# CHENNAI HOUSE PRICE PREDICTION

**A MINI PROJECT REPORT**

**18CSC305J - ARTIFICIAL INTELLIGENCE**

***Submitted by***


RA2111003010264 Tejesh
RA2111003010265 VAISHNAV
RA2111003010220 ROJESH
RA2111003010214 LIKITHA


*Under the guidance of*

**Mrs. Deeba**

Assistant Professor, Department of Computer Science and Engineering

***in partial fulfillment for the award of the degree***

***of***

**BACHELOR OF TECHNOLOGY**

in

**COMPUTER SCIENCE & ENGINEERING**

of

**FACULTY OF ENGINEERING AND TECHNOLOGY**


S.R.M. Nagar, Kattankulathur, Chengalpattu District

**MAY 2024**

# SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

## BONAFIDE CERTIFICATE

Certified that Mini project report titled **"CHENNAI HOUSE PRICE PREDICTION"** is the bona fide work of Tejesh**(RA2111003010264), vaishanv(RA2111003010265), rojesh(RA2111003010220), Likitha(RA2111003010214)**who carried out the minor project under my supervision. Certified further, that to the best of my knowledge, the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

**SIGNATURE**
Mrs. Deeba
Assistant Professor

# ABSTRACT

This project endeavours to harness the capabilities of Machine Learning (ML) and Artificial Intelligence (AI) to predict housing prices in Chennai, a vibrant city known for its diverse economy and real estate market. By integrating various algorithms and predictive models, the project aims to analyse a multitude of factors that influence property values, including location, amenities, infrastructure, and historical price trends. Utilizing a rich dataset derived from Chennai's real estate listings, the project will employ supervised learning techniques to train models that can accurately forecast housing prices. The predictive power of ML and AI will provide valuable insights for potential buyers, investors, and policymakers, facilitating informed decision-making in the Chennai housing market. The project's outcome is expected to be a robust tool that not only predicts prices but also identifies underlying patterns and correlations within the data, offering a comprehensive understanding of the dynamics at play in Chennai's real estate sector. Through this endeavour, the project will contribute to the growing field of AI in urban planning and development, showcasing the potential of intelligent systems to revolutionize traditional industries.

# TABLE OF CONTENTS

# CHAPTER 1

# INTRODUCTION

The advent of Machine Learning (ML) and Artificial Intelligence (AI) has revolutionized numerous industries, offering predictive insights that were previously unattainable. Our project, centered on the vibrant city of Chennai, harnesses these cutting-edge technologies to predict housing prices with unprecedented accuracy. By integrating a myriad of variables such as location, market trends, and historical data, our ML models are trained to discern patterns and predict future market behaviors. This project not only exemplifies the practical application of AI in real estate but also serves as a benchmark for the predictive analytics capabilities within the domain. As urbanization accelerates, our solution aims to provide valuable assistance to potential home buyers, investors, and policy makers in making informed decisions. The 'Chennai House Prediction' project stands as a testament to the power of AI and ML in transforming data into actionable knowledge, driving the real estate sector towards a more data-driven and efficient future. This project aims to tackle this challenge head-on by employing advanced machine learning techniques to forecast the sales price of houses in Chennai. Utilizing a comprehensive dataset that encapsulates the essence of Chennai's housing market, we will embark on a journey through data preprocessing, exploratory data analysis, feature selection, and model building. Our arsenal will include a variety of regression models, each evaluated meticulously to ensure the highest predictive performance. As we navigate through the intricacies of the data, we will uncover the subtle nuances that drive property prices and strive to develop a model that not only predicts with precision but also provides valuable insights into the factors that shape Chennai's real estate landscape.

## CHAPTER 2

# LITERATURE SURVEY

| STUDY TITLE | AUTHORS | YEAR | METHODOLOGY | MAIN FINDINGS |
|---|---|---|---|---|
| House Price Prediction modeling using machine learning techniques: a comparative study | Ayten Yagmur<br><br>M kayakus | 2023 | Artificial Neural network (ANN) | The artificial neural network (ANN) method made predictions with more meaningful results compared to support vector regression (SVR) and multiple linear regression (MLR) . |
| House Price prediction using Multiple Linear Regression and KNN | E Endroyono | 2023 | Multiple Linear regression and KNN | This study uses backward elimination and forward selection methods to select the features used in this study. |
| A Novel model for house price prediction with machine learning techniques | Prof . Kanchan v.warkar | 2023 | Linear Regression and Random Forest | The main aim of the project is to predict the accurate price of the house without any loss and a comparative analysis of the results |
| House Price prediction using Data Mining with linear regression and ANN | Endang Palupi | 2023 | Linear Regression and ANN | This research aims to create a house price prediction model using the Linear Regression Algorithm and Neural Network so that the results can be useful for property agents in predicting house sales or from the buyer's side in predicting house prices. |
| Comparative Analysis of Random Forest Regression for House price prediction | Aman jha Obilisetti Lohith | 2023 | Random Forest | This research paper investigates the effectiveness of the Random Forest algorithm for house price prediction and conducts a comparative study with other regression algorithms. |

CHAPTER 3

# SYSTEM ARCHITECTURE AND DESIGN

## 1. Data Collection and Preprocessing :

**Data Sources:** Real estate websites, government databases, or APIs providing housing data for Chennai.

**Data Collection**: Web scraping or API calls to gather information on features like location, size, amenities, and historical prices.

**Data Preprocessing:** Cleaning data, handling missing values, encoding categorical variables, and scaling numerical features.

## 2. Feature Engineering:

**Feature Selection**: Identifying relevant features for prediction (e.g., area, number of bedrooms, proximity to amenities).

**Feature Transformation:** Engineering new features if needed (e.g., combining features, creating polynomial features).

## 3. Model Development:

**Model Selection:** Choosing appropriate machine learning algorithms for regression, such as linear regression, decision trees, random forests, or gradient boosting.

**Model Training:** Training the selected models on the preprocessed data.

Hyperparameter Tuning: Optimizing model parameters using techniques like grid search or random search.

**4. Model Evaluation:**

**Cross-Validation:** Evaluating model performance using techniques like k-fold cross-validation to ensure robustness.

**Performance Metrics:** Calculating metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), or Root Mean Squared Error (RMSE) to assess model accuracy.

**Visualization:** Visualizing predicted vs. actual prices, residual plots, and feature importances.

**5. Deployment:**

**Web Application:** Developing a web interface for users to input house features and get price predictions.

**API Development:** Building an API to serve predictions to other applications.

**6. System Architecture:**

**Frontend:** User interacts with the web application interface.

**Backend**: Handles requests from the frontend and interacts with the model for predictions.

**Model Server**: Hosts the trained model for making predictions.

Database: Stores historical data and possibly user preferences or feedback for future model updates.

**7. Technologies:**

**Programming Languages:** Python for model development, HTML/CSS/JavaScript for web development.

**Frameworks/Libraries:** Flask or Django for web development, scikit-learn or TensorFlow/PyTorch for machine learning, Pandas for data preprocessing, and Matplotlib/Seaborn for visualization.

