data set based in Bengaluru, India, so that individuals can sell or purchase homes with the proper price in mind and avoid overpaying or underpaying for their properties. Users can anticipate property prices properly without having to delve through complex code and manually update information by creating a web app based on the model.

1.3 Scope

Based on the provided data set, which contains different attributes like the number of rooms, balconies, bathrooms, kitchens, the area the house is built in, the type of house, and other information, this application's machine learning model can determine house prices in Bengaluru. We will utilize the model by creating a Web app which will help make it simple for users to access their estimated home prices.

1.4 Objectives

To assist the residents of Bengaluru, the application must be able to identify appropriate property prices based on specified attributes in the dataset. The aim is to make the web app widely accessible so that everyone in Bengaluru may use it to forecast property prices when they need to sell their homes or acquire new ones. This would enable them to have the appropriate amount of cash on hand.

2. Literature Review

Machine Learning is a subset of Artificial Intelligence (AI) that is based on the use of data and algorithms to copy how the humans think and learn to gradually improve its accuracy.

Machine Learning is a key feature of the data science field. With the use of different statistical methods and techniques, different algorithms are trained to classify or predict values to get insights into a project [8]. These insights help in making decisions in different applications. Machine Learning falls into three categories that are Supervised learning, Unsupervised Learning, and Semi-supervised learning. The type of machine learning we use in this project is supervised learning which uses labeled data to help train algorithms to get a value as an output. This type of learning helps in many real-world problems such as classifying spam in a separate folder from your inbox. Some methods used in supervised learning are Linear Regression, Random Forest, etc. [3]

A machine learning approach called linear regression predicts a value based on the independent variables in a data set. The input value and the output value are linearly related according to the linear regression model. There is a hypothesis function in the linear regression algorithm.[7]

$$h_{\theta}(x) = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4 + \dots + \theta_n x_n$$

$$\text{Parameters: } \theta = \{\theta_0, \theta_1, \theta_2, \dots, \theta_n\}$$

$$\text{Features: } x = \{x_0, x_1, x_2, \dots, x_n\}$$

Here, we need to identify the parameter to determine which model fits us best, and the method employs a cost function to obtain the appropriate parameter values.[7]

$$\frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$$

Where the hypothesis $h_{\theta}(x)$ is given by the linear model

$$h_{\theta}(x) = \theta^T x = \theta_0 + \theta_0 x_1$$

To check the accuracy of our model we use the r-square test which gives shows us how fit our model is. The model should not be over-fitted or under-fitted, or this will result in inaccuracy.

We need the r-square value to be closest to 1 which is an ideal case that shows us that the model is perfectly fitted and is not inaccurate. We use MSE test also known as the Mean Square Error

test that calculates the average of the squares of all the errors in the model predictions to give us the accuracy of the model

3. Methodology

3.1 Software Used

- Jupyter Notebook

This is an interactive web-based platform for producing notebook files that may be used for ML and data research. In Jupyter, each file is referred to as a notebook and has an ordered list of inputs, outputs that are each included in a cell. These cells could include charts, text, numbers, and code. A notebook is a JSON file, typically ending in ".ipynb." [13]

