**HOUSE PRICE PREDICTION USING MACHINE LEARNING**

**Prepared by: Tarun Mondal**

**Table of Contents**

[**Objective**](#_n2yopppgmbr5) **2**

[**Methodology**](#_4al1ey1rmq13) **2**

[**End-to-end process with solution architecture**](#_8n8seygnqb6h) **2**

[Importing libraries in Python](#_j4er0b6hqjd3) 2

[Data exploration](#_nq6j5jgfuivr) 2

[Data preprocessing](#_5bdlpe9xqeme) 4

[Exploratory data analysis (EDA)](#_4j1he0cm1jgd) 5

[Model development](#_xxsr8bt48lqc) 7

[**Challenges faced**](#_n66ncrpgyp64) **9**

[**Complexity level**](#_dhl7r1wlxkiy) **9**

# **Objective**

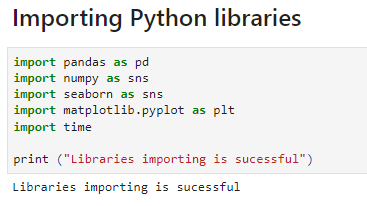
The objective is to predict house prices in Bangalore using a dataset of 7,120 entries and 108 features using KNN Regressor.

# **Methodology**

The project aims to predict Bangalore house prices using a dataset of 7,120 observations and 108 variables. The target variable in this project is continuous, indicating the suitability of regression algorithms. In this project, the KNN Regressor algorithm has been developed using the Scikit-learn framework in Python.

# **End-to-end process with solution architecture**

## **Importing libraries in Python**

****

**Figure 1: Importing libraries in Python**

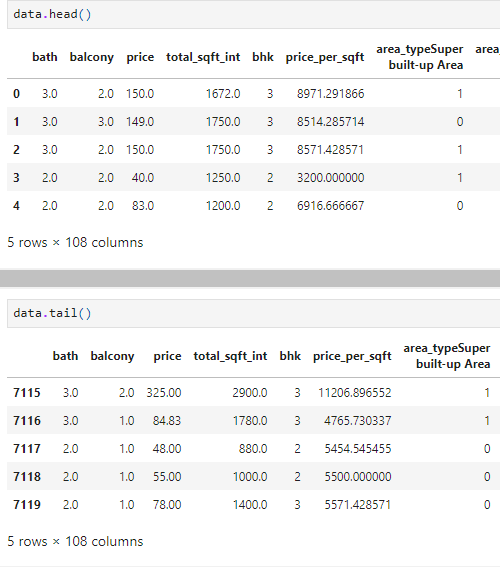
The necessary libraries, including Pandas for data handling, Seaborn and Matplotlib for visualisation, have been successfully imported.

## **Data exploration**

****

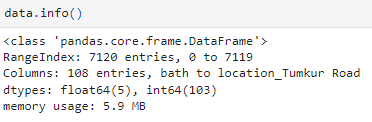
**Figure 2: Loading the dataset**

The dataset has been loaded in Python (Jupyter Notebook Environment) using the ‘read’ function from pandas library.

****

**Figure 3: Head and tail of the dataset**

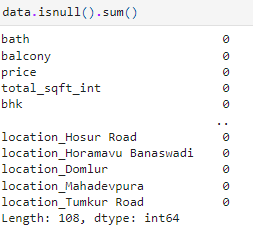
***Figure 3*** displays a real estate dataset, retrieved using the head () and tail () functions from the Pandas library. These functions are used to present a quick snapshot of the data by showing the first and last five records. The dataset contains various columns related to property details, such as the number of bathrooms, balconies, price, total square footage, number of bedrooms (BHK), price per square foot, area type, availability, and location. The location is encoded using one-hot encoding, with multiple columns representing different areas.

****

**Figure 4: Info of the dataset**

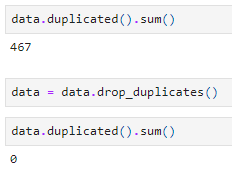
The shape and info of the dataset reveal that it contains 7,120 entries and 108 columns **(Refer to Figure 4).**

## **Data preprocessing**

****

**Figure 5: Data preprocessing**

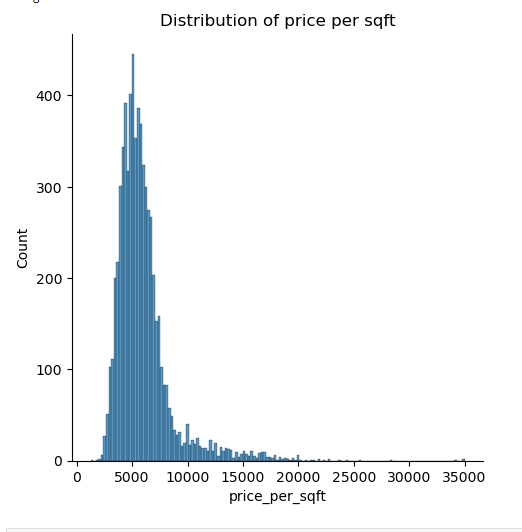
Missing values in the dataset have been checked using the isnull ().sum () function, revealing that none of the 108 columns contains any missing values. This indicates that the dataset is complete and free from null values, eliminating the need for data cleaning steps such as dropping or imputing missing values. Additionally, no mention of duplicate value checks is included in the current analysis, but the absence of missing data ensures that further preprocessing for handling null values is not necessary for this project.