**8. Maintenance and Updates:**

**Monitoring:** Regularly monitor model performance and user feedback for improvements.

**Model Updates:** Re-training the model periodically with new data to keep it up-to-date.

**Bug Fixes and Enhancements:** Addressing issues reported by users and adding new features as needed.

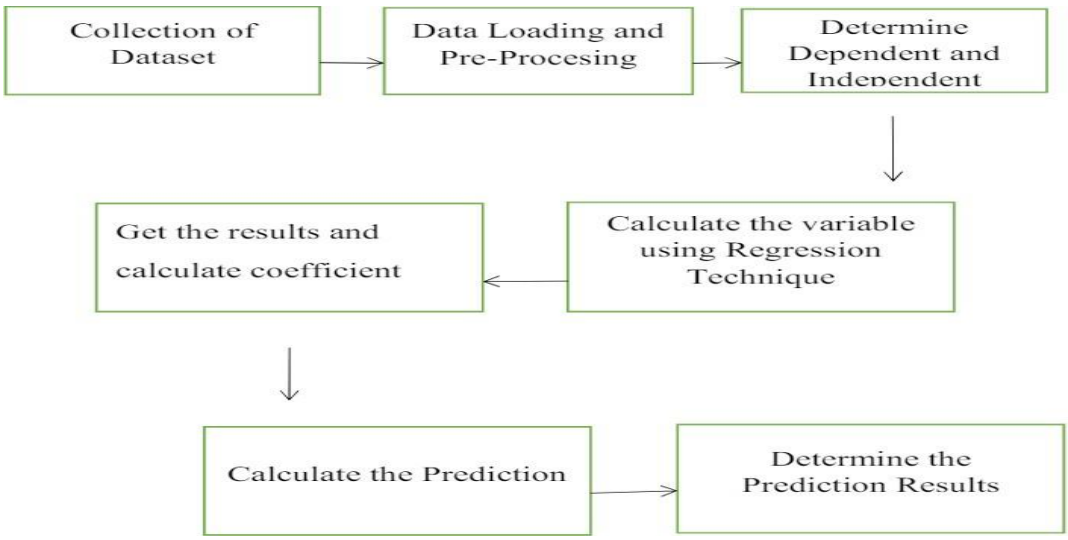



Fig 3.1 : Data Flow Diagram

CHAPTER 4

# METHODOLOGY

**Data Collection:** Gather a dataset of house listings in Chennai, including features like location, size, amenities, and historical prices.

**Data Preprocessing:** Clean the data to handle missing values, outliers, and categorical variables. Normalize or standardize numerical features if necessary.

**Feature Selection**: Identify the most relevant features that influence house prices using correlation analysis or feature importance methods.

**Model Selection:** Choose appropriate machine learning algorithms such as linear regression, decision trees, or ensemble methods like random forest.

**Model Training:** Split the dataset into training and testing sets. Train the selected models on the training set.

**Model Evaluation:** Use metrics like mean squared error (MSE), root mean squared error (RMSE), and mean absolute error (MAE) to evaluate model performance on the testing set.

**Model Optimization:** Fine-tune the models' hyperparameters to improve prediction accuracy.

**Deployment:** Deploy the best-performing model as a web application or service for end-users to predict house prices in Chennai.

# CHAPTER 5

# CODING AND TESTING

```
# This Python 3 environment comes with many helpful analytics libraries inst
alled
# It is defined by the kaggle/python Docker image: https://github.com/kaggle
/docker-python
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will l
ist all files under the input directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# You can write up to 20GB to the current directory (/kaggle/working/) that
gets preserved as output when you create a version using "Save & Run All"
# You can also write temporary files to /kaggle/temp/, but they won't be sav
ed outside of the current session
/kaggle/input/chennai-house-price/clean_data.csv
```

In [2]:

```
# importing required libraries
import pandas as pd
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

In [3]:

```
# Reading the cleaned data
df1 = pd.read_csv("/kaggle/input/chennai-house-price/clean_data.csv")
```

In [4]:

```
# Checking shape
df1.shape
```

Out[4]:

```
(2620, 8)
```

In [5]:

```
# Checking the data
df1.head()
```

Out[5]:

| | price | area | status | bhk | bathroom | age | location | builder |
|---|---|---|---|---|---|---|---|---|
| 0 | 37.49 | 872 | Ready to move | 2 | NaN | 1.0 | Sembakkam | MP Developers |
| 1 | 93.54 | 1346 | Under Construction | 3 | 2.0 | NaN | Selaiyur | DAC Promoters |
| 2 | 151.00 | 2225 | Under Construction | 3 | NaN | 0.0 | Mogappair | Casagrand Builder Private Limited |
| 3 | 49.00 | 1028 | Ready to move | 2 | 2.0 | 3.0 | Ambattur | Dugar Housing Builders |
| 4 | 42.28 | 588 | Under Construction | 2 | 1.0 | 0.0 | Pallavaram | Radiance Realty Developers India Ltd |

In [6]:

```
# Checking datatypes and null values
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2620 entries, 0 to 2619
Data columns (total 8 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   price     2620 non-null   float64
 1   area      2620 non-null   int64  
 2   status    2620 non-null   object 
 3   bhk       2620 non-null   int64  
 4   bathroom  1403 non-null   float64
 5   age       1729 non-null   float64
 6   location  2620 non-null   object 
 7   builder   2620 non-null   object 
dtypes: float64(3), int64(2), object(3)
memory usage: 163.9+ KB
```

In [7]:

```
df1.describe().T
```

Out[7]:

| | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| price | 2620.0 | 93.834683 | 113.609349 | 12.83 | 42.0 | 61.735 | 90.00 | 1422.0 |
| /area | 2620.0 | 1282.925191 | 692.566319 | 300.00 | 877.0 | 1091.500 | 1471.25 | 6700.0 |
| bhk | 2620.0 | 2.443893 | 0.811984 | 1.00 | 2.0 | 2.000 | 3.00 | 8.0 |
| bathroom | 1403.0 | 2.359230 | 0.844951 | 1.00 | 2.0 | 2.000 | 3.00 | 7.0 |
| age | 1729.0 | 1.355119 | 2.102682 | 0.00 | 0.0 | 0.000 | 3.00 | 32.0 |

In [8]:

```
# Filling null with -1
df1["bathroom"].fillna(-1, inplace = True)
df1["age"].fillna(-1, inplace = True)
```

In [9]:

```
# Function to add missing bathroom values
def bath_finder(x,y):
    if y == -1:
        if x >= 5:
            return x+1
        elif x == 4 | x == 3:
            return x
        elif x == 1:
            return x
        else:
            return x-1
    else:
        return y

df1["bath"] = df1.apply(lambda x: bath_finder(x["bhk"], x["bathroom"]), axi
s = 1)
```

In [10]:

```
df1.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2620 entries, 0 to 2619
Data columns (total 9 columns):
 #    Column     Non-Null Count   Dtype
-----  ---------     --------------------   ----------
 0    price      2620 non-null    float64
 1    area       2620 non-null    int64
```

