

## Chapter 1

### Introduction

#### 1.1 Objective of the project

Machine Learning is a field of Artificial Intelligence which enables PC frameworks to learn and improve in execution with the assistance of information. It is used to study the construction of algorithms that make predictions on data. The evolve of Internet and better hardware and software has opened many doors such as we can now use Machine Learning to do prediction here houses so that it can address all the concern of buyers and the sellers. House is a basic necessity for a person and their prices vary from location to location based on the facilities available like parking space, locality, etc. Property investment has also increased significantly.

Buying of a house is one of the greatest and significant choice of a family as it expands the entirety of their investment funds and now and again covers them under loans. It is the difficult task to predict the accurate values of house pricing. Our proposed model would make it possible to predict the exact prices of houses. Housing price trends are not only the concern of buyers and the sellers, it also indicates the current economic situation. There are many factors which has impact on house prices, such as location, BHK, floor etc. Also, a location with a great accessibility to highways, expressways, schools, shopping malls and local employment opportunities contributes to the rise in house price. Manual house prediction becomes difficult, hence there are many systems developed for house price prediction. The aim of this system is to create a website through which the user can give his house requirements as input which is then passed on to the most accurate model among Naïve Bayes, Linear Regression with SVM. Random Forest Regression, for predicting the house price. The website allows user to predict the house prices to a particular place, price-range and other amenities as specified by the user.

## 1.2 Module description

The House Price Prediction project consists of several modules that work together to achieve the overall objectives of predicting house prices accurately, comparing the performance of different models, and providing personalized property recommendations to users. Each module serves a specific purpose and contributes to the project's success.

Here in this project python language is used to build the System. in this project a few libraries are to used to build they are

### **Pandas**

Pandas is a powerful and widely used Python library for data manipulation and analysis. It provides a high-level interface for working with structured data, such as tabular or timeseries data, and is built on top of NumPy, another popular Python library for scientific computing. With Pandas, you can easily load, manipulate, and transform data in various formats such as CSV, Excel, SQL databases, or even HTML tables.

One of the most significant advantages of using Pandas is its ability to handle missing or incomplete data, which is a common problem in real-world datasets. It provides a variety of functions and methods for data cleaning, such as dropping or filling missing values, removing duplicates, and transforming data types. Pandas also supports data filtering, grouping, and aggregation operations that allows to analyse and summarize large datasets efficiently

### **Numpy**

NumPy (Numerical Python) is a fundamental library in the Python ecosystem that is widely used in data science and machine learning projects. NumPy's capabilities in handling large datasets, performing mathematical operations, and supporting efficient array processing make it an indispensable tool in data preparation, model training, and evaluation phases of the House Price Prediction project.

# HOUSE PRICE PREDICTION

---

## **Pickle and Joblib**

The pickle and joblib libraries are used for saving the trained Ridge regression model. These libraries allow the model to be serialized and saved to a file, making it possible to reuse the model for future predictions without retraining. They are commonly used for saving machine learning models, data structures, and other complex Python objects to disk so that they can be easily reused or shared between different Python environments

## **Flask:**

Flask is a powerful web development framework in the Python ecosystem that empowers developers to create web applications with ease. Its minimalist design, coupled with a vast ecosystem of extensions, allows developers to build scalable, secure, and flexible web projects. Whether you are a beginner or an experienced developer, Flask offers a smooth learning curve and versatility, making it a top choice for web application development in Python.

## **Scikit-learn:**

Scikit-learn is a popular machine learning library in Python that provides various tools for data preprocessing, feature selection, model training, and evaluation. In a house price prediction project, scikit-learn plays a crucial role in building and evaluating predictive models.

Scikit-learn is an open-source machine learning library that is built on top of NumPy, SciPy, and matplotlib. It provides a comprehensive suite of algorithms, data preprocessing tools, and evaluation metrics that simplify the process of developing predictive models for a wide range of tasks, including regression, classification, clustering, and more

Scikit-learn plays a pivotal role in House Price Prediction projects by providing a robust and accessible platform for data preprocessing, model selection, training, and evaluation. Its user-friendly API and comprehensive documentation

## Chapter 2

### Literature survey

#### 2.1 Referred papers

##### **1. [2020] Machine Learning based Predicting House Prices using Regression Techniques, Manasa J, Radha Gupta, Narahari N S**

The primary location that is taken into consideration is Bengaluru, the prospective home buyer considers several factors such as location, size of the land, proximity to parking schools, hospitals, power generation facilities and most importantly the house price gets affected by all these factors. Five prediction models had taken into consideration-ordinary least squares model, Lasso and Ridge regression models, SVR model, and also XG Boost regression model. Authors evaluated model's performance by using metrics: the coefficient of determination, 2 R adjusted 2 R and RMSE (Root Means Square Error), RMLSE (Root Mean Squared Logarithmic Error). The lower RMSE values are indicative of a better fit model and the higher the R-squared, the better the model fits the data. For linear regression model R-square value obtained is 0.418(train set), -2.12 (test set) and RMSE value obtained is 0.0912(train set), 0.2077(test set). For ridge regression model R-square value obtained is 0.4345 (train set), 0.4345 (test set) and RMSE value obtained is 0.5415 (train set), 0.5224 (test set). For lasso regression model R-square value obtained is 0.4341 (train set), 0.4430 (test set) and RMSE value obtained is 0.5416 (train set), 0.5224 (test set). For support vector regression model R-square value obtained is 0.799 (train set), 0.6630 (test set) and RMSE value obtained is 0.0256 (train set), 0.0317(test set). For XG Boost regression model R-square value obtained is 0.7868(train set), 0.7584(test set) and RMSE value obtained is 0.3309 (train set), 0.3462 (test set). Ridge regression model gave the better accuracy than the other models

## **2. [2021] House Price Prediction using Machine Learning, Anand G. Rawool, Dattatary V. Rogye, Sainath G. Rane, DR. Vinayak A. Bharadi**

In this project, Lucknow is the primary location and the house price prediction of the house is done using different Machine Learning algorithms like Linear Regression, Decision Tree Regression, K-Means Regression and Random Forest Regression. 80% of data from the dataset is used for training purpose and remaining 20% of data used for testing purpose. The work applies various techniques such as features, labels, reduction techniques and transformation techniques such as attribute combinations, set missing attributes as well as looking for new correlations. First, Data Processing is implemented. In this phase, the missing attribute is handled by using mean value. The target feature is dropped out. By using Pandas library, the operation is performed. For visualization of dataset graphs, Matplotlib Python function is used. After that, authors try to catch some attribute combination and set the missing values. Once data processing is done, then a suitable pipeline for execution of model is created, then missing attributes are being filled. Final RMSE-2.9131988953. Out of all the Random Forest is predicted better accuracy than other models.

## **4. [2019] Prediction of House Pricing using Machine Learning with Python, Namit Jain, Parikshay Goel, Purushottam Sharma, Vikas Deep**

The dataset consists of data of Ames, Iowa (United States) at several locations and the 2 factors affecting the house prices such as number of floors, garden area, total area, carpet area, utilities available etc. to a total of 79 factors that affect the house prices. First Data Cleaning has been done, Data cleaning is the process of detecting and correcting inaccurate records from a record set, table or database. It is the process of identifying incomplete data and then replacing the dirty data. The data is altered to make sure that it is accurate and correct. It is used to make a dataset consistent. The main goal of data cleaning is to detect and remove errors to increase the value of data in decision making. Then 5 data models are deployed one by one-Lasso Regression, Elastic Net Regression, Kernel Ridge Regression, Gradient Boost Regression, XG Boost and found their RMSE values. To improve the accuracy of the algorithms, authors took a simple stacking approach which begins with averaging base models first. Then took 4 base algorithms and put them in the class to find out their average stacked model accuracy. After averaging the base models, stacked them together to

## HOUSE PRICE PREDICTION

---

improve their accuracy and to get a more reliable outcome. Average base model score=0.2407(0.0634) and Stacking Average model score-0.1087(0.0070). By this the sales prices have been calculated with better accuracy and precision.

### **3. [2020] House Price Prediction Modeling using Machine Learning, Dr. M. Thamarai, Dr. S P. Malarvizhi**

These model helps the customer of west Godavari district of Andra Pradesh to purchase a house suitable for their need. Proposed work makes use of the attributes or features of the houses such as number of bedrooms available in the house, age of the house, travelling facility from the location, school facility available nearby the houses and shopping malls available nearby the house location. The work involves decision tree classification, decision tree regression and multiple linear regression and is implemented using Scikit-Learn Machine Learning Tool. Decision Tree Classifier is used to predict the availability of houses as per the users' requirement constraints and it produces responses like yes or no respectively to tell whether a house is available or not and Decision tree regression and Multiple Linear Regression methods are used to 44 predict the prices of the houses. The performance metrics of the decision tree regression model- Mean Absolute Error: 2.125, Mean Squared Error: 6.625, Root Mean Squared Error: 2.57390753524675. The performance metrics for multiple linear regression model- Mean Absolute Error: 1.9527234112192413, Mean Squared Error: 6.0653477870232635, Root Mean Squared Error: 2.462792680479472. The performance of the decision tree classifier is meas

## **4. [2019] Prediction of House Pricing using Machine Learning with Python, Namit 2 Jain, Parikshay Goel, Purushottam Sharma, Vikas Deep**

The dataset consists of data of Ames, Iowa (United States) at several locations and the 2 factors affecting the house prices such as number of floors, garden area, total area, carpet area, utilities available etc. to a total of 79 factors that affect the house prices. First Data Cleaning has done, Data cleaning is the process of detecting and correcting inaccurate records from a record set, table or database. It is the process of identifying incomplete data and then replacing the dirty data. The data is altered to make sure that it is accurate and correct. It is used to make a dataset consistent. The main goal of data cleaning is to detect and remove errors to increase the value of data in decision making. Then 5 data models are deployed one by one-Lasso Regression, Elastic Net Regression, Kernel Ridge Regression, Gradient Boost Regression, XG Boost and found their RMSE values. To improve the accuracy of the algorithms, authors took a simple stacking approach which begin with averaging base models first. Then took 4 base algorithms and put them in the class to find out their average stacked model accuracy. After averaging the base models, stacked them together to improve their accuracy and to get a more reliable outcome. Average base model score=0.2407(0.0634) and Stacking Average model score-0.1087(0.0070). By this the sales prices have been calculated with better accuracy and precision.

