**Estimation of Housing Prices by Machine Learning**

**-by Lokesh Dandumahanti**

**Content**

1. Synopsis
2. Overview of Data
3. Data Cleaning (Handling NA values)
4. Feature Engineering
5. Dimensionality Reduction
6. Outlier Removal Using Business Logic
7. One Hot Encoding for location
8. Model Building
9. Accuracy
10. Testing Model
11. Export the tested model to pickle file
12. Stream lit Interface
13. Problems faced during project
14. Synopsis
15. Overview of data

For this taking, I have imported Bengaluru Housing Prices dataset from the Kaggle.com.

In this I have done the following processes

1. Importing necessary libraries
   1. Pandas
   2. Numpy
   3. Matplotlib
   4. Sklearn
2. Reading of csv file
3. Getting basic dimensions of the data
   1. Shape
   2. Columns
   3. Unique
   4. Value\_count
   5. Drop
4. Data Cleaning

As there were a lot of NA values, we need to clean the data prior further analysis. For this fillna and dropna are the two functions. As the NA values are comparatively lower(90) to the total number of values (13320), I have opted to drop the NA values.

1. Feature Engineering

Data present in the csv files cannot make sense for proper analysis. For analysis, we need more sophisticated features. For this we are going to add new features in the data i.e BHK and price per square feet. These two will be more humane and necessary parameters for a common person to do analysis.

1. BHK (Bedroom, Hall, Kitchen)

The size parameter is reformed into BHK parameter by taking the numerical values in the cell entries.

1. Price per square feet

Range of square feet is given. We will be taking mean of the range and then total price is being divided with the mean of range to give the approximate price per square feet value. These values are then added to a special column by name “price\_per\_sqft”.

1. Dimensionality Reduction

Most of the data points are useless for price estimations as their data point are quite less. In order to reduce complexity and increase the accuracy, we are going to remove areas have less than 10 data points.

1. Outlier removal using business logic

It is not possible that for an area of average 10000 Rs, to have an area of 1,00,000Rs. This is probably some human error while creating the dataset. By removing outliers in the dataset, we are going to improve the accuracy of the estimation

1. One Hot Encoding for location

For giving input to the machine learning model, it is necessary that all inputs need to be in numerical quantities. So in order to convert names of location to numerical equivalents, we need to use One Hot Encoding and integrate all the encodings as separate features into the excel file.

1. Model Building

In this, the cleaned csv file is used as input to create a machine learning model. For this we have chosen Linear Regression model.

1. Accuracy
2. Testing model
3. Export the tested model to a pickle format
4. Stream Lit Interface
5. Problems faced during the project

During the process of building this project, I have tried, failed, found out my mistake. I was in this loop multiple times till I succeeded. This hands on journey polished my skills and understanding of machine learning and its applications as a whole. Following are few of the problems I faced

1. Building an Interface for the model

Initially I though of using HTML, CSS and Javascript stack to build the website. Then of using nginx as server script and AWS EC model to deploy and publish it online. Due to lack on knowledge on integration of website to aws server, I had to quit the idea of publishing webapp in that way.

Instead I chose to deploy via Streamlit, an upcoming python library and server provider, whose library is very simple to understand and helps in constructing user interfaces in few lines of codes. This has been instrumental for me in several projects helping me compensate for my lack of expertise in web development.

As this approach is not industry approved and neither scalable, I would hone my skills in web development as well in my journey as Data analyst.

1. One Hot Encoding

This had been one of the challenging thing to implement while building interface. I had tough time figuring out on how to encode the locations so that it can be fed to machine learning algorithm. I tried multiple ways for it. First I thought of creating a separate csv file for OHE and other features. Then I thought of integrating both of them.

1. Loss of file

Due to some reason, jupyter file was not opening in my vscode. Hopefully I was resourceful and imported the file to google drive and then accessed it via google colab. Though after some reboots I was able to access my file on my vscode, but this left me with some valuable insights. Though this seems quite silly, but this made me understand the importance saving the progress in multiple locations.

1. Not having general idea of project

Due to this I had committed multiple errors in one hot encoding problem. By having a general idea of project through flowcharts of logic flow, the overall pipeline of the ML models, I believe I could have actually built the project in comparatively less time. By building project reports as the project progress, one can record one’s accomplishments and learning on that day. This would also help to revisit the whole code in case issues start occurring in the code in future.