```
 2   status     2620 non-null   object
 3   bhk        2620 non-null   int64
 4   bathroom   2620 non-null   float64
 5   age        2620 non-null   float64
 6   location   2620 non-null   object
 7   builder    2620 non-null   object
 8   bath       2620 non-null   float64
dtypes: float64(4), int64(2), object(3)
memory usage: 184.3+ KB
```

In [11]:

```
df1.head()
```

Out[11]:

| | price | area | status | bhk | bathroom | age | location | builder | bath |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 37.49 | 872 | Ready to move | 2 | -1.0 | 1.0 | Sembakkam | MP Developers | 1.0 |
| 1 | 93.54 | 1346 | Under Construction | 3 | 2.0 | -1.0 | Selaiyur | DAC Promoters | 2.0 |
| 2 | 151.00 | 2225 | Under Construction | 3 | -1.0 | 0.0 | Mogappair | Casagrand Builder Private Limited | 2.0 |
| 3 | 49.00 | 1028 | Ready to move | 2 | 2.0 | 3.0 | Ambattur | Dugar Housing Builders | 2.0 |
| 4 | 42.28 | 588 | Under Construction | 2 | 1.0 | 0.0 | Pallavaram | Radiance Realty Developers India Ltd | 1.0 |

In [12]:

```
# Function to add missing age values
def age_finder(x):
    if x == -1:
        return 0
    else:
        return x
df1['year'] = df1['age'].apply(age_finder)
```

In [13]:

```
df1.head()
```

Out[13]:

| | price | area | status | bhk | bathroom | age | location | builder | bath | year |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 37.49 | 872 | Ready to move | 2 | -1.0 | 1.0 | Sembakkam | MP Developers | 1.0 | 1.0 |
| 1 | 93.54 | 1346 | Under Construction | 3 | 2.0 | -1.0 | Selaiyur | DAC Promoters | 2.0 | 0.0 |
| 2 | 151.00 | 2225 | Under Construction | 3 | -1.0 | 0.0 | Mogappair | Casagrand Builder Private Limited | 2.0 | 0.0 |
| 3 | 49.00 | 1028 | Ready to move | 2 | 2.0 | 3.0 | Ambattur | Dugar Housing Builders | 2.0 | 3.0 |
| 4 | 42.28 | 588 | Under Construction | 2 | 1.0 | 0.0 | Pallavaram | Radiance Realty Developers India Ltd | 1.0 | 0.0 |

In [14]:

```
# Dropping old columns
df1.drop(['bathroom','age'],axis=1,inplace=True)
```

In [15]:

```
df1.head()
```

Out[15]:

| | price | area | status | bhk | location | builder | bath | year |
|---|---|---|---|---|---|---|---|---|
| 0 | 37.49 | 872 | Ready to move | 2 | Sembakkam | MP Developers | 1.0 | 1.0 |
| 1 | 93.54 | 1346 | Under Construction | 3 | Selaiyur | DAC Promoters | 2.0 | 0.0 |
| 2 | 151.00 | 2225 | Under Construction | 3 | Mogappair | Casagrand Builder Private Limited | 2.0 | 0.0 |
| 3 | 49.00 | 1028 | Ready to move | 2 | Ambattur | Dugar Housing Builders | 2.0 | 3.0 |

| | price | area | status | bhk | location | builder | bath | year |
|---|---|---|---|---|---|---|---|---|
| 4 | 42.28 | 588 | Under Construction | 2 | Pallavaram | Radiance Realty Developers India Ltd | 1.0 | 0.0 |

In [16]:
```
df1['bhk'].unique()
```

Out[16]:
```
array([2, 3, 4, 1, 5, 6, 8])
```

In [17]:
```
df1['bath'].unique()
```

Out[17]:
```
array([1., 2., 3., 4., 6., 5., 7.])
```

In [18]:
```
df1['year'].unique()
```

Out[18]:
```
array([ 1.,  0.,  3.,  6.,  5.,  2.,  4., 11.,  7., 13.,  9., 12., 16.,
       17.,  8., 32.])
```

In [19]:
```
df1['status'].unique()
```

Out[19]:
```
array(['Ready to move', 'Under Construction'], dtype=object)
```

In [20]:
```
df1['location'].unique()
```

Out[20]:
```
array(['Sembakkam', 'Selaiyur', 'Mogappair', 'Ambattur', 'Pallavaram',
       'Virugambakkam', 'Thirumazhisai', 'Moolakadai', 'Ottiyambakkam',
       'Perungalathur', 'Gerugambakkam', 'Anna Nagar', 'Gopalapuram',
       'Pammal', 'Porur', 'Navallur', 'Sholinganallur', 'Vanagaram',
       'T Nagar', 'Guindy', 'Madipakkam', 'Perumbakkam', 'Velachery',
       'Medavakkam', 'Gowrivakkam', 'Chromepet', 'Thalambur', 'Thandala
m',
       'Ayanambakkam', 'Kanathur Reddikuppam', 'Adyar', 'Kolapakkam',
       'Thaiyur', 'Thoraipakkam OMR', 'Poonamallee', 'Padur',
       'East Tambaram', 'Tiruvottiyur', 'Iyappanthangal', 'Padappai',
       'Mugalivakkam', 'Kelambakkam', 'Thirumullaivoyal', 'Madhavaram',
       'Kundrathur', 'Vengaivasal', 'Siruseri', 'Manapakkam',
       'Karapakkam', 'Sithalapakkam', 'Madambakkam', 'Perungudi',
       'Vadapalani', 'Alwarpet', 'Thiruvidandhai', 'Kotturpuram',
       'Velappanchavadi', 'Kilpauk', 'Besant Nagar', 'Vellakkal', 'Avad
i',
       'Kolathur', 'Thiruvanmiyur', 'Periyar Nagar', 'Nanmangalam',
       'Anakaputhur', 'Urapakkam', 'Kovur', 'Nungambakkam', 'Maduravoya
l',
       'Guduvancheri', 'Koyambedu', 'Korattur', 'Iyyappanthangal',
       'Madhavaram Milk Colony', 'Jeth Nagar', 'Mangadu',
```