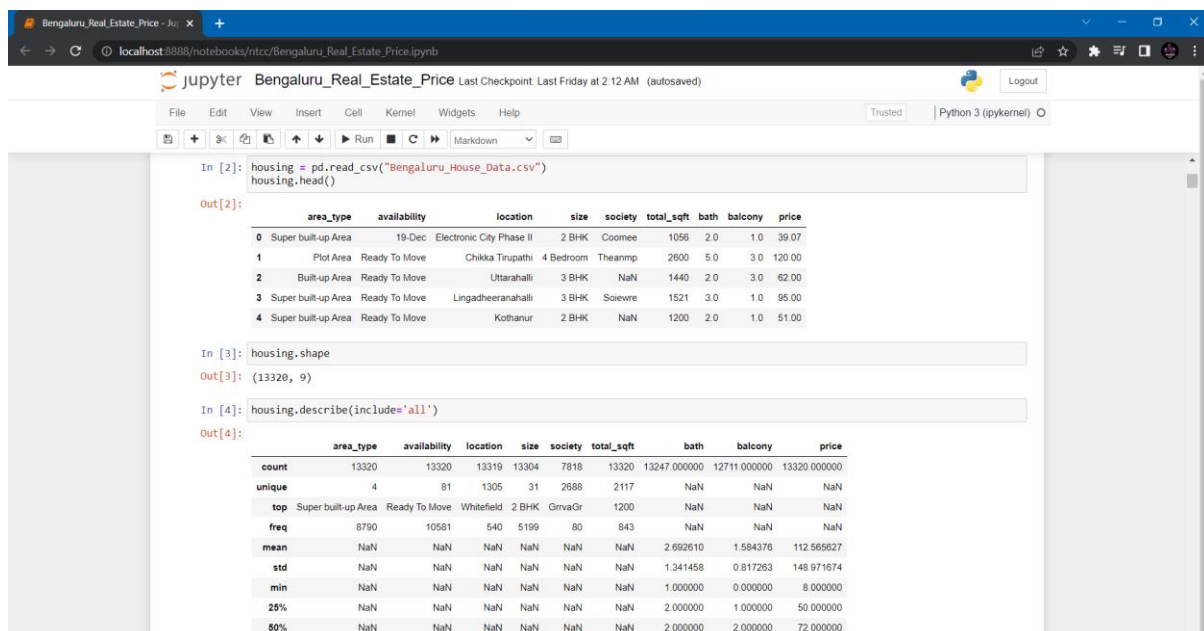


Fig 3.1 Notebook in Jupyter

- Visual Studio

Microsoft's Visual Studio is an IDE for creating, editing, testing, and publishing software on a variety of platforms (such as .NET, Unity, Universal Windows Platform, Xamarin. Forms, Blazor, etc.). Standard editors and debuggers are offered by the most famous integrated development environment. Compilers, code completion tools (like IntelliSense), visual designers (like Blend for Visual Studio), and direct interaction with Azure Integration and

capabilities are all available in Visual Studio. To ease simplicity of the software, it includes several more user-created features using the IDE's NuGet package management.

