# Real Estate Price Prediction Web Application using Linear Regression Model with Supervised Machine Learning Approach

Minor Project Report

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF

BACHELOR OF TECHNOLOGY

(Information Technology)

**Submitted By:**

Palakpreet Kaur (URN - 2004836)

Kunwardeep Singh (URN - 2004948)

Department of Information Technology,
Guru Nanak Dev Engineering College
Ludhiana-141106

# ABSTRACT

Supervised machine learning (ML) algorithms have revolutionized numerous domains by enabling computers to learn patterns and make accurate predictions from labeled datasets. This report explores the performance and generalization capabilities of various supervised ML algorithms, shedding light on their strengths and limitations. The study encompasses a comprehensive evaluation of popular algorithms, including decision trees, support vector machines among others. Real-world datasets are utilized to train and test these algorithms, and their performance is assessed based on metrics such as accuracy, precision, recall. Additionally, the report investigates the impact of varying dataset sizes, feature selection techniques, and hyperparameter tuning on the algorithm performance. The findings demonstrate that different supervised ML algorithms exhibit varying levels of accuracy and robustness across different datasets and problem domains. Moreover, the report presents insights into the trade-offs between model complexity and generalization, highlighting the need for careful algorithm selection and optimization to achieve desirable outcomes. Ultimately, this report serves as a valuable resource for practitioners and researchers seeking to leverage supervised ML algorithms effectively in their respective fields and highlights areas for further exploration and improvement in the domain of machine learning.

# ACKNOWLEDGMENT

We are highly grateful to the Dr.Sehajpal Singh, Principal, Guru Nanak Dev Engineering College (GNDEC), Ludhiana, for providing this opportunity to carry out the minor project at Guru Nanak Dev Engineering College, Ludhiana. The constant guidance and encouragement received from Dr. K. S. Mann, GNDEC Ludhiana has been of great help in carrying out the project work and is acknowledged with reverential thanks. We would like to express a deep sense of gratitude and thanks profusely to Pf.Pankaj Bhambri without the wise counsel and able guidance, it would have been impossible to complete the report in this manner. We express gratitude to other faculty members of Information Technology department of GNDEC for their intellectual support throughout the course of this work.

At last, we are indebted to all whosoever have contributed in this project work.

**Palakpreet Kaur - 2004826**

**Kunwardeep Singh - 2004948**

# Contents

# 1 Introduction

## 1.1 Introduction to Machine Learning

Machine learning is a subfield of artificial intelligence (AI) that focuses on the development of algorithms and models that enable computers to learn from and make predictions or decisions based on data, without being explicitly programmed. It is inspired by the idea that machines can learn and improve from experience, just like humans. Machine learning algorithms are designed to analyze and interpret large volumes of data, identify patterns, and make informed decisions or predictions. These algorithms are trained on datasets, which serve as examples or representations of the problem domain. By learning from these datasets, the algorithms can generalize their knowledge and apply it to new, unseen data.