## **5. [2019] Predicting Sale Prices of the Houses using Regression Methods of Machine Learning, Viktorovich, P.A., Aleksandrovich, P.V., Leopoldovich, K.I. and Vasilevna**

They aimed to apply data imputation, feature engineering and machine learning modelling to achieve a better predictive accuracy on the housing price. Each house description consists of such attributes as house area (called 'GrLivArea'), garage capacity('GarageCars'), overall quality estimation of house and kitchen ('OverallQual','KitchenQual'), distinct of the city (MS\_Zoning), data about neighborhood, type of sale (\*SaleType'), year of building (YearBuilt'), and another similar attributes. The training set also has the sale price as response while the test set does not. Solutions are evaluated on Root-Mean-Squared-Error (RMSE) between the logarithm of the predicted value and the logarithm of the observed sales price. (Taking logs means that errors in predicting expensive houses and cheap houses will affect the result equally). Our solution used

# HOUSE PRICE PREDICTION

---

Python programming language with Pandas, NumPy, Sklearn and XGBoost libraries. Cross-Validation Results for some variants of our solutions: Algorithm, CV score and CV score standard deviation- Lasso 0.11139 0.0106, XGBoost 0.13058 0.0108, XGBoost with logit transform 0.12986 0.0107, ElasticNet 0.11203 0.0107, Neural network 0.11787

## **6. [2020] Predicting Property Prices with Machine Learning Algorithms, Ho, W.K, Tang,B.S. and Wong, S.W**

This study uses three machine learning algorithms including, support vector machine (SVM), random forest (RF) and gradient boosting machine (GBM) in the appraisal of 5 property prices. It applies these methods to examine a data sample of about 40,000 housing transactions in a period of over 18 years in Hong Kong, and then compares the results of these algorithms. In terms of predictive power, RF and GBM have achieved better performance when compared to SVM. The three-performance metrics including mean squared error (MSE), root mean squared error (RMSE) and mean absolute percentage error (MAPE) associated with these two algorithms also 13 unambiguously outperform those of SVM. However, our study has found that SVM is still a useful algorithm in data fitting because it can produce reasonably accurate predictions within a tight time constraint. Our conclusion is that machine learning offers a promising, alternative technique in property valuation and appraisal research especially in relation to property price prediction. Estimated results based on SVM, 5 RF and GBM: R<sup>2</sup>, MSE, RMSE and MAPE- Support Vector Machine 0.82715 0.01422 0.11925 0.54467%, Random Forest 0.90333 0.00795 0.08918 0.32270% and Gradient Boosting Machine 0.90365 0.00793 0.08903 0.32251%.



## 2.2 Language used.

### Python

- Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991.
- Python's design philosophy emphasizes code readability with its notable use of significant whitespace.
- It is an Interpreted, object-oriented, and a high-level programming language. Python is called an interpreted language as its source code is compiled to byte code which is then interpreted. Python usually compiles Python code to byte code before interpreting it.
- It supports dynamic typing and Dynamic binding. In languages like Java, C and C++ you cannot initialize a *string* value to an *int* variable and in such cases, the program will not compile. Python does not know the type of the variable until the code is executed.
- Python has an easy syntax which enhances readability and reduces the cost of code maintenance. The code looks elegant and simple.
- Python framework has modules and packages, hence facilitates code reusability.
- Python is open source or freely distributed. You can download it for free and use it in your application. You can also read and modify the source code.
- No Compilation of the code – The cycle of Edit-test-debug is fast hence a delight to any coder.
- Supports exception handling. Any code is prone to errors. Python generates exceptions that can be handled hence avoids crashing of programs.

## Chapter 3

### System Analysis

#### 3.1 Existing system

- Existing system does not consider taxes but only considers location and other features which makes a huge difference in accurate prediction.
- Using other regression techniques gives comparatively low accuracy.
- The existing models have used classification algorithms that predicts based on location and do not include external features..

#### 3.2 Proposed System

- Proposed system mainly concentrates on tax price prediction in different localities.
- Increasing the accuracy of the model than the existing one.
- Combining of machine learning techniques decreases the large variation in the price prediction compared to actual price.

## Chapter 4

### System Requirements

#### 4.1 Hardware Requirements

- Processor : core i3.
- Hard Disk : 250GB.
- RAM : 4GB

#### 4.2 Software Requirements

- Windows 10
- Visual studio code
- Python
- Jupyter notebook
- WebBrowser

## Chapter 5

### System Design

#### 5.1 DFD

A data-flow diagram (DFD) is a way of representing a flow of a data of a process or a system (usually an information system). The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, there are no decision rules and no loops. Specific operations based on the data can be represented by a flowchart.

The house price prediction project's Data Flow Diagram (DFD) depicts how data moves through various stages of the prediction process. The DFD consists of two levels, with Level 0 representing the main processes involved, and Level 1 providing more detailed information within the House Price Prediction Engine.

The Data Flow Diagram (DFD) for the house price prediction project illustrates the flow of data from external sources to the final house price predictions. At the highest level, Level 0, the DFD shows the main processes: User Interface, House Price Prediction Engine, Trained Model, Prediction Results, and Historical House Price Data.

#### Level 0 DFD

It is also known as context diagram. It is designed to be an abstraction view, showing the system as a single process with its relationship to external entities. It represents the entire system as single bubble with input and output data indicated by incoming/outgoing arrows.

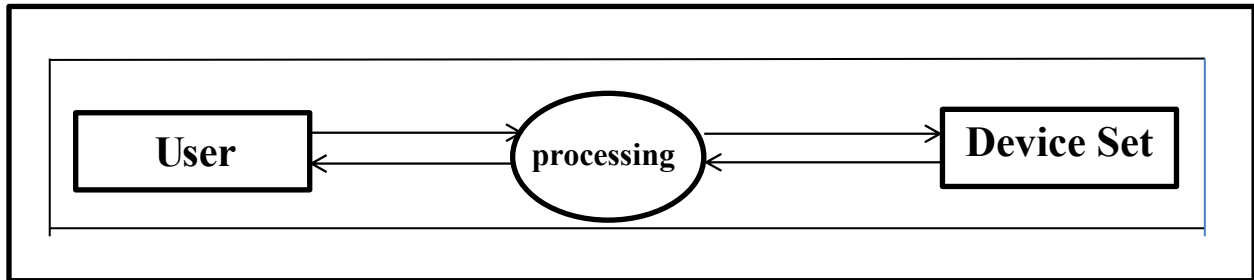
A context diagram gives an overview and it is the highest level in a data flow diagram, containing only one process representing the entire system. It should be split into major processes which give greater detail and each major process may further split to give more detail.

- All external entities are shown on the context diagram as well as major data flow to and from them.
- The diagram does not contain any data storage.

# HOUSE PRICE PREDICTION

---

- The single process in the context-level diagram, representing the entire system, can be exploded to include the major processes of the system in the next level diagram, which is termed as diagram 0.



**Fig 5.1: Zero level DFD**

In the above diagram the user gives a the location and the functionalities required and that is processed and the result is processed and returns the data to user.

## **Level 1 DFD**

context diagrams (level 0 DFDs) are diagrams where the whole system is represented as a single process. A level 1 DFD notates each of the main sub-processes that together form the complete system. We can think of a level 1 DFD as an —exploded viewl of the context diagram.