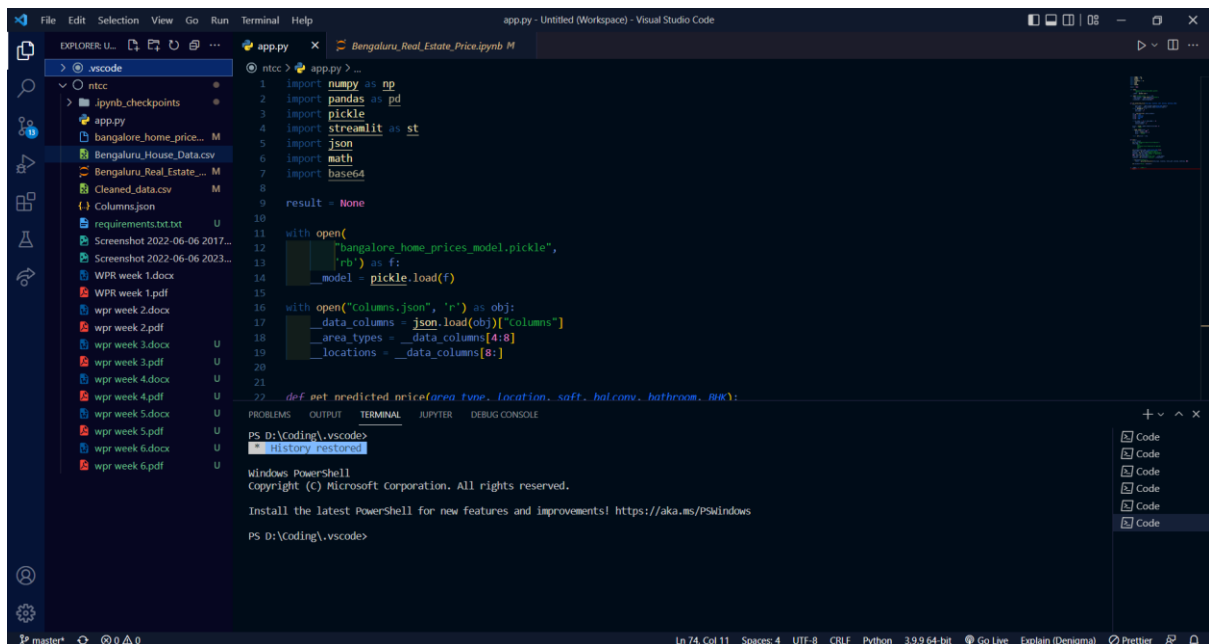


Fig 3.2 Visual Studio code

3.2 Programming Language Used

Python is a general-purpose, high-level, interpreted programming language. Code readability is prioritised in its design philosophy, which makes heavy use of indentation.

Python has garbage collection and dynamic typing. It supports a variety of programming paradigms, including procedural, object-oriented, and functional programming as well as structured programming (especially this). Due to its extensive standard library, it is frequently referred to as a "batteries included" language.

3.3 Data Set

The data collection comprises details about homes in Bengaluru, an Indian city. 13320 records detailing various homes along with their prices. Each home entry has unique characteristics, including the type of space, accessibility, location, scale, overall space, number of balconies, bathrooms, and social class. We choose this data set to show how a ML model can be created since it has all the important information about a house as well as some faults and missing data that can be filled in.

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

	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

Fig 3.3 Data Set

```
housing.shape
```

```
(13320, 9)
```

Fig 3.4 Size of Data set

```
housing.describe(include='all')
```

	area_type	availability	location	size	society	total_sqft	bath	balcony	price
count	13320	13320	13319	13304	7818	13320	13247.000000	12711.000000	13320.000000
unique	4	81	1305	31	2688	2117	NaN	NaN	NaN
top	Super built-up Area	Ready To Move	Whitefield	2 BHK	GrrvaGr	1200	NaN	NaN	NaN
freq	8790	10581	540	5199	80	843	NaN	NaN	NaN
mean	NaN	NaN	NaN	NaN	NaN	NaN	2.692610	1.584376	112.565627
std	NaN	NaN	NaN	NaN	NaN	NaN	1.341458	0.817263	148.971674
min	NaN	NaN	NaN	NaN	NaN	NaN	1.000000	0.000000	8.000000
25%	NaN	NaN	NaN	NaN	NaN	NaN	2.000000	1.000000	50.000000
50%	NaN	NaN	NaN	NaN	NaN	NaN	2.000000	2.000000	72.000000
75%	NaN	NaN	NaN	NaN	NaN	NaN	3.000000	2.000000	120.000000
max	NaN	NaN	NaN	NaN	NaN	NaN	40.000000	3.000000	3600.000000

Fig 3.5 Description of Data set

3.4 Libraries used in code

- NumPy

This library uses the Python programming language to manage data in big, multi-dimensional arrays and matrices as well as many other high-level mathematical operations.[12]

- Pandas

This Python programming language software package may be used in data analysis and manipulation. Additionally, it provides data structures for modifying tables.[11]

- Seaborn & Matplotlib

These are Python libraries that aid in making of statistical graphs and other types of plots, including scatterplots, boxplots, and line plots.[8]

- Scipy

This open-source library, which also includes modules for integration, algebra, and linear algebra, is also used for technical computing. among many other unique functions, optimization. [9]

- Scikit-learn

This is a machine learning library for Python. It includes different regression, classification techniques as well as support for the random forest, k-means, etc. [10]

- StreamLit Framework

The Python framework called StreamLit is used to create Web applications for ML and data science models. This framework makes it simple to create web applications and deploy them on StreamLit.