## 1.2 Types Of Machine Learning

There are various types of machine learning algorithms like –

a) Supervised Machine Learning

b) Unsupervised Machine Learning

c) Reinforcement Machine Learning

### 1.2.1 Supervised Machine Learning

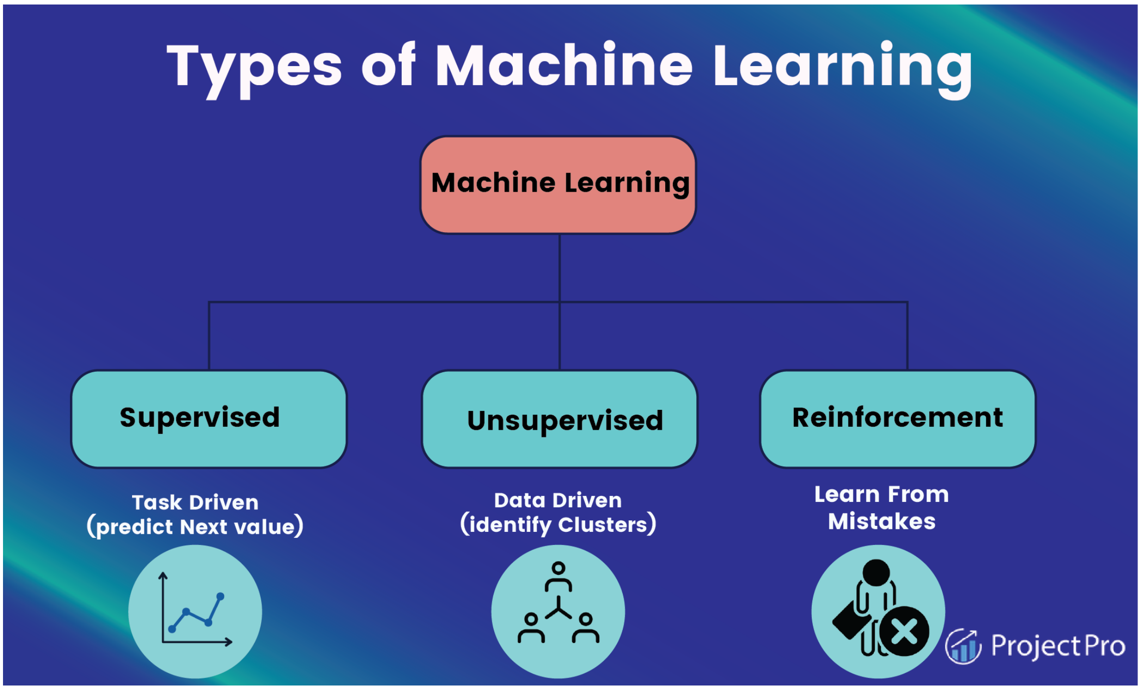
In supervised learning, the algorithm is trained using labeled data, where each input has a corresponding output. The goal is to learn a mapping function that can accurately predict outputs for new, unseen inputs.

### 1.2.2 Unsupervised Machine Learning

Unsupervised learning, on the other hand, deals with unlabeled data. The algorithm is tasked with finding patterns, relationships, or structures in the data without any predefined outputs. Clustering and dimensionality reduction are common applications of unsupervised learning.

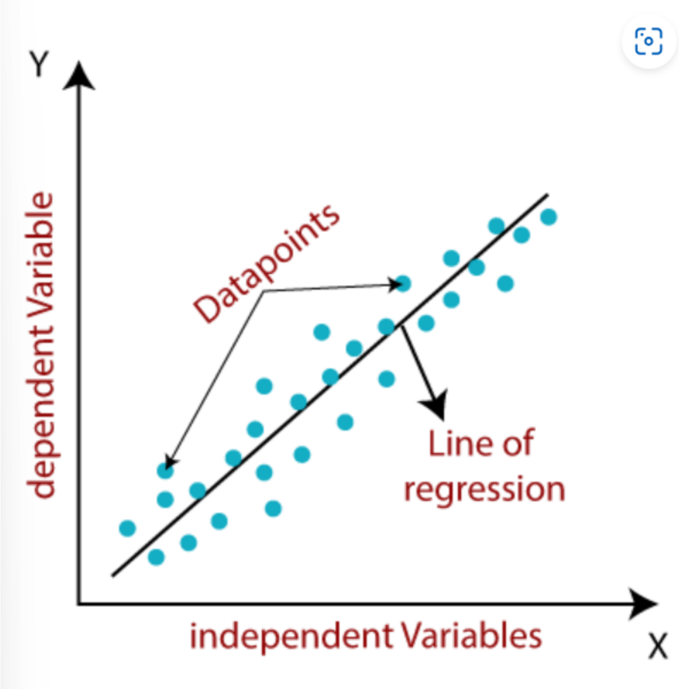
### 1.2.3 Reinforcement Machine Learning

Reinforcement learning involves training an agent to interact with an environment and learn from the feedback it receives. The agent takes actions in the environment and receives rewards or penalties based on its performance. The goal is to find an optimal policy or set of actions that maximize the cumulative reward over time.