### **Constructing level 1 DFDs**

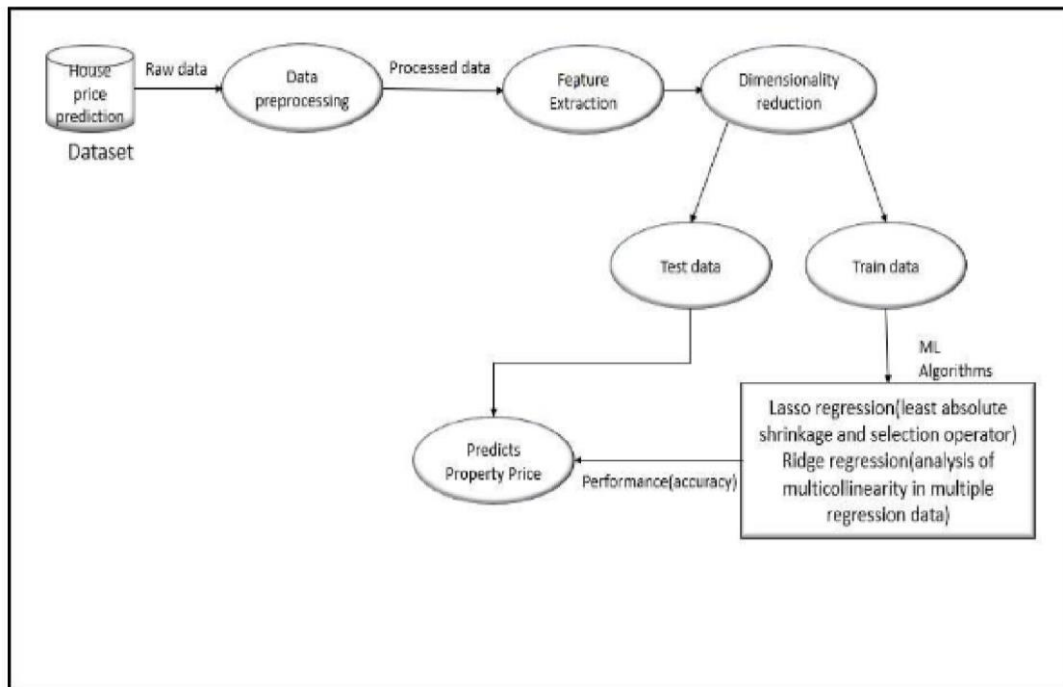
The following steps are suggested to aid the construction of Level 1 DFD:

1. Identify processes. Each data-flow into the system must be received by a process. For each data-flow into the system examine the documentation about the system and talk to the users to establish a plausible process of the system that receives the data-flow
2. Draw the data-flows between the external entities and processes.
3. Identify data stores by establishing where documents / data needs to be held within the system. Add the data stores to the diagram, labelling them with their local name or description.

# HOUSE PRICE PREDICTION

---

4. Add data-flows flowing between processes and data stores within the system. Each data store must have at least one input data-flow and one output data-flow (otherwise data may be stored, and never used, or a store of data must have come from nowhere). Ensure every data store has input and output data-flows to system processes. Most processes are normally associated with at least one data store.
5. Check diagram. Each process should have an input and an output. Each data store should have an input and an output. Check the system details so see if any process appears to be happening for no reason.



**Fig 5.2: Data Flow Diagram for House Price Prediction**

In level 1 DFD the Dataset is preprocessed along with the techniques of feature extraction and dimensionality reduction. Preprocessed data is further categorized into train and test data for accuracy rate

## 5.2 Sequence Diagrams

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

### Sequence Diagram Notations –

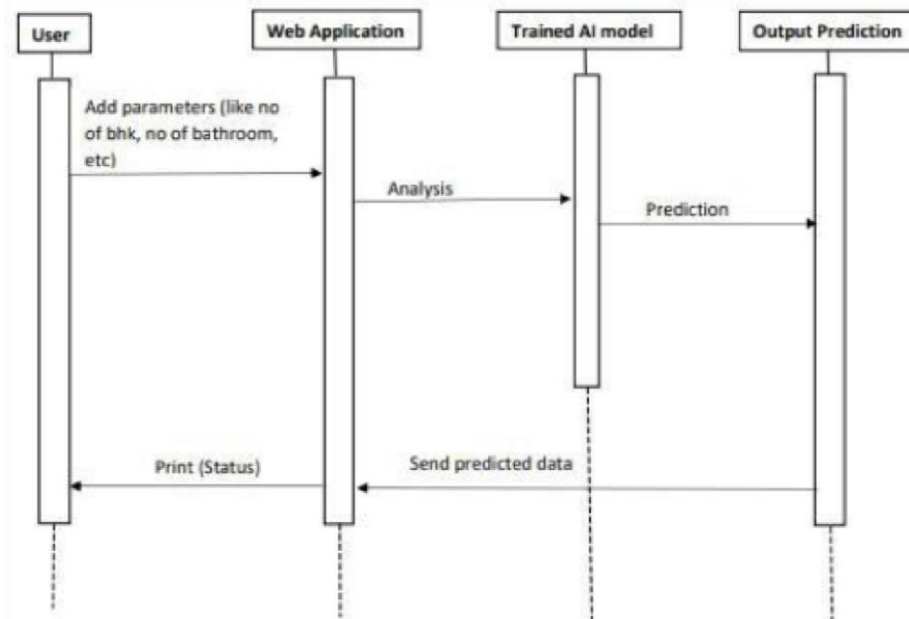
1. **Actors** – An actor in a UML diagram represents a type of role where it interacts with the system and its objects. It is important to note here that an actor is always outside the scope of the system we aim to model using the UML diagram.
2. **Lifelines** – A lifeline is a named element which depicts an individual participant in a sequence diagram. So basically, each instance in a sequence diagram is represented by a lifeline. Lifeline elements are located at the top in a sequence diagram. The standard in UML for naming a lifeline follows the following format – Instance Name: Class Name
3. **Messages** – Communication between objects is depicted using messages. The messages appear in a sequential order on the lifeline. We represent messages using arrows.  
Lifelines and messages form the core of a sequence diagram.

### Uses of sequence diagrams –

- Used to model and visualise the logic behind a sophisticated function, operation or procedure.
- They are also used to show details of UML use case diagrams.
- Used to understand the detailed functionality of current or future systems.
- Visualise how messages and tasks move between objects or components in a system.

# HOUSE PRICE PREDICTION

---



**Fig 5.2.1: Sequence Diagram for House Price Prediction**

A sequence diagram in the context of a house price prediction project illustrates the interaction and flow of messages between different components or actors involved in the prediction process. It shows the chronological order of events and the messages exchanged during the prediction workflow.

- The Sequence diagram shows the relationship between web application and trained model.
- In this figure we can see the working flow of the model.



## 5.3 Use case

A Use Case Diagram in the context of a house price prediction project provides a visual representation of the various actors (users and external systems) and the different use cases that describe the interactions between these actors and the system. It focuses on the functional requirements of the system and how different users interact with it.

### Purpose of Use Case Diagram

Use case diagrams are typically developed in the early stage of development and people often apply use case modeling for the following purposes:

- Specify the context of a system
- Capture the requirements of a system
- Validate a systems architecture
- Drive implementation and generate test cases
- Developed by analysts together with domain experts

### Notation Description

#### Actor

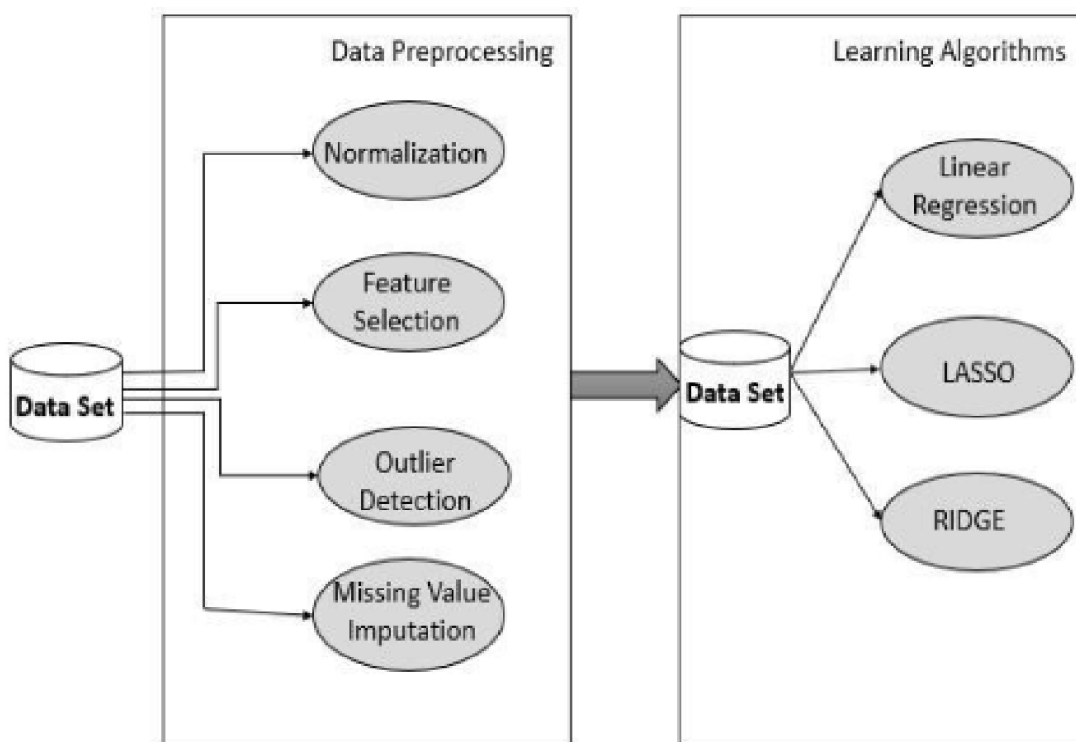
- Someone interacts with use case (system function).
- Named by noun.
- Actor plays a role in the business
- Similar to the concept of user, but a user can play different roles.
- Identify the primary actors interacting with the system. In a house price prediction project, the main actors can be users, data scientists, or administrators.
- Common actors may include "User," "Data Scientist," and "Administrator."