3.5 Flow Chart of Project

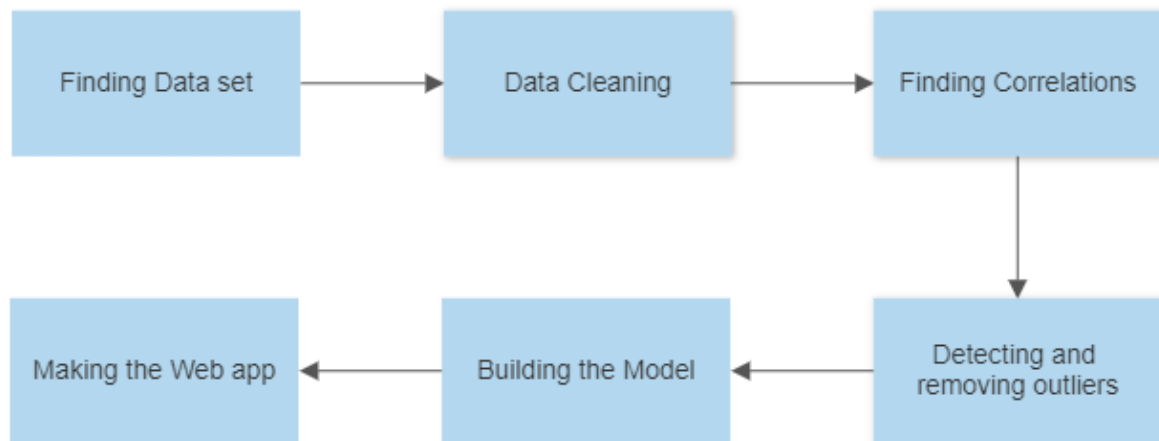


Fig 3.6 Flow Chart

3.6 Implementation

- Data Cleaning

We need to clean up our data set before we start building our model since it has a lot of flaws that make it difficult to utilize for the model. The first step is to identify any empty values and replace them either directly or by determining the mean of other values for that property. We find the NULL values by using python functions which tell us about the empty values in the data set then we check if the empty value is important or not if we need to keep the value then we take mean of all other values and exchange the missing values with the mean if it is not important, we can just drop the value. After that, we eliminate some aspects and qualities that are not crucial for developing our model, we drop two factors like society and availability which are not very important in predicting prices so we remove them from our optimized data set because it may affect our model and may cause inaccuracy. Ultimately, we must make sure that our data have comparable dimensions for example with the total area of a house so we need to define functions to convert all types of dimensions to a single type so that we can easily use that in the model rather than entering all types of different dimensions.

```
#Problem 1
housing_copy = housing.copy()
housing.drop(['availability', 'society'], axis=1, inplace=True)
```

Fig 3.7 Dropping unimportant factors

```
#Problem 2
housing.isnull().sum()

area_type      0
location       1
size           16
total_sqft     0
bath           73
balcony        609
price          0
dtype: int64

# Dropping Missing Values
housing.dropna(subset = ['location', 'size', 'bath'], inplace=True)
housing.shape

(13246, 7)
```

Fig 3.8 Finding NULL values and dropping them

```
#Replacing Missing Values with mean
housing['balcony'].replace(np.nan, housing['balcony'].mean(), inplace=True)

housing.isnull().sum()

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

Fig 3.9 Removing NULL values by replacing them with mean

- Finding Correlations

Finding some connections between the variables in the given data set will help us determine which one is most crucial and how its value affects the other variables. By creating a correlation matrix, we may determine how the values of one characteristic have changed w.r.t the values of other attributes. We can also add more factors so that we can see how the model is affected by our other factor. This model consists of factor like prices, area of property so we can make a factor of our own like price per sqft. which would help us to remove existing outliers from the data set to get better results for our model