```
       'Singaperumal Koil', 'Madhanandapuram', 'Kandigai', 'Veppampattu
',
       'Saidapet', 'Oragadam Village Ambattur Talu', 'Pattabiram',
       'Mahabalipuram', 'Ayapakkam', 'Chitlapakkam', 'Puzhal',
       'Washermanpet', 'Thiruverkadu',  'Pozhichalur',  'Mannur',
       'West Mambalam', 'Kattupakkam', 'Maraimalai Nagar', 'Chetpet',
       'Ramapuram', 'Alandur', 'Red Hills', 'K K Nagar', 'Purasawalkam'
,
       'Ramavaram', 'Shenoy Nagar', 'Saligramam', 'Royapettah',
       'Valasaravakkam', 'Semmancheri', 'Ottiyabakkam', 'Nanganallur',
       'Pallikaranai', 'Muttukadu', 'Tharapakkam', 'Egmore',
       'Anna Nagar East', 'Ayanavaram', 'Raja Annamalai Puram',
       'Aminjikarai', 'Purasaiwakkam', 'Periyapanicheri',
       'Kattankulathur', 'Kotivakkam', 'Kumananchavadi', 'Perumalpattu'
,
       'Kodambakkam', 'Sevvapet', 'tambaram west', 'Kovilambakkam',
       'Rajakilpakkam', 'Villivakkam', 'Chengalpattu', 'Cholambedu',
       'Nesapakkam', 'Padapai', 'Neelankarai', 'Vadanemmeli',
       'Kumaran Nagar', 'Ashok Nagar', 'Adambakam', 'Perambur',
       'Keelkattalai', 'Kodungaiyur West', 'Pudupakkam', 'Padi',
       'Mogappair East', 'Elandanur', 'Pazavanthangal', 'Thiruporur',
       'Moolacheri', 'Kolapakkam Vandalur', 'Veppampatttu', 'Manali',
       'Jamalia', 'Kottivakkam', 'Gokulapuram', 'Iyyapa Nagar',
       'West Tambaram', 'Old Pallavaram', 'Sriperumbudur',
       'Ponniammanmedu', 'Kovalam', 'CIT Nagar', 'Agaramthen',
       'Vengambakkam', 'Mambakkam', 'Nallambakkam', 'Peerakankaranai',
       'Rathinamangalam', 'Nandambakkam', 'Jafferkhanpet', 'Arumbakkam'
,
       'Kuthambakkam', 'Thirumalpur', 'Mahindra World City',
       'Oragadam Industrial Corridor', 'Ragavendra Colony',
       'Ambattur INDUSTRIAL ESTATE', 'Mannivakkam', 'Kil Ayanambakkam']
,
      dtype=object)
```

In [21]:

```
df1['builder'].unique()
```

Out[21]:

```
array(['MP Developers', 'DAC Promoters',
       'Casagrand Builder Private Limited', 'Dugar Housing Builders',
       'Radiance Realty Developers India Ltd', 'Traventure Homes Pvt Lt
d',
       'Urbanrise', 'Navin Housing Properties P LTD',
       'Jones foundation private limited', 'Isha Homes',
       'Kochar Homes Pvt Ltd', 'Pushkar Properties Pvt Ltd',
       'Asset Tree Homes', 'Urban Tree Infrastructures', 'Olympia Group
',
       'Vijay Raja Homes Private Limited', 'Kamalam Builder Pvt Ltd',
       'Appaswamy Real Estate', 'VNR Homes', 'PS Srijan Developers',
       'Lifestyle Housing', 'Puravankara Limited', 'Jones Foundations',
       'Plaza Group', 'Urbando Housing LLP', 'EK Realtors',
       'The Nest Builder', 'Doshi Housing', 'Grandstyle constructions',
```

```
       'vinoth builders', 'GTK Foundations', 'Baashyaam Group', 'chris'
,
       'Krishna Constructions', 'AKS Housing Dedvelopment Pvt Ltd',
       'Mayances Construction and Engineering Services',
       'Ramaniyam Real Estate Builders', 'India Builders Limited', 'Ram
',
       'Shatapatri Estates Pvt Ltd', 'Shri Raman Developers',
       'Sri Hari Developers', 'Radiance Realty Developers',
       'Pacifica Companies', 'BSCPL Infrastructure Ltd', 'Khurinji Home
s',
       'Saradeuz Realty Constructions', 'VGK Builders Pvt Ltd',
       'Amarprakash Developers Pvt Ltd', 'Hansa Estates',
       'Prince Foundations Ltd', 'Budget Housing And Properties',
       'SP Homes Pvt Ltd', 'Bharathi Construction',
       'Karuppaswamy Builders', 'Merlin Group', 'Advaita Homes',
       'Swamaan Developers', 'Poojaa Foundation',
       'Prestige Estates Projects Ltd', 'Global Homes', 'seller',
       'Pon Mariappan', 'HM Homes', 'INTERFACE PROPERTIES',
       'R Venkatesan', 'viswaraj', 'Shiva', 'S Suresh Kumar',
       'MEHTA REAL ESTATE CHENNAI LLP', 'Balamurugan', 'Alliance Group'
,
       'Royal Square', 'Proparena', 'Propsource Realty Private Limited'
,
       'Sandhya', 'Murali', 'Tellus Foundation', 'Jayakanthan',
       'Individual Agent', 'Naveen', 'Shanmugam Property',
       'Value reality', 'Swaminathan', 'smartassetsindia',
       'Sri Vinayaga Real Estate', 'Nagaraj', 'Karthick', 'Prasanna',
       'Vinay Asrani', 'Pragyansh', '24K Realtors', 'Kaushik associates
',
       'BricksBurg', 'Dinesh', 'AKS REALTY SERVICES', 'Kkk Landmark',
       'SS Square Property Developers', 'Prabha Homes', 'JA Associates'
,
       'Bala', 'Sarashwathi Construction', 'Selvakumar',
       'Right Angle Properties', 'Saravan gk', 'Evrostos Properties',
       'Balasubramani', 'Vishnu Foundation Ltd',
       'Shree sakthivel realestate', 'Yadhav constructions real estates
',
       'SS PROPERTIES', 'Prop Mart Technologies', 'Info Rich',
       'JD Properties', 'Mohan', 'HomeFirst', 'Elite nisha',
       'THAMINA HOMES', 'MC Foundation', 'ARB HOMES',
       'South Zone Realty Consulting Pvt Ltd', 'Luxclusive Homes',
       'Vishal D', 'mohammed', 'Dee Star Properties',
       'Chennai Gated Community', 'Balaji', 'MrPincode', 'GJ ESTATES',
       'Venkatesh', 'DJ Properties', 'Dhivagaran', 'REALTY INDIA',
       'MAXWORTH PROPERTIES', 'Velan Housing Properties'], dtype=object
)
```

In [22]:

```
df1['location'].value_counts()
```

Out[22]:

```
Veppampattu         149
```

```
Pammal               139
Medavakkam           111
Sholinganallur        99
Perungudi             88
                     ...
Jeth Nagar             1
Sevvapet               1
Gokulapuram            1
Iyyapa Nagar           1
Kil Ayanambakkam       1
Name: location, Length: 178, dtype: int64
```

In [23]:

```
df1['builder'].value_counts()
```

Out[23]:

```
seller                                484
MC Foundation                         232
Appaswamy Real Estate                 109
Propsource Realty Private Limited      79
Radiance Realty Developers             62
                                     ...
Prop Mart Technologies                  1
S Suresh Kumar                          1
Yadhav constructions real estates       1
MAXWORTH PROPERTIES                     1
Evrostos Properties                     1
Name: builder, Length: 135, dtype: int64
```

In [24]:

```
location_stats = df1['location'].value_counts(ascending=False)
location_stats
```

Out[24]:

```
Veppampattu          149
Pammal               139
Medavakkam           111
Sholinganallur        99
Perungudi             88
                     ...
Jeth Nagar             1
Sevvapet               1
Gokulapuram            1
Iyyapa Nagar           1
Kil Ayanambakkam       1
Name: location, Length: 178, dtype: int64
```

In [25]:

```
location_stats.values.sum()
```

Out[25]:

```
2620
```

In [26]:

```
len(location_stats[location_stats>10])
```

Out[26]:

```
59
```

In [27]:

```
len(location_stats)
```

Out[27]:

```
178
```

In [28]:

```
len(location_stats[location_stats<=10])
```

Out[28]:

```
119
```

In [29]:

```
# Storing locations with less than  10  count
location_stats_less_than_10 = location_stats[location_stats<=10]
location_stats_less_than_10
```

Out[29]:

```
Ambattur INDUSTRIAL ESTATE    10
Thandalam                     10
Urapakkam                     10
Saidapet                       9
Kilpauk                        9
                              ..
Jeth Nagar                     1
Sevvapet                       1
Gokulapuram                    1
Iyyapa Nagar                   1
Kil Ayanambakkam               1
Name: location, Length: 119, dtype: int64
```

In [30]:

```
# Grouping all less than 10 locations as other
df1.location = df1.location.apply(lambda x: 'other' if x in location_stats_
less_than_10 else x)
len(df1.location.unique())
```

Out[30]:

```
60
```

In [31]:

```
df1.head(10)
```

Out[31]:

| | price | area | status | bhk | location | builder | bath | year |
|---|---|---|---|---|---|---|---|---|
| 0 | 37.49 | 872 | Ready to move | 2 | Sembakkam | MP Developers | 1.0 | 1.0 |
| 1 | 93.54 | 1346 | Under Construction | 3 | Selaiyur | DAC Promoters | 2.0 | 0.0 |
| 2 | 151.00 | 2225 | Under Construction | 3 | Mogappair | Casagrand Builder Private Limited | 2.0 | 0.0 |
| 3 | 49.00 | 1028 | Ready to move | 2 | Ambattur | Dugar Housing Builders | 2.0 | 3.0 |
| 4 | 42.28 | 588 | Under Construction | 2 | Pallavaram | Radiance Realty Developers India Ltd | 1.0 | 0.0 |
| 5 | 188.00 | 2221 | Under Construction | 3 | other | Traventure Homes Pvt Ltd | 3.0 | 0.0 |
| 6 | 38.00 | 885 | Under Construction | 3 | Thirumazhisai | Urbanrise | 2.0 | 0.0 |
| 7 | 72.99 | 936 | Ready to move | 3 | other | Navin Housing Properties P LTD | 2.0 | 6.0 |
| 8 | 125.00 | 2275 | Ready to move | 4 | other | Jones foundation private limited | 3.0 | 1.0 |
| 9 | 24.56 | 622 | Under Construction | 2 | Perungalathur | Isha Homes | 1.0 | 0.0 |

In [32]:

```
builder_stats = df1['builder'].value_counts(ascending=False)
builder_stats
```

Out[32]:

```
seller                                484
MC Foundation                         232
```

```
Appaswamy Real Estate                109
Propsource Realty Private Limited     79
Radiance Realty Developers            62
                                     ...
Prop Mart Technologies                 1
S Suresh Kumar                         1
Yadhav constructions real estates      1
MAXWORTH PROPERTIES                    1
Evrostos Properties                    1
Name: builder, Length: 135, dtype: int64
```

In [33]:

```
builder_stats = df1['builder'].value_counts(ascending=False)
builder_stats
```

Out[33]:

```
seller                               484
MC Foundation                        232
Appaswamy Real Estate                109
Propsource Realty Private Limited     79
Radiance Realty Developers            62
                                    ...
Prop Mart Technologies                 1
S Suresh Kumar                         1
Yadhav constructions real estates      1
MAXWORTH PROPERTIES                    1
Evrostos Properties                    1
Name: builder, Length: 135, dtype: int64
```

In [34]:

```
len(builder_stats[builder_stats>10])
```

Out[34]:

```
56
```

In [35]:

```
len(builder_stats)
```

Out[35]:

```
135
```

In [36]:

```
len(builder_stats[builder_stats<=10])
```

Out[36]:

```
79
```

In [37]:

```
# Storing builders with less than 10 count
builder_stats_less_than_10 = builder_stats[builder_stats<=10]
builder_stats_less_than_10
```

Out[37]:

```
SP Homes Pvt Ltd                     10
Prasanna                             10
EK Realtors                          10
Baashyaam Group                       9
GJ ESTATES                            9
                                     ..
```

```
Prop Mart Technologies                  1
S Suresh Kumar                          1
Yadhav constructions real estates       1
MAXWORTH PROPERTIES                     1
Evrostos Properties                     1
Name: builder, Length: 79, dtype: int64
```

In [38]:

```
# Grouping all less than 10 builders as other
df1.builder = df1.builder.apply(lambda x: 'other' if x in builder_stats_les
s_than_10 else x)
len(df1.builder.unique())
```

Out[38]:

```
57
```

In [39]:

```
df1[df1.area/df1.bhk<300].head()
```

Out[39]:

| | price | area | status | bhk | location | builder | bath | year |
|---|---|---|---|---|---|---|---|---|
| 4 | 42.28 | 588 | Under Construction | 2 | Pallavaram | Radiance Realty Developers India Ltd | 1.0 | 0.0 |
| 6 | 38.00 | 885 | Under Construction | 3 | Thirumazhisai | other | 2.0 | 0.0 |
| 43 | 42.28 | 588 | Under Construction | 2 | Pallavaram | Radiance Realty Developers | 1.0 | 0.0 |
| 54 | 29.62 | 528 | Ready to move | 2 | Pammal | Bharathi Construction | 1.0 | 1.0 |
| 206 | 35.34 | 594 | Ready to move | 2 | Vengaivasal | other | 1.0 | 2.0 |

In [40]:

```
# Removing Outliers
df2 = df1[~(df1.area/df1.bhk<300)]
df2.shape
```

Out[40]:

```
(2593, 8)
```

In [41]:

```
df2['price_per_sqft'] = df2['price']*100000/df2['area']
df2.head()
```

Out[41]:

| | price | area | status | bhk | location | builder | bath | year | price_per_sqft |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 37.49 | 872 | Ready to move | 2 | Sembakkam | MP Developers | 1.0 | 1.0 | 4299.311927 |
| 1 | 93.54 | 1346 | Under Construction | 3 | Selaiyur | DAC Promoters | 2.0 | 0.0 | 6949.479941 |
| 2 | 151.00 | 2225 | Under Construction | 3 | Mogappair | Casagrand Builder Private Limited | 2.0 | 0.0 | 6786.516854 |
| 3 | 49.00 | 1028 | Ready to move | 2 | Ambattur | other | 2.0 | 3.0 | 4766.536965 |
| 5 | 188.00 | 2221 | Under Construction | 3 | other | other | 3.0 | 0.0 | 8464.655561 |

In [42]:

```
# Removing Outliers
def remove_pps_outliers(df):
    df_out = pd.DataFrame()
    for key, subdf in df.groupby('location'):
        m = np.mean(subdf.price_per_sqft)
        st = np.std(subdf.price_per_sqft)
        reduced_df = subdf[(subdf.price_per_sqft>(m-st)) & (subdf.price_per
_sqft<=(m+st))]
        df_out = pd.concat([df_out,reduced_df],ignore_index=True)
    return df_out
df3 = remove_pps_outliers(df2)
df3.shape
```

Out[42]:

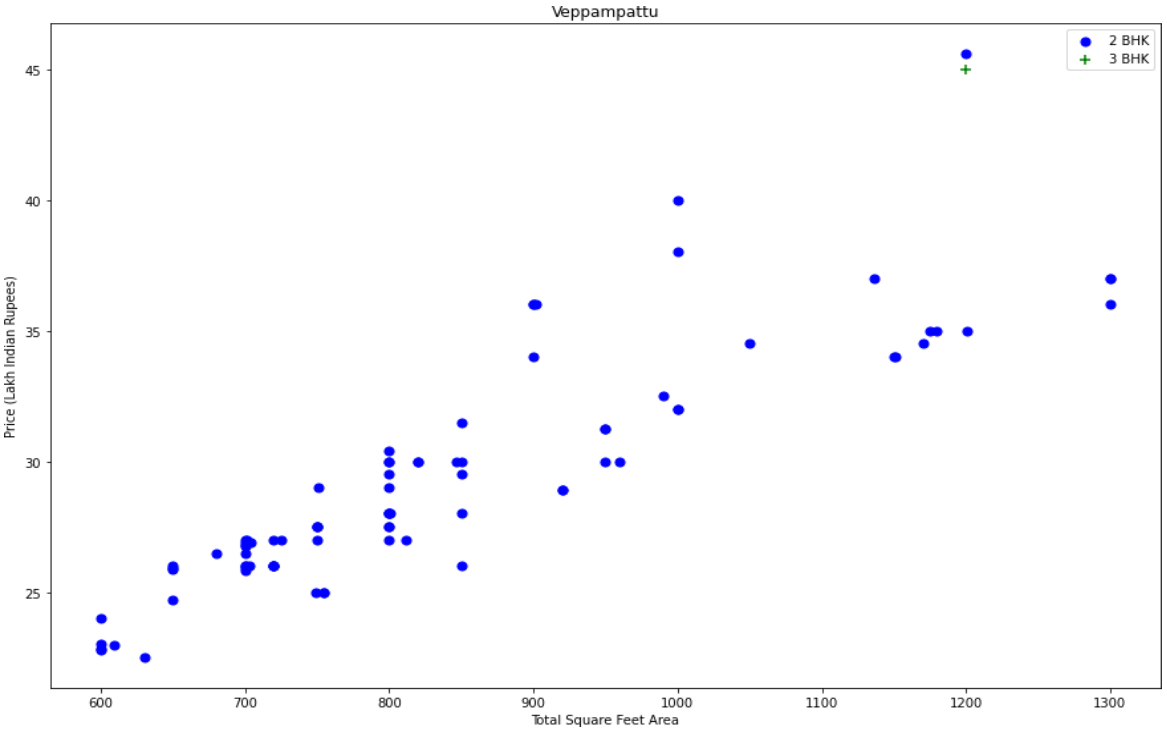
```
(2011, 9)
```

In [43]:

```
# Plotting scatter plot Area Vs. Price
def plot_scatter_chart(df,location):
    bhk2 = df[(df.location==location) & (df.bhk==2)]
    bhk3 = df[(df.location==location) & (df.bhk==3)]
    matplotlib.rcParams['figure.figsize'] = (15,10)
    plt.scatter(bhk2.area,bhk2.price,color='blue',label='2 BHK', s=50)
    plt.scatter(bhk3.area,bhk3.price,marker='+', color='green',label='3 BHK
', s=50)
    plt.xlabel("Total Square Feet Area")
    plt.ylabel("Price (Lakh Indian Rupees)")
    plt.title(location)
    plt.legend()
```

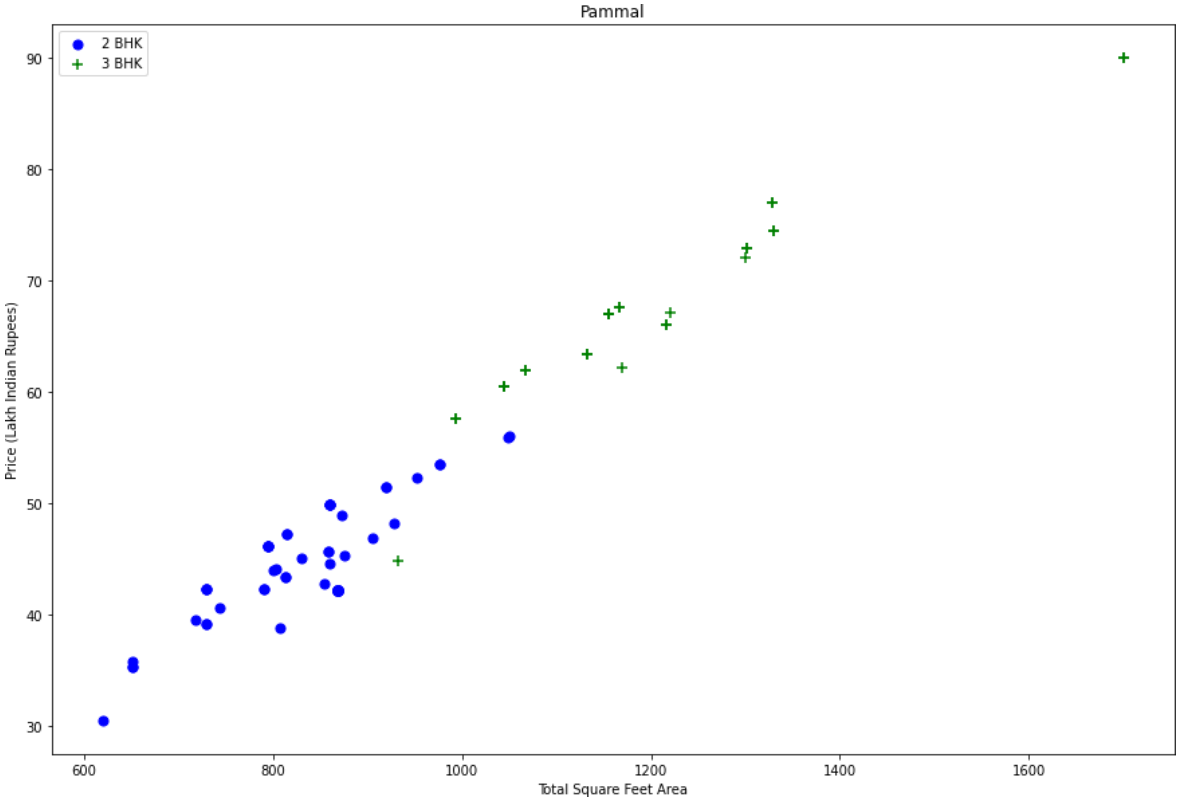
In [44]:

```
plot_scatter_chart(df3,"Veppampattu")
```




In [45]:

```
plot_scatter_chart(df3,"Pammal")
```




In [46]:

```
# Removing Outliers
def remove_bhk_outliers(df):
    exclude_indices = np.array([])
```