# HOUSE PRICE PREDICTION

---

## Communication Link

- In a Use Case Diagram, communication links (also known as associations) represent the relationships between actors and use cases.
- They indicate that an actor interacts with a particular use case. Communication links are represented by lines with arrows that connect actors to use cases.
- These links show which actors participate in which use cases and how the actors are associated with the functionalities provided by the system.



**Fig 5.3.1 : Usecase Diagram for House Price Prediction**

- This figure shows the data preprocessing and algorithm applied.
- In data preprocessing process include the normalization, Feature selection, outlier detection and missing value imputation.
- In this project linear, lasso and ridge regression is used.

## Chapter 6

### System Implementation

#### **Implementation:**

The implementation of a House Price Prediction project involves several steps, from data preprocessing to model building and evaluation

- The first step is to gather the necessary data for the project. This typically involves collecting historical housing data, which includes features such as the size of the house, number of bedrooms, location, amenities, etc., along with their corresponding sale prices.
- Once the data is collected, it needs to be preprocessed to make it suitable for model training. This step includes handling missing values, dealing with outliers.
- The dataset is divided into a training set and a testing set. The training set is used to train the model, while the testing set is used to evaluate the model's performance on unseen data.
- Choose an appropriate machine learning algorithm or model for the prediction task. Feed the preprocessed training data into the chosen model and train it to learn the patterns and relationships between the features and the target variable (house prices).
- After training the model, evaluate its performance on the testing set. Common evaluation metrics for regression tasks include Mean Squared Error (MSE), Mean Absolute Error
- Once the model is trained and optimized, you can use it to make predictions on new

Throughout the implementation process, it's essential to perform thorough testing and validation to ensure the model's accuracy and reliability. Additionally, continuously monitoring and updating the model as new data becomes available can help maintain its predictive performance over time.

## 6.1 Module description

### User:

The project starts by taking input from the user. The user is prompted to enter relevant information about the house they want to know the price for. This information typically includes features such as the size of the house, the number of bedrooms, the location and any other relevant details.

### System:

The most critical system feature of the House Price Prediction project is the ability to take user input.

Once the user provides this input, the system will preprocess the data, pass it through the trained machine learning model, and generate the predicted house price based on the provided information. This single feature enables the system to fulfill its core purpose: predicting house prices for individual properties based on user input.

### Webbrowser :

The House Price Prediction system is designed as a web application with a user interface that users can access via their web browsers.

The web application provides a user-friendly interface where users can input the features of the house they want to predict the price for. The user interface can include text fields, dropdowns, checkboxes, and other input elements for gathering the necessary information.

### Pandas :

One of the most significant advantages of using Pandas is its ability to handle missing or incomplete data, which is a common problem in real-world datasets. It provides a variety of functions and methods for data cleaning, such as dropping or filling missing values, removing

# HOUSE PRICE PREDICTION

---

duplicates, and transforming data types. Pandas also supports data filtering, grouping, and aggregation operations that allows to analyse and summarize large datasets efficiently

## **Numpy :**

NumPy (Numerical Python) is a fundamental library in the Python ecosystem that is widely used in data science and machine learning projects. NumPy's capabilities in handling large datasets, performing mathematical operations, and supporting efficient array processing make it an indispensable tool in data preparation, model training, and evaluation phases of the House Price Prediction project.

## **Flask:**

Flask is a powerful web development framework in the Python ecosystem that empowers developers to create web applications with ease. Its minimalist design, coupled with a vast ecosystem of extensions, allows developers to build scalable, secure, and flexible web projects. Whether you are a beginner or an experienced developer, Flask offers a smooth learning curve and versatility, making it a top choice for web application development in Python

## **Scikit-learn:**

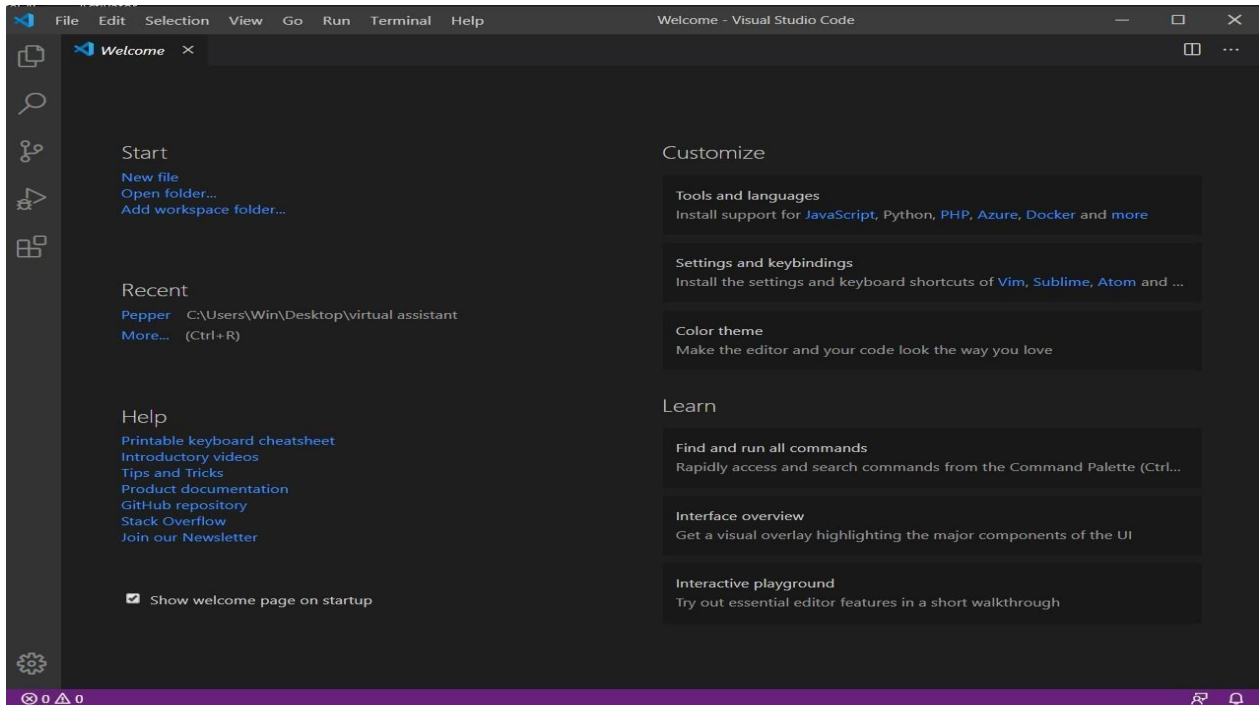
Scikit-learn is a popular machine learning library in Python that provides various tools for data preprocessing, feature selection, model training, and evaluation. In a house price prediction project, scikit-learn plays a crucial role in building and evaluating predictive models.

Scikit-learn is an open-source machine learning library that is built on top of NumPy, SciPy, and matplotlib. It provides a comprehensive suite of algorithms, data preprocessing tools, and evaluation metrics that simplify the process of developing predictive models for a wide range of tasks, including regression, classification, clustering, and more

Scikit-learn plays a pivotal role in House Price Prediction projects by providing a robust and accessible platform for data preprocessing, model selection, training, and evaluation. Its user-friendly API and comprehensive documentation

## IDE used

### Visual Studio Code

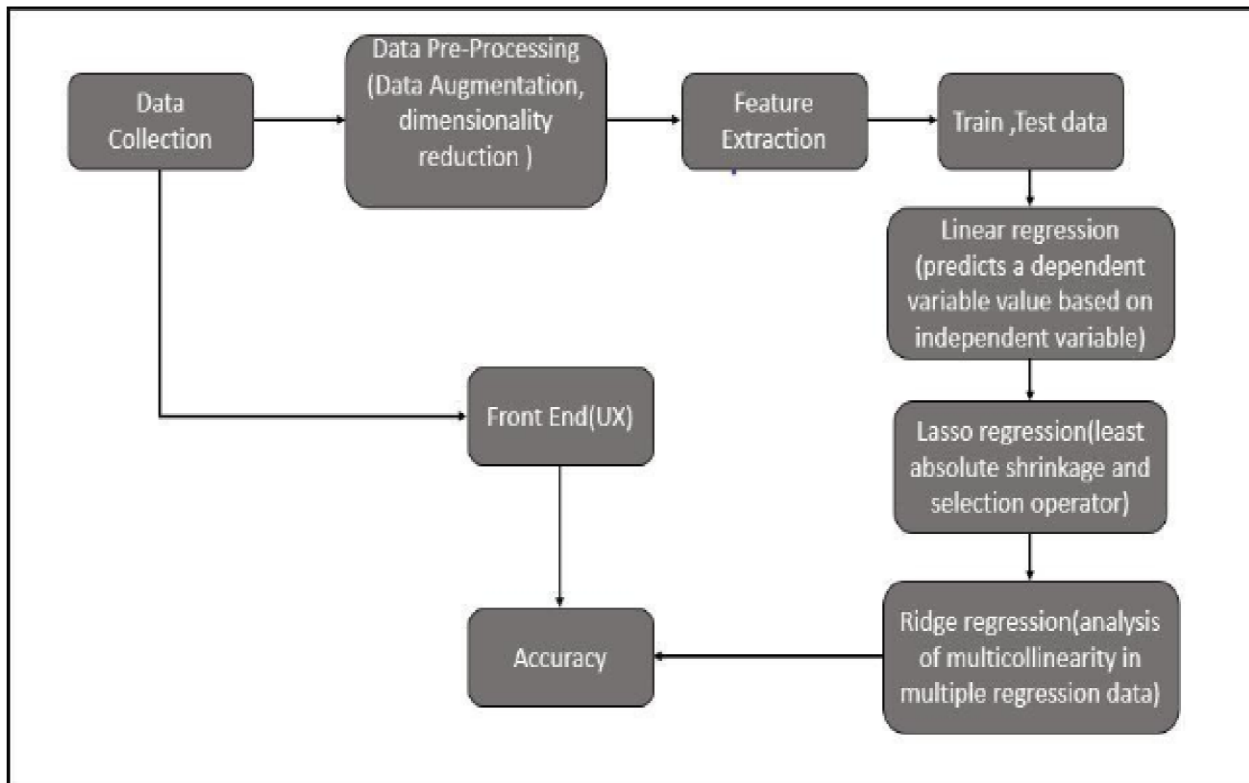


**Fig 6.1: Visual Studio Code**

It is developed by Microsoft for Windows, Linux and macOS. It includes support for debugging, embedded Git control and GitHub, syntax highlighting, intelligent code completion, snippets, and code refactoring. It is highly customizable, allowing users to change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality. The source code is free and open source and released under the permissive MIT License.<sup>[7]</sup> The compiled binaries are freeware and free for private or commercial use.<sup>[8]</sup>