```
housing['price_per_sqft'] = housing['price']*100000 / housing['total_sqft']
corr_matrix = housing[['total_sqft', 'bathroom', 'balcony', 'BHK', 'price', 'price_per_sqft']].corr()
corr_matrix['price'].sort_values(ascending=False)

price          1.000000
bathroom       0.456350
BHK            0.397222
balcony        0.104212
total_sqft     0.048798
price_per_sqft 0.028458
Name: price, dtype: float64
```

Fig 3.10 Correlation Matrix

- Finding and removing outliers

Outliers are the observation in a data set that lies far from the normal observations. These outliers need to be removed so that the model can be accurate. We detect outliers by using Box plots which show us how far the value is spaced in general. To remove outliers we can do this in different ways like removing the outlier observation directly like we did with empty values from the Data cleaning step or we can apply flooring and capping methods where we set values for floor and cap where we drop the values that are not in the scope of the floor or cap for example if the data set shows that a house containing 2 rooms and 10 bathrooms this shows that this observation is an outlier because it is abnormal to have 10 bathrooms so we can set a cap for this observation and remove all observations which are abnormal, another technique to remove outliers is to replace the abnormal values by mean or median of that respective factor.

```
tenth_percentile = housing['total_sqft'].quantile(0.10)
ninetieth_percentile = housing['total_sqft'].quantile(0.90)
housing['total_sqft'] = np.where(housing['total_sqft'] < tenth_percentile, tenth_percentile, housing['total_sqft'])
housing['total_sqft'] = np.where(housing['total_sqft'] > ninetieth_percentile, ninetieth_percentile, housing['total_sqft'])

housing['total_sqft'].describe()

count    13246.000000
mean     1438.315056
std      467.922813
min       900.000000
25%      1100.000000
50%      1275.000000
75%      1678.750000
max      2408.500000
Name: total_sqft, dtype: float64
```

Fig 3.11 Using Quantile Flooring and Capping

```

median = housing['bathroom'].quantile(0.50)
upper_out = housing['bathroom'].quantile(0.95)
housing['bathroom'] = np.where(housing['bathroom'] > upper_out, median, housing['bathroom'])

housing['bathroom'].describe()

```

count	13246.000000
mean	2.492300
std	0.889633
min	1.000000
25%	2.000000
50%	2.000000
75%	3.000000
max	5.000000
Name: bathroom, dtype: float64	

Fig 3.12 Removing outliers using Mean/Median Imputation

- Model Building

To begin creating the model, we first split the data set into two sections: the train split and the test split which can be done by using pre-built functions used the sklearn library or we can make our own function which splits the data set for us. It is crucial to split the data set evenly since otherwise; the model can be over or under-fitted. The data set's train split part is used to train the model algorithm to predict values and we can compare the results from the r^2 test of the train split with the test split to see if the model is accurate enough, and then we use the test split section to check if the model is working properly. In the end, we employ two techniques to evaluate the correctness of our model, which can be represented using various tests like the k-mean square error test which calculate the square of the errors in the model to show the level of accuracy of the model and r-square is used to show the fit of the model, the ideal value for this test is 1 but it is very difficult to attain the value of 1, both these test provide us with an numerical value to show accuracy of our model.

```

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.17, random_state=42)
print(f"{X_train.shape} \n {X_test.shape} \n {Y_train.shape} \n {Y_test.shape}")

```

(7178, 189)
 (1471, 189)
 (7178,)
 (1471,)

Fig 3.13 Train/Test Splitting

```
# R^2 Value - Train
r2_val_t = LinearModel.score(X_train, Y_train)
r2_val_t
```