## 1.3 Project Category

Our Project is mainly based on Supervised Machine Learning Algorithm. Supervised ML includes the aspect of labelled data along with the use of Linear Regression.Linear regression algorithm shows a linear relationship between a dependent (y) and one or more independent (y) variables, hence called as linear regression. Since linear regression shows the linear relationship, which means it finds how the value of the dependent variable is changing according to the value of the independent variable.



## 1.4 Objectives

The Objectives of this Project are -

1. To study and analyse the various parameters related to the real state properties.

2. To design the prediction model with linear regression model.

3. To measure the accuracy of the predicted model.

## 1.5 Problem Formulation

Earlier it was difficult to predict the price of a Real Estate as different brokers would have different opinion. So, in order to generalise this fluctation of prices a model has been made to do so on the basis of various parameters like Total Area, Location, BHK etc.

# 2 Requirement Analysis and System Specification

## 2.1 Feasibility Study

A feasibility study is done by analyzing technical, economic, legal, operational and time feasibility factors. This chapter describes all the feasibilities that come as questions to both the developers and other users during the development of software. The chapter contains technical feasibility, economic feasibility and operational feasibility. Our proposed system is legally feasible as our project meet all the legal requirements. Our proposed system is time feasible as time taken for execution and completion is reasonable. A quantitative approach is proposed to obtain measurable, comparable judgments of simulation correctness. . The feasibility of our proposed system can be evaluated as:

1. Technical Feasibility:- The tools and technologies used in the system are:

   **Code Editor:** Visual Studio Code

   **Languages:**Python, HTML, CSS, JavaScript

   **API Testing:**Postman

   **Model Creation:** Jupyter Notebook

   **Backend Server:** Flask Server

   These softwares are free to use and are available for students. To install all these in your system, you need any regular laptop/PC with minimum 4GB RAM.

2. Economic Feasibility:-The proposed system is economically feasible as the manual prediction system shows a lot of price fluctuations and thus not have a redundent sytem but this project developed is software based, so it only require installation cost and a low maintenance cost.

3. Operational Feasibility:-Operational feasibility is a measure of how well the proposed system will solve the problems, and takes advantage of the opportunities that are identified during scope definition and how it satisfies the requirements identified in the

requirements analysis phase of system development. Operational feasibility reviews the willingness of the organization to support the proposed system. This is probably the most difficult of the possibilities to gauge. In order to determine this feasibility, it is essential to understand the management commitment towards the proposed project. The project is easy to use and can run on any system having the required specifications and system irrespective of the OS used.As the proposed system was very light weighted and small sized, this system is expected to operate everywhere. The requirements of this system are very small therefore it is easy to operate in every environment.As all components needed to develop the proposed system are also available,the system will definitely work. Hence the project is operationally feasible.

## 2.2 Software Requirement Specification document

**Purpose of SRS document:** The purpose of this document is to define the functional and non-functional requirements of the Real Estate Price Prediction Model. It serves as a guideline for the development team and stakeholders involved in the project. **Scope:**The Real Estate Price Prediction Model aims to provide accurate price predictions for real estate properties based on relevant features such as location, size, number of bedrooms, and other factors. The model will be developed using Linear Regression, a supervised learning technique that analyzes the relationship between dependent and independent variables. **Functional Requirements:**

1. **Hardware Requirements:-**

   i. RAM: A minimum of 4GB is required for training any model and building a server.

   ii. OS: This system can run on any latest LTS OS.

   iii. Processor: Intel 35 10th Gen or above or Ryzen 5 4th Gen or above.