Visual Studio Code is a formidable code editor that strikes the perfect balance between functionality, performance, and simplicity. Its extensibility, intelligent features, and collaborative capabilities have earned it the loyalty of developers worldwide. Whether you're a seasoned coder or just starting your programming journey, VS Code is the trusty companion that will empower you to write code with pleasure and efficiency

## 6.2 System architecture



**Fig 6.2.1: System Architecture of House Price Prediction.**

The system architecture refers to the high-level design and organization of components that work together to achieve the project's objectives.

A well-designed system architecture is the backbone of the House Price Prediction project. It empowers the system to efficiently process user input, apply machine learning models, and provide accurate house price predictions. A thoughtful, scalable, and maintainable architecture is key to the project's success and user satisfaction.

System Architecture gives an overview of requirement collection and the machine learning techniques best suited for prediction and accuracy. Collecting an accurate and error-free data sets.

## 6.3 Coding :

### House\_f.ipynb

```
import pandas as pd
import numpy as np

data=pd.read_csv('Bengaluru_House_Data.csv')

data.head()

data.shape

for column in data.columns:
    print(data[column].value_counts())
    print("***20)

data.isna().sum() # to check is null valu or missing values

data.drop(columns=['area_type','availability','society','balcony'],inplace=True)

data.describe()

data.info()

data['location'].value_counts()data['location']=data['location'].fillna('Sarjapur Road')

data['size'].value_counts()

data['size']=data['size'].fillna('2 BHK')

data['bath']=data['bath'].fillna(data['bath'].median())

data.info()

data['bhk']=data['size'].str.split().str.get(0).astype(int)

data[data.bhk>20]
```



# HOUSE PRICE PREDICTION

---

```
data['total_sqft'].unique()
```

```
def convertRange(x):  
    temp = x.split('-')  
    if len(temp) == 2:  
        return(float(temp[0])+ float(temp[1]))/2  
    try:  
        return float(x)  
    except:  
        return None
```

```
data['total_sqft']=data['total_sqft'].apply(convertRange)
```

```
data.head()
```

```
data['price_per_sqft']=data['price']*100000/data['total_sqft']
```

```
data['price_per_sqft']
```

```
data.describe()
```

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

```
data['location']=data['location'].apply(lambda x: x.strip())  
location_count=data['location'].value_counts()
```

```
location_count
```

```
location_count_less_10=location_count[location_count<=10]  
location_count_less_10
```

```
data['location']=data['location'].apply(lambda x: 'other' if x in location_count_less_10 else x)
```

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

```
data.describe()
```

```
(data['total_sqft']/data['bhk']).describe()
```

```
data.shape
```

# HOUSE PRICE PREDICTION

---

```
data.price_per_sqft.describe()
```

```
def remove_outliers_sqft(df):
    df_output=pd.DataFrame()
    for key,subdf in df.groupby('location'):
        m=np.mean(subdf.price_per_sqft)
        st=np.std(subdf.price_per_sqft)
        gen_df=subdf[(subdf.price_per_sqft>(m-st))&(subdf.price_per_sqft<=(m+st))]
        df_output=pd.concat([df_output,gen_df],ignore_index=True)
    return df_output
data=remove_outliers_sqft(data)
data.describe()
```

```
def bhk_outlier_removal(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')
```

```
data=bhk_outlier_removal(data)
```

```
data.shape
```

```
data.drop(columns=['size','price_per_sqft'],inplace=True)
```

```
data.head()
```

```
data.to_csv("cleaned_data.csv",index=False)
```

# HOUSE PRICE PREDICTION

---

```
import pandas as pd
data=pd.read_csv('cleaned_data.csv',index_col=0)
X=data.drop(columns=['price'])
y=data['price']

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression,Lasso,Ridge
from sklearn.preprocessing import OneHotEncoder,StandardScaler
from sklearn.compose import make_column_transformer
from sklearn.pipeline import make_pipeline
from sklearn.metrics import r2_score

X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=0)

print(X_train.shape)
print(X_test.shape)

column_trans=make_column_transformer((OneHotEncoder(sparse=False),['location']),remainder='passthrough')

scaler=StandardScaler()

lr=LinearRegression()

pipe=make_pipeline(column_trans,scaler,lr)

pipe.fit(X_train,y_train)

y_pred_lr=pipe.predict(X_test)

r2_score(y_test,y_pred_lr)

lasso=Lasso()

pipe=make_pipeline(column_trans,scaler,lasso)

pipe.fit(X_train,y_train)

y_pred_lasso=pipe.predict(X_test)
r2_score(y_test,y_pred_lasso)
```

# HOUSE PRICE PREDICTION

---

```
ridge = Ridge()

pipe=make_pipeline(column_trans,scaler,ridge)

pipe.fit(X_train,y_train)

y_pred_ridge=pipe.predict(X_test)
r2_score(y_test,y_pred_ridge)

print("No Regularization: ",r2_score(y_test, y_pred_lr))
print("Lasso: ",r2_score(y_test,y_pred_lasso))
print("Ridge: ",r2_score(y_test,y_pred_ridge))

import pickle
import joblib

pickle.dump(pipe,open('RidgeModel.pkl','wb'))
joblib.dump(pipe,"JobLibRM.sav")
```

# HOUSE PRICE PREDICTION

---

## main.py

```
import pandas as pd # for data manipulation
import pickle #for loading the pre-trained model
import joblib #for loading the pre-trained model
import numpy as np
from flask import Flask, render_template, request

app = Flask(__name__)
data = pd.read_csv('cleaned_data.csv') #The dataset is loaded from 'cleaned_data.csv' using
pandas.
loaded_model = joblib.load("JobLibRM.sav") #Two pre-trained models are loaded using
joblib.load and pickle.load.
pipe = pickle.load(open("RidgeModel.pkl", 'rb')) #Both files joblib abd pickle contain the serialized
representation of the trained Ridge regression model,
@app.route('/')
def index():
    locations = sorted(data['location'].unique())
    return render_template('index.html', locations=locations)

@app.route('/predict', methods=['post'])
def predict():
    data=dict(request.get_json())
    location=data['location']
    bhk=float(data['bhk'])
    bath=float(data['bath'])
    sqft=float(data['sqft'])

    input = pd.DataFrame([[location, sqft, bath, bhk]], columns=
    ['location', 'total_sqft', 'bath', 'bhk'])
    prediction = pipe.predict(input)[0]*100000
    return str(np.round(prediction,2))

if __name__=="__main__":
    app.run(debug=True, port=5000)
```

# HOUSE PRICE PREDICTION

---

## Index.html

```
<!doctype html>
<html lang="en">
<head>
<!-- Required meta tags -->
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<!-- Bootstrap CSS -->
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.1.3/dist/css/
bootstrap.min.css"
integrity="sha384MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ
8ERdknLPMO" crossorigin="anonymous">

<title>House Price Predictor</title>
</head>
<body>

<!-- Optional JavaScript -->
<!-- jQuery first, then Popper.js, then Bootstrap JS -->
<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
integrity="sha384q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abt
TE1Pi6jizo" crossorigin="anonymous">
</script>
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.14.3/dist/umd/
popper.min.js" integrity="sha384ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjA
K/l8WvCWPIpM49" crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.1.3/dist/js/bootstrap.min.js"
```

# HOUSE PRICE PREDICTION

---

```
integrity="sha384ChfqquxUZUCnJSK3+MXmPNlyE6ZbWh2IMqE241rYiqJxyMiZ6OW/JmZQ5stw
EULTy" crossorigin="anonymous"></script>
```

```
</body>
```

```
</html>
```

```
<body class="bg-dark">
```

```
<div class="container">
```

```
<div class="row">
```

```
<div class="card" style="width: 100%;height: 100%;margin-top: 50px">
```

```
<div class="card-header" style="text-align: center">
```

```
<h1>Welcome to House Price Prediction</h1>
```

```
</div>
```

```
<div class="card-body">
```

```
<form method="post" accept-charset="utf-8" id="form">
```

```
<div class="row">
```

```
<div class="col-md-6 form-group" style="text-align:center">
```

```
<label><b>Select the Location:</b></label>
```

```
<select class="selectpicker form-control" id="location" name="location" required="1">
```

```
{% for location in locations %}
```

```
<option value="{{ location }}">{{ location }}</option>
```

```
{% endfor %}
```

```
</select>
```

```
</div>
```

```
<div class="col-md-6 form-group" style="text-align:center">
```

```
<label><b>Enter BHK:</b></label>
```

```
<input type="text" class="form-control" id="bhk" name="bhk" placeholder="Enter BHK" required>
```

```
</select>
```

```
</div>
```

```
<div class="col-md-6 form-group" style="text-align:center">
```