```
0.8287986476450676
```

```
# R^2 Value - Test
r2_val = LinearModel.score(X_test, Y_test)
r2_val
```

```
0.8245817863750294
```

Fig 3.14 R^2 Test values for train and test split

```
: # mse value - Test
MSE = mse(Y_test, Y_pred)
MSE
```

```
: 669.3182392469604
```

```
: # mse value - train
MSE_t = mse(Y_train, Y_pred_train)
MSE_t
```

```
: 631.1215268148104
```

Fig 3.15 MSE test values for train and test split

- Building Web Application

We use the python framework StreamLit to make and deploy the Web App because it is easy for the user to use web applications on the internet and predict prices rather than inputting values manually in the code. StreamLit helps in making the web app as it is easy to use and works on python language, you can use pre-built functions to add checkboxes, line charts, select boxes, buttons. We must use the data and our model from our jupyter notebook and integrate these things in the pre-built functions to make the web app. Further we can upload the project on GitHub and can directly deploy the Web application form the streamlit website online.

4. Result

After performing Data cleaning and finding and removing outliers the Data set has reduced in size as you can see above in the figure. We successfully made the machine learning model that was made using linear regression operates flawlessly and makes extremely accurate predictions. Additionally, the web application utilized to make the model usable and accessible is fully functional.