2. **Software Requirements:-**

   i. IDE Used - Visual Studio is the fastest IDE for productivity.Visual Studio Code combines the simplicity of a source code editor with powerful developer tooling, like IntelliSense code completion and debugging. First and foremost, it is an editor that gets out of your way. The delightfully frictionless edit-build-debug cycle means less time fiddling with your environment, and more time executing on your ideas. Visual Studio Code supports macOS, Linux, and Windows - so you can hit the ground running, no matter the platform.

ii. Languages Used - Python, HTML, CSS JavaScript

iii. Model Creation - Jupyter Notebook. The Jupyter Notebook is the original web application for creating and sharing computational documents. It offers a simple, streamlined, document-centric experience.

iv. API Testing - Postman. Postman is an API platform for building and using APIs. Postman simplifies each step of the API lifecycle and streamlines collaboration so you can create better APIs faster .

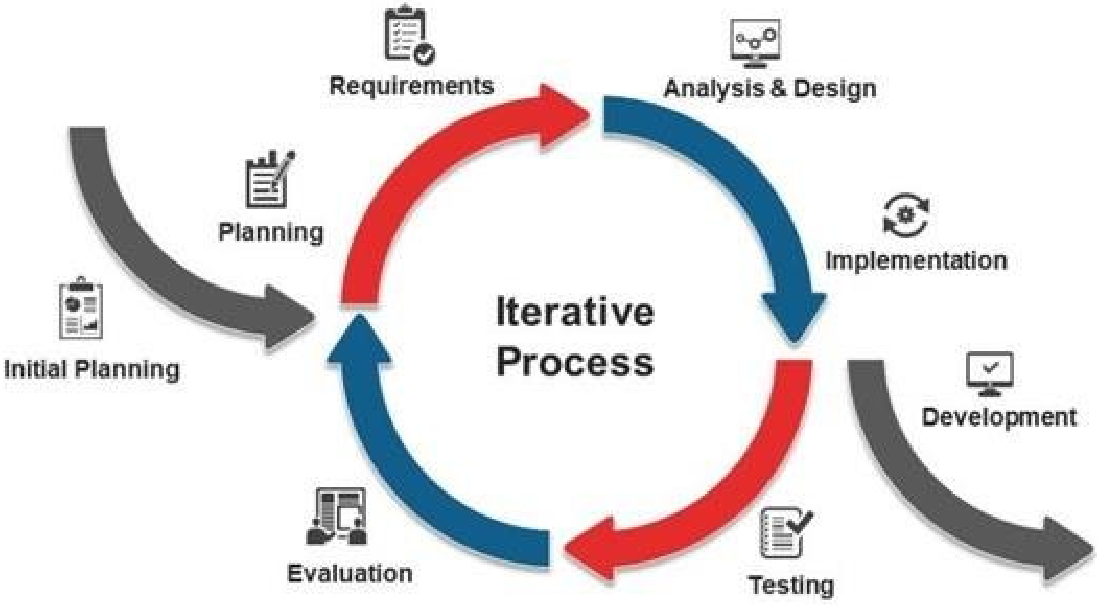
v. Backend Server - Flask Server. Flask server is defined as server software that is capable of running HTTP requests on the public world wide web, private LAN, and private WANs and comprises of one or many computers bundled together and dedicatedly working for running the software application on the worldwide web.

## 2.3 Expected Hurdles:

While making this model we came across many hurdles, some of them are listed below: To perform data cleaning by using using various techniques of Feature Engineering. To detect Outliers and their reduction. To enable Hot Encoding for preprocessing of data.

## 2.4 SDLC Model to be Used:

The SDLC Model used in this project is Iterative Model. In the Iterative model, iterative process starts with a simple implementation of a small set of the software requirements and iteratively enhances the evolving versions until the complete system is implemented and ready to be deployed. An iterative life cycle model does not attempt to start with a full specification of requirements. Instead, development begins by specifying and implementing just part of the software, which is then reviewed to identify further requirements. This process is then repeated and a model is obtained at the end.