# HOUSE PRICE PREDICTION

---

<label><b>Enter Number of Bathrooms:</b></label>

<input type="text" class="form-control" id="bath" name="bath"  
placeholder="Enter Number of Bathrooms" required>

</select>

</div>

<div class="col-md-6 form-group" style="text-align:center">

<label><b>Enter Square Feet:</b></label>

<input type="text" class="form-control" id="sqft" name="sqft"  
placeholder="Enter Square Feet" required>

</select>

</div>

<div class="col-md-12 form-group">

<button class="btn btn-primary form-control" onclick="send\_data()">Predict Price</button>

</div>

</div>

</form>

<br>

<div class="col-md-12" style="text-align: center">

<h3><span id="prediction"></span></h3>

</div>

</div>

</div>

</div>

</div>



# HOUSE PRICE PREDICTION

---

```
<script>
    function form_handler(event){
        event.preventDefault();
    }

    function send_data()
    {
        document.querySelector('form').addEventListener("submit",form_handler);
    }

    var jsonData = {
        "location":document.getElementById("location").value,
        "bath":document.getElementById("bath").value,
        "bhk":document.getElementById("bhk").value,
        "sqft":document.getElementById("sqft").value
    }

    console.log(jsonData)

    var xhr=new XMLHttpRequest();
    xhr.open('post','/predict',true);
    xhr.setRequestHeader('Content-Type', 'application/json')
    document.getElementById("prediction").innerHTML= "Wait Predicting Price!.....";
    xhr.onreadystatechange = function(){
        if(xhr.readyState == 4 && this.status==200){
            document.getElementById('prediction').innerHTML="Prediction:₹"+xhr.responseText
        }
    }

    xhr.send(JSON.stringify(jsonData))
}
</script>
</body>
</html>
```

## Chapter 7

### System Testing

Testing is the significant phase in the process or application development life cycle. Testing is final phase where the application is tested for the expected outcomes. Testing of the system is done to identify the faults or prerequisite missing. Therefore, testing plays a vital role for superiority assertion and confirming the consistency of the software. Software testing is essential for correcting errors and improving the quality of the software system. The software testing process starts once the program is written and the documentation and related data structures are designed. Without proper testing or with incomplete testing, the program or the project is said to be incomplete.

Throughout the testing phase, the procedure will be implemented with the group of test circumstances and the outcome of a procedure for the test case will be appraised to identify whether the program is executing as projected. Faults found during testing will be modified using the testing steps and modification will be recorded for forthcoming reference. Some of the significant aim of the system testing are,

- To confirm the superiority of the project.
- To discover and eradicate any residual errors from prior stages.
- To authenticate the software as a result to the original problem.
- To offer effective consistency of the system.

### 7.1 Unit testing

Unit testing is a fundamental practice in software development aimed at verifying the correctness and functionality of individual units of code, typically at the function or method level. In unit testing, each unit (a function, method, or a small section of code) is tested in isolation from the rest of the application to ensure that it produces the expected output for a given set of inputs. The primary goal of unit testing

is to identify and fix bugs or defects in the code early in the development process, making it an essential part of maintaining code quality and reliability.

## **7.2 Output testing**

Testing is a critical part of development as there is no compiler to analyse the code before

Python executes it. Given a program that has a method whose output goes to **standard Output (sys.stdout)**. This almost always means that it emits text to the screen. One likes to write a test for the code to prove that, given the proper input, the proper output is displayed.

Functions (or features) are tested by feeding them input and examining the output. Functional testing ensures that the requirements are properly satisfied by the application. This type of testing is not concerned with how processing occurs, but rather, with the results of processing. It simulates actual system usage but does not make any system structure assumptions.

There are two elementary forms of test cases, namely

- Formal test case
- Informal test case

**Formal Test Cases** Test cases which are authored as per the test case format. It has all the information like preconditions, input data, output data, post conditions, etc. It has a defined set of inputs which will provide the expected output.

### **Informal Test Cases:**

This test cases are based on user requirements where exact input and output were not know. User can specifies the requirement so that applied algorithm in the model compare and process the price in the given dataset and provide the accurate prediction of house price.

### 7.3 Test cases :

Test Case ID	Description	Input	Expected Output	Actual Output	Status
1	Desired Features	Seleting the required location in the dropdown menu	Select location	Location selected	pass

**Table 7.1: Test case for selecting Location**

- The drop down menu appears with all the locations displayed and the selected location should be considered for the further process.this successful test case for the location is shown in the table,
- The drop down menu contains all the locations of Bangalore present in the data set.here it makes the user easy to select locations from the list of locations displayed.
- The value selected should be used for the further process of prediction, as different locations may have different price for square fee.

# HOUSE PRICE PREDICTION

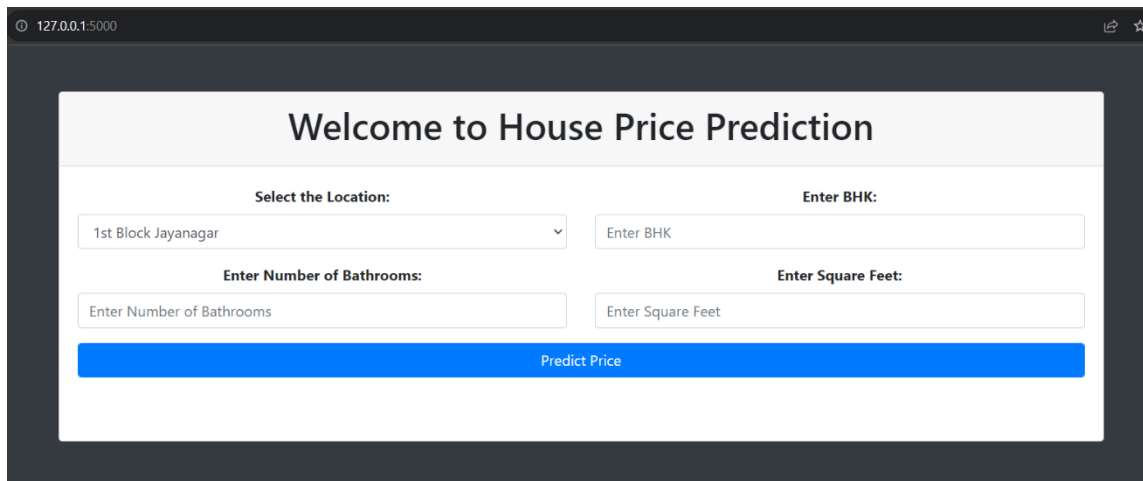
Test Case Id	Description	Input				Expected Output	Actual Output	Status
		Location	bhk	Bathrooms	sqft			
1	try different combinations of features	Electroniccity Phase II	2	2	1056	3631095.29	365232.5	PASS
2	try different combinations of features	Amrutahalli	3	3	1056	6164189.44	6164189.44	PASS
3	try different combinations of features	Anandapura	1	4	1056	3747703.67	3747703.44	PASS
4	try different combinations of features	Bhoganahalli	4	5	1056	8467731.96	8467731.67	PASS
5	try different combinations of features	Choodasandra	2	6	1056	56389071.11	5432031.01	FAIL
6	try different combinations of features	EPIP Zone	3	7	1056	8888643.22	8888643.22	PASS
7	try different combinations of features	Girinagar	1	8	1056	28222775.63	28222775.63	PASS
8	try different combinations of features	HSR layout	2	9	1056	7216425.63	7216425.63	PASS
9	try different combinations of features	Jakkur	4	10	1056	11492908.54	114929008.5	PASS
10	try different combinations of features	NRI layout	6	11	1056	8937873.39	8937873.39	PASS

**Table 7.2: Test cases**

## Chapter 8

### Snapshots

#### Screenshot 1:



Welcome to House Price Prediction

Select the Location: 1st Block Jayanagar

Enter BHK:

Enter Number of Bathrooms:

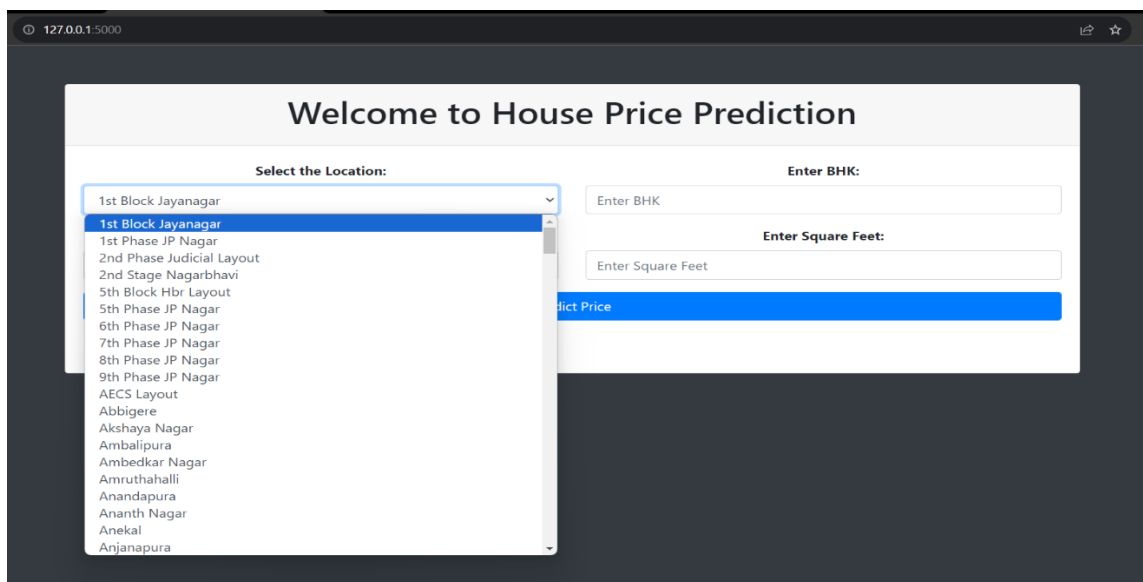
Enter Square Feet:

Predict Price

**Fig 8.1:main Page**

This is the Frontend Design of the Project

#### Screenshot 2:



Welcome to House Price Prediction

Select the Location: 1st Block Jayanagar

Enter BHK:

Enter Number of Bathrooms:

Enter Square Feet:

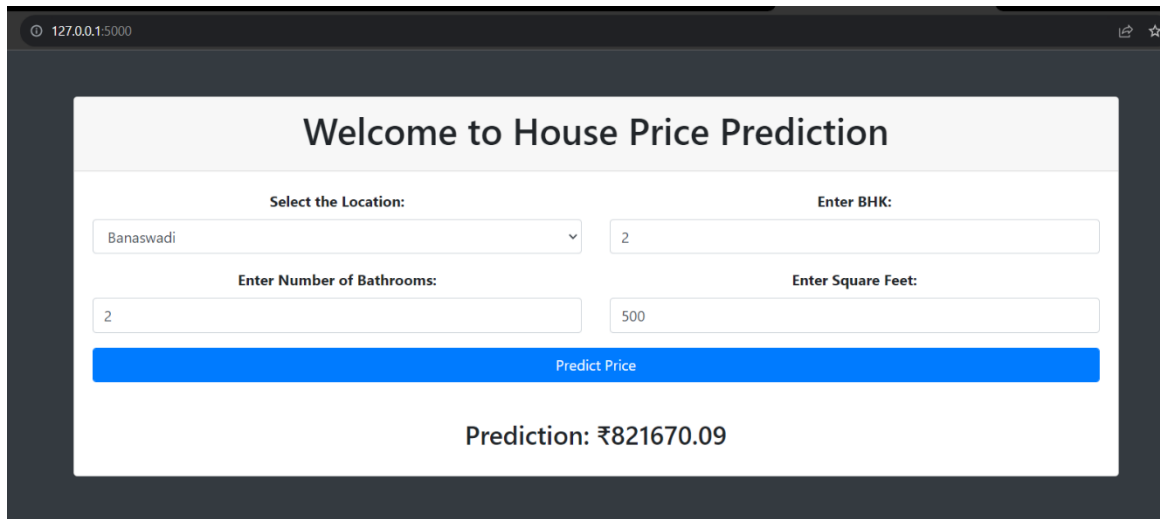
Predict Price

**Fig 8.2:main Page**

# HOUSE PRICE PREDICTION

---

## Screenshot 3:

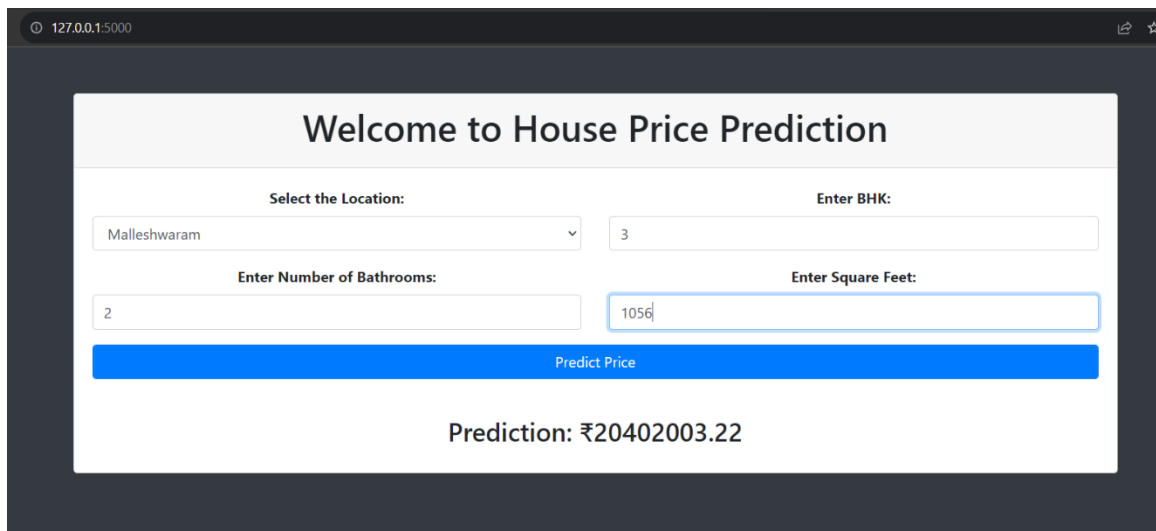


The screenshot shows a web application titled "Welcome to House Price Prediction". It features four input fields: "Select the Location:" with a dropdown menu showing "Banaswadi", "Enter BHK:" with a text input showing "2", "Enter Number of Bathrooms:" with a text input showing "2", and "Enter Square Feet:" with a text input showing "500". Below these fields is a blue button labeled "Predict Price". The prediction result is displayed as "Prediction: ₹821670.09".

**Fig 8.3: Test Example 1**

This screenshot contains the Value Predicted at the Location Banaswadi with 2bathrooms and 3BHK of 1056sqf area

## Screenshot 4:



The screenshot shows the same web application with different inputs: "Select the Location:" dropdown shows "Malleshwaram", "Enter BHK:" text input shows "3", "Enter Number of Bathrooms:" text input shows "2", and "Enter Square Feet:" text input shows "1056". The blue "Predict Price" button is visible. The prediction result is displayed as "Prediction: ₹20402003.22".

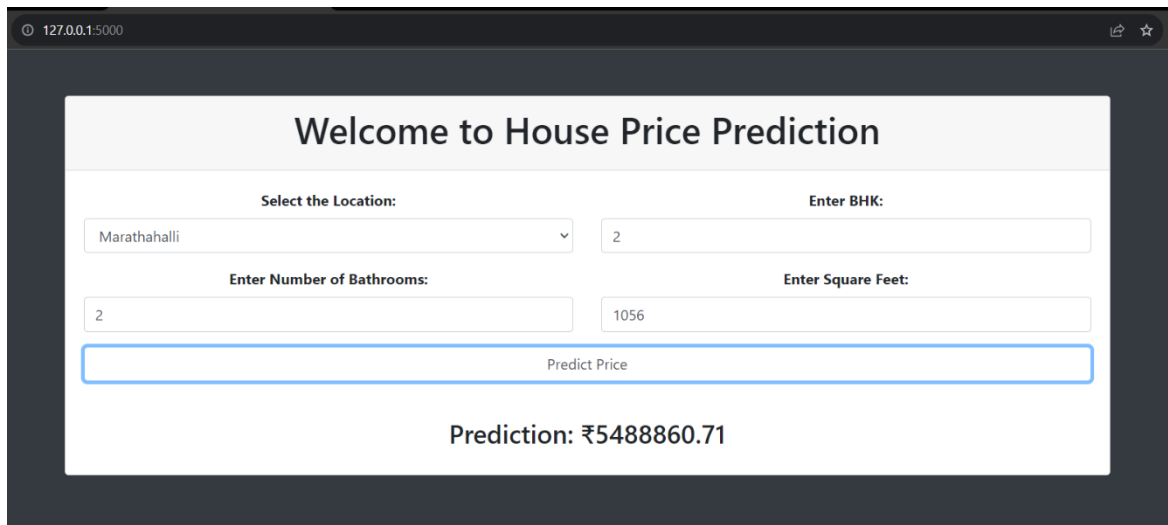
**Fig 8.4: Test Example 2**

This screenshot contains the Value Predicted at the Location Malleshwarm with 2 bathrooms and 3BHK of 1056sqf area