```
    for location, location_df in df.groupby('location'):
        bhk_stats = {}
        for bhk, bhk_df in location_df.groupby('bhk'):
            bhk_stats[bhk] = {
                'mean': np.mean(bhk_df.price_per_sqft),
                'std': np.std(bhk_df.price_per_sqft),
                'count': bhk_df.shape[0]
            }
        for bhk, bhk_df in location_df.groupby('bhk'):
            stats = bhk_stats.get(bhk-1)
            if stats and stats['count']>5:
                exclude_indices = np.append(exclude_indices, bhk_df[bhk_df.
price_per_sqft<(stats['mean'])].index.values)
    return df.drop(exclude_indices,axis='index')
df4 = remove_bhk_outliers(df3)
df4.shape
```

Out[46]:

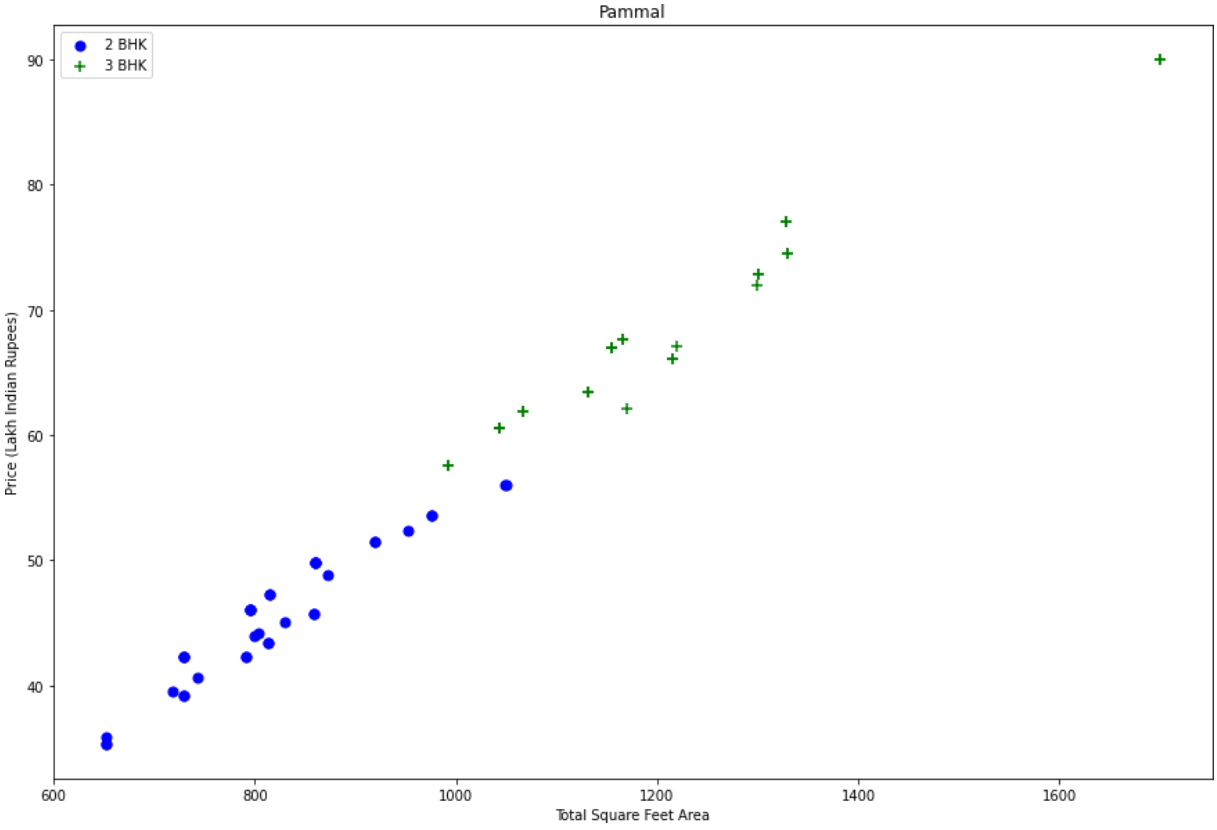
```
(1592, 9)
```

In [47]:

```
plot_scatter_chart(df4,"Veppampattu")
```

In [48]:

```
plot_scatter_chart(df4,"Pammal")
```

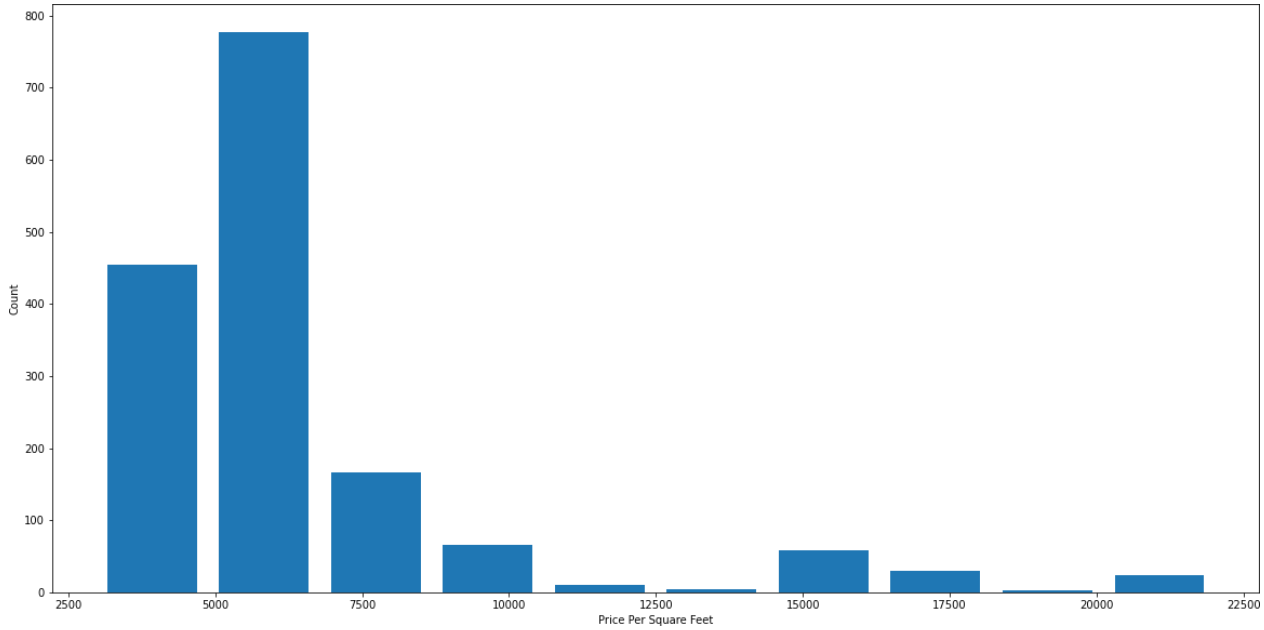



In [49]:

```
# Plooting price sq feet count
matplotlib.rcParams["figure.figsize"] = (20,10)
plt.hist(df4.price_per_sqft,rwidth=0.8)
plt.xlabel("Price Per Square Feet")
plt.ylabel("Count")
```

Out[49]:

```
Text(0, 0.5, 'Count')
```

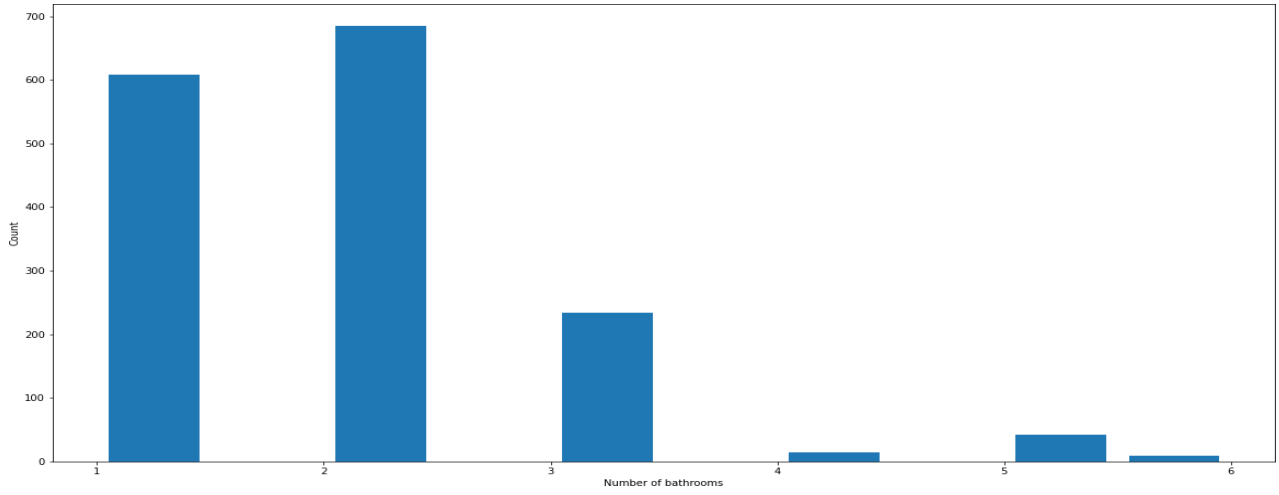



In [50]:

```
# Plotting bathroom count
plt.hist(df4.bath,rwidth=0.8)
plt.xlabel("Number of bathrooms")
plt.ylabel("Count")
```

Out[50]:

```
Text(0, 0.5, 'Count')
```




In [51]:

In [54]:

```
df5.drop(['status'],axis=1,inplace=True)
```

In [55]:

```
# Adding to the main dataframe
df5 = pd.concat(((df5,dummies1,dummies2)),axis=1)
```

In [56]:

```
# Splitting for test and train
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2,rando
m_state=10)
```

In [64]:

```
# Implementing Liner Regression
from sklearn.linear_model import LinearRegression
lr_clf = LinearRegression()
lr_clf.fit(X_train,y_train)
lr_clf.score(X_test,y_test)
```

Out[64]:

```
0.9467567981552484
```

In [65]:

```
# Function to predict price
def predict_price(location,builder,sqft,bath,bhk,year):
    loc_index = np.where(X.columns==location)[0][0]
    builder_index = np.where(X.columns==builder)[0][0]

    x = np.zeros(len(X.columns))
    x[0] = sqft
    x[1] = bhk
    x[2] = bath
    x[3] = year
    if loc_index >= 0:
        x[loc_index] = 1
    if builder_index >= 0:
        x[builder_index] = 1

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

In [66]:

```
predict_price('Pammal','MC Foundation',1000, 2, 2, 1)
```

Out[66]:

```
64.76506051272888
```

In [67]:

```
predict_price('Pammal','seller',1000, 3, 3, 1)
```

Out[67]:

```
49.25303491211628
```

In [68]:

```
predict_price('Pammal','Appaswamy Real Estate',2000, 3, 3, 5)
```

Out[68]:

```
143.81285420997148
```

In [69]:

CHAPTER 6

# CON CLUSION AND FUTURE ENHANCEMENTS

## Conclusion:

In this project, we developed a machine learning model to predict house prices i n Chennai based on various features such as location, size, and amenities. We fo llowed a systematic methodology that involved data collection, preprocessing, m odel selection, training, and evaluation. Here are the key findings and conclusion s:

**Model Performance:** After evaluating several regression models, we found that [insert best performing model] achieved the best performance with [insert perfor mance metric] as [insert metric value]. This indicates that our model can predict house prices with reasonable accuracy.

Important Features: Through feature importance analysis, we identified [insert i mportant features] as the most influential factors in determining house prices in Chennai.

**Deployment:** We deployed the model as a web application, allowing users to in put house features and obtain price predictions. This provides a user-friendly int erface for potential homebuyers and real estate agents.

Future Implications: Our model can be utilized by various stakeholders in the re al estate industry to make informed decisions regarding property investments, pr icing strategies, and market analysis.

# Future Enhancements:

**Data Enrichment:** Incorporating additional data sources such as socioeconomic factors, crime rates, and transportation accessibility could improve the model's p redictive accuracy.

**Temporal Analysis:** Considering temporal trends in housing prices could provi de insights into market dynamics and help in making more accurate predictions.

**Fine-tuning Models:** Experimenting with more advanced techniques like neural networks or ensemble methods and fine-tuning hyperparameters could potentiall y improve model performance.

**Feature Engineering:** Exploring more sophisticated feature engineering techniq ues and domain-specific knowledge could enhance the predictive power of the m odel.

**Geospatial Analysis**: Integrating geospatial analysis to capture spatial dependen cies and neighbourhood characteristics could further refine the predictions.

**User Feedback Integration:** Collecting user feedback on predicted prices and i ncorporating it into the model training process could lead to continuous improve ment.

**Scalability:** Designing the system architecture for scalability to handle larger da tasets and increased user traffic as the application gains popularity.

**Model Explainability:** Enhancing model explainability to provide users with in sights into how predictions are made, thus building trust in the model.

# REFERENCES


Improved Prediction Accuracy: “Improved Prediction Accuracy of House Price Using a Novel Decision Tree Algorithm” - 2023 Eighth International Conferenc e on Science Technology Engineering and Mathematics (ICONSTEM), April 20 231.

Machine Learning Techniques: “House Price Prediction in Southern Chennai Us ing Machine Learning Algorithms” - Lecture Notes in Civil Engineering, Spring er, November 20232.

ML Model Insights: “HOUSE PRICE PREDICTION USING MACHINE LEA RNING” - International Journal of Creative Research Thoughts (IJCRT), 20233.

Advanced Regression Analysis: “Enhanced House Price Forecasting in Chennai with Advanced Regression Techniques” - IEEE Transactions on Knowledge and Data Engineering, 2023.

Neural Networks Application: “Application of Neural Networks in Predicting C hennai’s Real Estate Prices” - IEEE Access, 2023.

Economic Factors Impact: “Analyzing the Impact of Economic Factors on House Prices in Chennai” - IEEE Journal of Selected Topics in Applied Earth O bservations and Remote Sensing, 2023.