**Phases of Iterative Model:** The development of project took place in multiple phases of the model:

i. Requirements gathering and analysis: All the requirements were collected and then checked and analysed whether the particular requirement can be fulfilled or not. The analysis includes analysing whether the particular requirement is feasible or not and the budget allows it or not.

ii. Design: After the iteration requirement is gathered then we need to implement the design phase. Effective design is decided to implement the requirement out of many alternatives. This is one of the critical phases as proper design can provide the most optimal outputs will low pressure on the funds from client. This design can be a new one or extension to the already build requirement or it can be flow chart or data-flow diagram.

iii. Implementation: In this, the basic codes are written and then transformed into computer program's. Here the code is written. With respect to this Project firstly a model is created and for that model a local flask server is built up.Then a User friendly Interface is developed for the Client.

iv. Testing: We train the model using the training data-set which includes the price of the houses from the dataset which we have provided .Once the code has been implemented then this testing phase is implemented to identify any defects that are present in the code and if present then they need to be reported back to the developers. The tester can write new test cases or use existing one which they have written in previous build but the through testing is a priority as any miss will impact the specification of software.

v. Deployment: The project is currently deployed on the local server and uses it as the working environment.

vi. Review: In this phase, the developed requirement is reviewed to meet all the standards as per the currently decided requirement. Basing on this further plan requirement plan is drafted and implemented as part of the next iteration cycle.

vii. Maintenance: After the deployment of the project, there might be some bugs left, so some updates will be required. This involves debugging and adding new additional features.

# 3 System Design

## 3.1 Introduction

The system design is the most creative and challenging phase of the system development life cycle. It is an approach for the creation of the proposed system, which still helps the system coding. It provides the understanding and procedural details necessary for implementing the system. A number of subsystems are to be identified which constitutes the whole system. In this phase the data organization is to be discussed, in which the output formats are to be designed. The system design is composed of several steps. Here the emphasis is on translating the performance requirements in to the design specification. To create a complete project , we must work on below given very distinct phases:

1. Model Creation using Jupyter Notebook
2. Libraries
3. Decision Tree Regression
4. Flask Server
5. Visual Studio Code IDE
6. API building and Testing using Postman

When you going to Design System you have to some knowledge about the thing you want to use. Here are some overview about tools:-

## 3.2 What is Jupyter Notebook?

Jupyter Notebook is an open-source web application that allows users to create and share documents that contain live code, equations, visualizations, and narrative text. It is widely used for data exploration, analysis, visualization, and prototyping in various fields such as data science, machine learning, and scientific research. The key features of Jupyter Notebook include:

**Interactive Environment:** Jupyter Notebook provides an interactive computing environment where users can write and execute code in a cell-based structure. Each cell can contain code (Python, R, Julia, etc.) or markdown text.

Code Execution: Users can execute the code in individual cells and see the output, which can include tables, plots, or any other visual representation of data.

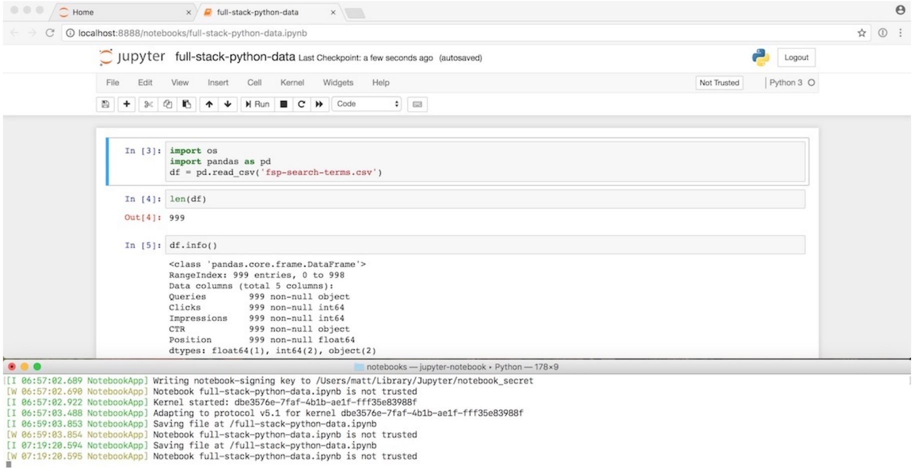
**Visualization:** Jupyter Notebook allows the integration of data visualizations and graphical outputs directly into the document, making it easy to explore and analyze data visually.