```
housing.shape
```

```
(8649, 190)
```

Fig 4.1 Size of Cleaned Data set

The difference between the actual values and those predicted by the model is shown in the following graph, and as can be seen, the predicted values are nearly identical to the actual values.

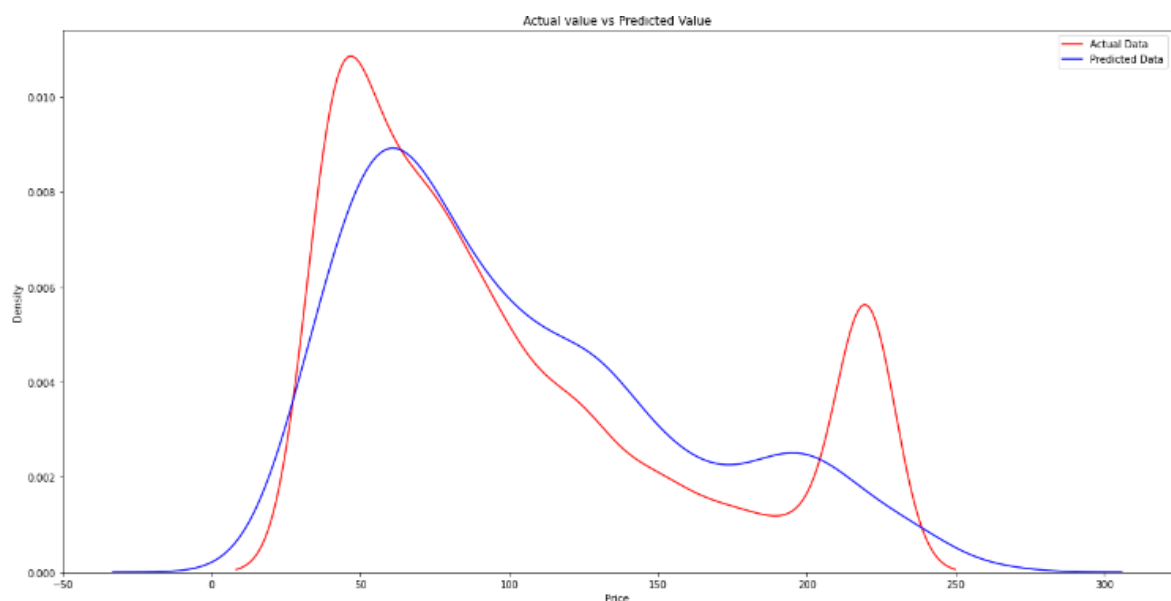
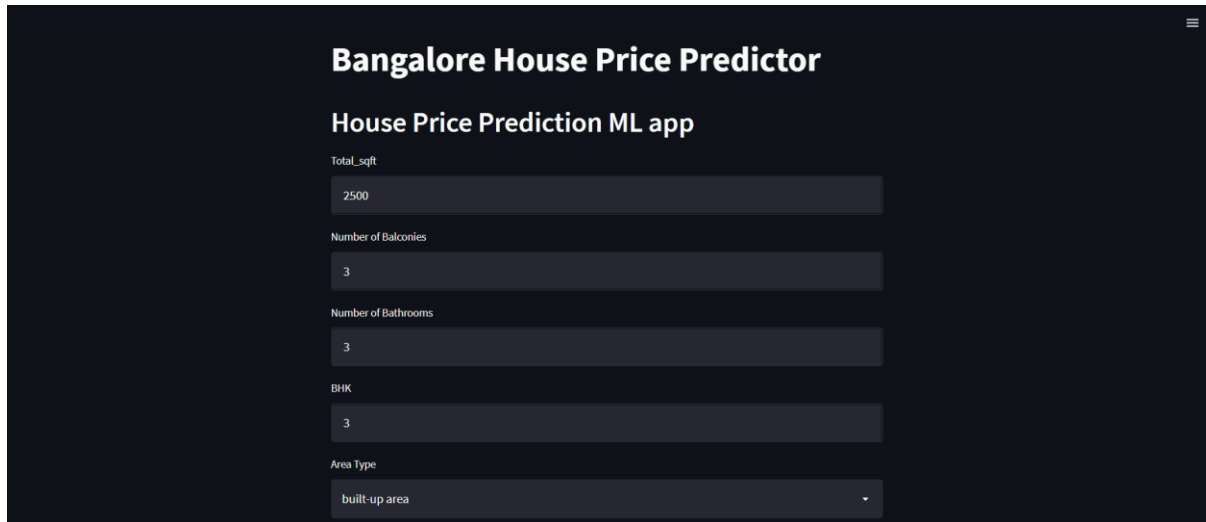


Fig 4.2 Actual vs. Predicted line plot

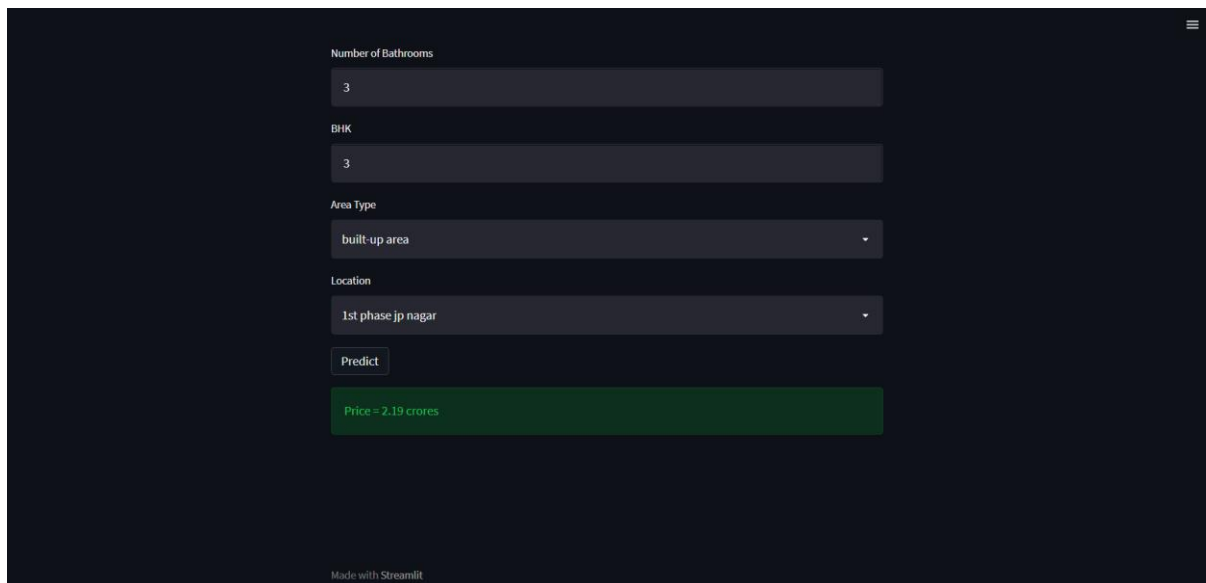
These are images of a web application that was made using Streamlit for user convenience. It also demonstrates how the model functions by predicting the price for the filled-in factors.



The screenshot shows the 'Bangalore House Price Predictor' web application. The title 'House Price Prediction ML app' is displayed below the main heading. The input fields are as follows:

Input Field	Value
Total_sqft	2500
Number of Balconies	3
Number of Bathrooms	3
BHK	3
Area Type	built-up area

Fig 4.3 Web App 1



The screenshot shows the same web application with the 'Predict' button clicked. The predicted price is displayed in a green box:

Input Field	Value
Number of Bathrooms	3
BHK	3
Area Type	built-up area
Location	1st phase jp nagar
Predict	Price = 2.19 crores

Fig 4.4 Web App 2

5. Conclusion

The project on house price prediction is now complete with the completion of the model and the web app. The general population of Bengaluru can use this model to examine the values of their homes if they want to sell them or purchase new ones. This project's restriction is that it only applies to one city, but we can improve its accuracy by applying more complicated machine learning techniques, integrating more datasets into the model to broaden the project's reach, and doing so by adding new cities to the project.

6. References:

- [1] *(PDF) house price prediction - researchgate*. (n.d.). Retrieved July 15, 2022, from https://www.researchgate.net/publication/349477129_House_Price_Prediction