# HOUSE PRICE PREDICTION

---

## Screenshot 5:

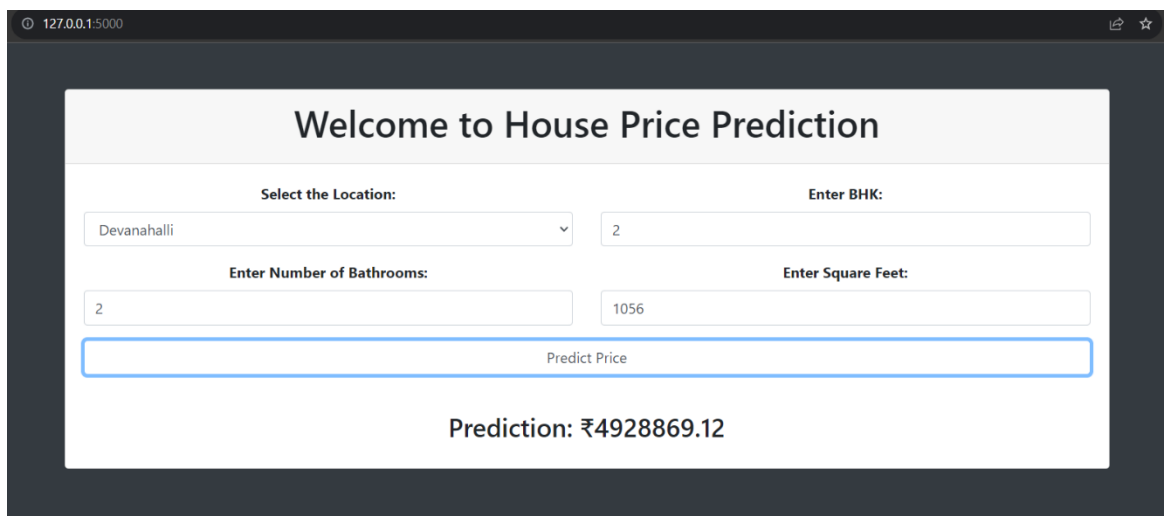


The screenshot shows a web application titled "Welcome to House Price Prediction". It features four input fields: "Select the Location:" with a dropdown menu showing "Marathahalli", "Enter BHK:" with a text input showing "2", "Enter Number of Bathrooms:" with a text input showing "2", and "Enter Square Feet:" with a text input showing "1056". Below these fields is a button labeled "Predict Price". The prediction result is displayed as "Prediction: ₹5488860.71".

**Fig 8.5: Test Example 3**

This screenshot contains the Value Predicted at the Location Marthahalli with 2bathrooms and 2BHK of 1056sqf area

## Screenshot 6:



The screenshot shows the same web application as Screenshot 5, but with the location changed to "Devanahalli" in the dropdown menu. The other input fields remain the same: "Enter BHK:" is "2", "Enter Number of Bathrooms:" is "2", and "Enter Square Feet:" is "1056". The "Predict Price" button is still present. The prediction result is displayed as "Prediction: ₹4928869.12".

**Fig 8.6: Test Example 4**

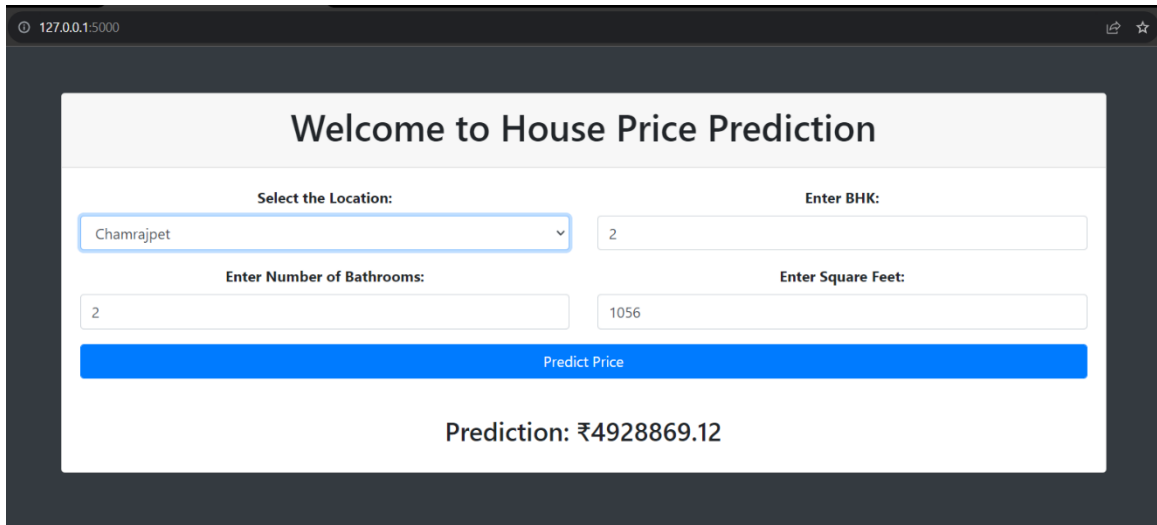
This screenshot contains the Value Predicted at the Location Devanahalli with 2bathrooms and 3BHK of 1056sqf area



# HOUSE PRICE PREDICTION

---

## Screenshot 7:



Welcome to House Price Prediction

Select the Location: Chamrajpet

Enter BHK: 2

Enter Number of Bathrooms: 2

Enter Square Feet: 1056

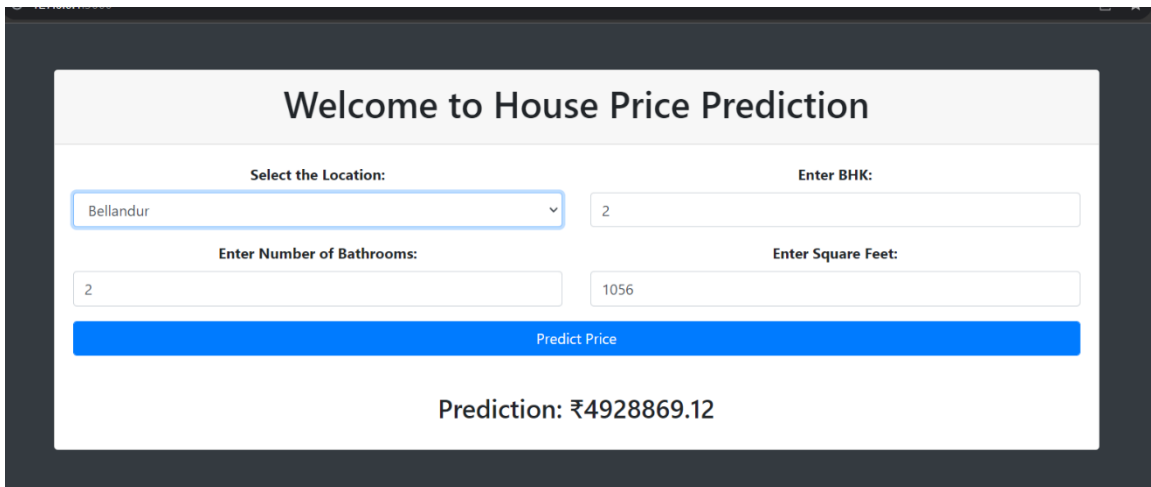
Predict Price

Prediction: ₹4928869.12

**Fig 8.7: Test Example 5**

This screenshot contains the Value Predicted at the Location Chamrajpet with 2bathrooms and 3BHK of 1056sqf area

## Screenshot 8 :



Welcome to House Price Prediction

Select the Location: Bellandur

Enter BHK: 2

Enter Number of Bathrooms: 2

Enter Square Feet: 1056

Predict Price

Prediction: ₹4928869.12

**Fig 8.8: Test Example 6**

This screenshot contains the Value Predicted at the Location Bellandur with 2bathrooms and 3BHK of 1056sqf area

## Chapter 9

### Conclusion

In conclusion, the house price prediction project has been a significant undertaking that has yielded valuable insights and outcomes. Through the meticulous analysis of various housing features and the utilization of advanced machine learning techniques, we have successfully developed a robust model capable of accurately forecasting house prices

Throughout the project, we employed a diverse dataset that encompassed a wide range of factors, such as location, size, area, and economic indicators, ensuring the model's capability to generalize to different scenarios. This model has exhibited remarkable performance, demonstrating its potential applicability in real-world scenarios for property valuation and investment decision

this house price prediction project lays the foundation for making informed decisions in the real estate market, aiding buyers, sellers, and investors in understanding property values more accurately. As technology and data science continue to advance, we envision a promising future for real estate forecasting, with the potential to revolutionize the industry and empower stakeholders with greater knowledge and foresight.

## Chapter 10

### Future Enhancements

Feature enhancement is crucial for improving the accuracy and performance of house price prediction models. By incorporating additional relevant features and enhancing existing ones, the model can capture more patterns and relationships in the data. Here are some feature enhancement techniques to consider:

**Age of the Property:** Calculate the age of the property based on the year of construction and the current year. Older properties might have different price dynamics compared to newer ones.

**Proximity to Amenities:** Calculate the distance to essential amenities like schools, hospitals, parks, supermarkets, and public transportation. Properties closer to these amenities may have higher prices.

**Neighbourhood Information:** Encode information about the neighbourhood, such as crime rates, average income, or school ratings. These factors can impact property prices significantly.

**Historical Price Trends:** Incorporate historical price data for the property or the neighbourhood to capture long-term price trends.

**Special Features:** Identify and encode features like swimming pools, fireplaces, or a view, as they can influence the property's desirability and price.

.

## Chapter 11

### References

- Stack overflow
- YouTube
- GitHub
- <https://www.geeksforgeeks.org/machine-learning/>
- [https://en.m.wikipedia.org/wiki/Machine\\_learning/](https://en.m.wikipedia.org/wiki/Machine_learning/)
- <https://www.ybifoundation.org/course/machine->