**Figure 6: Duplicate values in the dataset**

**Figure 6** represents that the dataset initially contained 467 duplicate entries, which were removed using drop\_duplicates(). After removal, no duplicate records remain in the dataset.

## **Exploratory data analysis (EDA)**

****

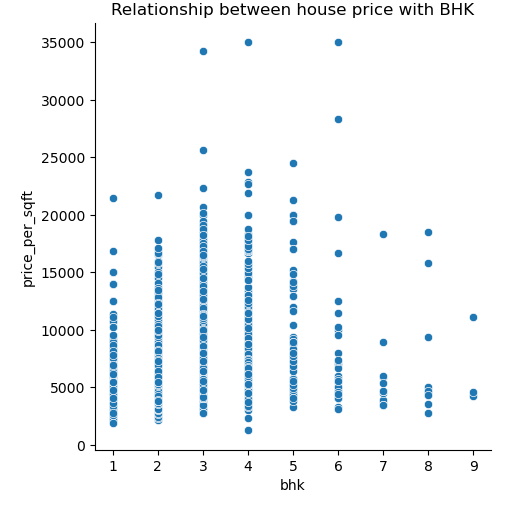
**Figure 7: Distribution plot for price\_per\_sqft**

Figure 7 shows a histogram depicting the distribution of property prices per square foot. Most properties have prices clustered between 2,000 and 10,000 per square foot, with fewer higher-priced outliers.

****

**Figure 8: Distribution plot for the relationship between house price and sqft**

The scatter plot between ‘price\_per\_sqft’ and ‘total\_sqft\_int’ revealed an approximately positive relationship between these two factors (**Refer to Figure 8**).

****

**Figure 9: Bar Plot for Relationship between house price with BHK**

The scatter plot in ***Figure 9*** shows the relationship between house price with BHK, where a linear relationship between BHK and house prices per sqft has been observed.

## **Model development**

****

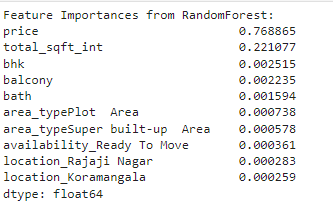
**Figure 10: Feature and target**

The target variable in this project is the ‘price\_per\_sqft’ **(Refer to Figure 10).**

****

**Figure 11: train-test splitting**

The data is split into 70% training and 30% testing using train\_test\_split, with random state 42 ensuring reproducibility.

****

**Figure 12: Feature Importances from RandomForest**

**Figure 12** shows that the feature importances from the RandomForest model indicate that 'price' is the most significant predictor, with 'total\_sqft\_int' also contributing notably, while other features have minimal impact.

****

**Figure 13: Feature and target**

Features Price, total\_sqft\_int, bhk, balcony, bath, Area are selected as X, and y represents the target, price\_per\_sqft **(Refer to Figure 10).**

****

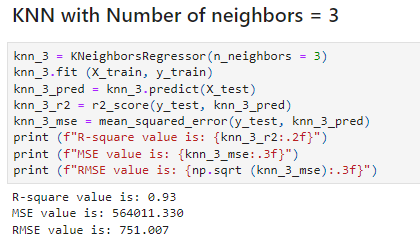
**Figure 14: train-test splitting**

The data is split into 80% training and 20% testing using train\_test\_split, with random state 1234 ensuring reproducibility **(Refer to Figure 14).**

****

**Figure 15: Model Performance**

Figure 15 shows The KNN model's performance metrics are: R-squared value of 0.91, Mean Squared Error (MSE) of 689,417.133, and Root Mean Squared Error (RMSE) of 830.311.

****

**Figure 16: Model Performance**

The KNN model with 3 neighbours has an R-squared value of 0.93, Mean Squared Error (MSE) of 564,011.330, and Root Mean Squared Error (RMSE) of 751.007 **(Refer to Figure 16).**

# **Challenges faced**

A key challenge was finding the optimal training-test split ratio to balance model performance and avoid overfitting or underfitting in KNN. major challenges faced in handling a large number of variables in the dataset. In order to tackle this challenge, feature selection has been performed using Random Forest regressor to identify the best possible features that contributed heavily to the fluctuation of house prices.

# **Complexity level**

The dataset contains 7,120 entries and 108 columns with no null values but 467 duplicates initially, which were removed. Despite being large and diverse, it’s manageable with a solid structure, allowing for effective data processing and analysis.