- [2] *Average housing prices up 38 per cent in last decade*. The Economic Times. (n.d.). Retrieved July 13, 2022, from <https://economictimes.indiatimes.com/wealth/real-estate/average-housing-prices-up-38-per-cent-in-last-decade/articleshow/76339156.cms>
- [3] By: IBM Cloud Education. (n.d.). *What is machine learning?* IBM. Retrieved July 13, 2022, from <https://www.ibm.com/in-en/cloud/learn/machine-learning#:~:text=Machine%20learning%20is%20a%20branch,learn%2C%20gradually%20improving%20its%20accuracy>.
- [4] *House price prediction - diva portal*. (n.d.). Retrieved July 15, 2022, from <https://diva-portal.org/smash/get/diva2:1456610/FULLTEXT01.pdf>
- [5] *Housing price project report - math to power industry*. (n.d.). Retrieved July 15, 2022, from <https://m2pi.ca/project/2020/bc-financial-services-authority/BCFSA-final.pdf>
- [6] <https://www.ijitee.org/wp-content/uploads/papers/v2i1/a0363112112.pdf> ... (n.d.). Retrieved July 15, 2022, from https://www.bibsonomy.org/bibtex/2e66ba24f6dd60bfd3f7a50b934d41477/ijitee_beiesp?lang=ru
- [7] *ML: Linear regression*. GeeksforGeeks. (2022, May 18). Retrieved July 15, 2022, from <https://www.geeksforgeeks.org/ml-linear-regression/>
- [8] *Statistical Data Visualization*¶. seaborn. (n.d.). Retrieved July 15, 2022, from <https://seaborn.pydata.org/>
- [9] Wikimedia Foundation. (2022, April 11). *Scipy*. Wikipedia. Retrieved July 15, 2022, from <https://en.wikipedia.org/wiki/SciPy>
- [10] Wikimedia Foundation. (2022, January 14). *Scikit-Learn*. Wikipedia. Retrieved July 15, 2022, from <https://en.wikipedia.org/wiki/Scikit-learn>
- [11] Wikimedia Foundation. (2022, July 13). *Pandas (software)*. Wikipedia. Retrieved July 15, 2022, from [https://en.wikipedia.org/wiki/Pandas_\(software\)](https://en.wikipedia.org/wiki/Pandas_(software))
- [12] Wikimedia Foundation. (2022, July 8). *NumPy*. Wikipedia. Retrieved July 15, 2022, from <https://en.wikipedia.org/wiki/NumPy>
- [13] Wikimedia Foundation. (2022, June 27). *Project jupyter*. Wikipedia. Retrieved July 15, 2022, from https://en.wikipedia.org/wiki/Project_Jupyter