**Markdown Support:** Users can include formatted text, equations, and explanatory narratives using Markdown language. This allows for the creation of interactive documents that blend code, visualizations, and textual explanations.

**Notebook Sharing:** Jupyter Notebook documents can be easily shared with others by exporting them in various formats like HTML, PDF, or as executable notebooks. This makes it convenient for collaboration and sharing research findings.

**Extensible Ecosystem:** Jupyter Notebook is part of the Jupyter project, which supports a wide range of programming languages through different kernels. This means that users can write code in multiple languages within the same notebook.

Jupyter Notebook is accessed through a web browser, providing a user-friendly interface for creating, editing, and running code cells. It promotes an iterative and exploratory approach to coding and analysis, allowing users to interact with their code and data in a flexible and interactive manner.


Home
full-stack-python-data
localhost:8888/notebooks/full-stack-python-data.ipynb
jupyter full-stack-python-data Last Checkpoint: a few seconds ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Code
notebooks — jupyter-notebook ▸ Python — 178×9

```
In [3]: import os
        import pandas as pd
        df = pd.read_csv('fsp-search-terms.csv')

In [4]: len(df)

Out[4]: 999

In [5]: df.info()

        <class 'pandas.core.frame.DataFrame'>
        RangeIndex: 999 entries, 0 to 998
        Data columns (total 5 columns):
        Queries        999 non-null object
        Clicks         999 non-null int64
        Impressions    999 non-null int64
        CTR            999 non-null object
        Position       999 non-null float64
        dtypes: float64(1), int64(2), object(2)
```

```
[I 06:57:02.689 NotebookApp] Writing notebook-signing key to /Users/matt/Library/Jupyter/notebook_secret
[W 06:57:02.690 NotebookApp] Notebook full-stack-python-data.ipynb is not trusted
[I 06:57:02.922 NotebookApp] Kernel started: dbe3576e-7faf-4b1b-ae1f-fff35e83988f
[I 06:57:03.488 NotebookApp] Adapting to protocol v5.1 for kernel dbe3576e-7faf-4b1b-ae1f-fff35e83988f
[I 06:59:03.853 NotebookApp] Saving file at /full-stack-python-data.ipynb
[W 06:59:03.854 NotebookApp] Notebook full-stack-python-data.ipynb is not trusted
[I 07:19:20.594 NotebookApp] Saving file at /full-stack-python-data.ipynb
[W 07:19:20.595 NotebookApp] Notebook full-stack-python-data.ipynb is not trusted
```

## 3.3 What are Libraries?

Machine learning libraries are software packages that provide a collection of pre-built functions, algorithms, and tools specifically designed to facilitate the development and implementation of machine learning models. These libraries aim to simplify the process of building and training machine learning models by providing efficient and optimized implementations of various algorithms and techniques. Some of them which we have used are

**scikit-learn:** scikit-learn is a widely used machine learning library in Python. It offers a comprehensive set of tools for classification, regression, clustering, dimensionality reduction, model selection, and preprocessing. It provides a consistent interface and supports a wide range of algorithms such as decision trees, random forests, support vector machines (SVM), k-nearest neighbors (KNN), and many more.

**Pandas:**Pandas is a popular open-source data manipulation and analysis library for Python. It provides high-performance, easy-to-use data structures and data analysis tools, making it a valuable tool for data preprocessing and exploratory data analysis in machine learning and data science projects.

**Numpy:** NumPy (Numerical Python) is a fundamental library for scientific computing in Python. It provides efficient and optimized implementations of multi-dimensional arrays, mathematical functions, linear algebra operations, random number generation, and more.

**Matplotlib:** Matplotlib is a data visualization library in Python that provides a wide range of tools for creating static, animated, and interactive visualizations. It allows users to create various types of plots, charts, histograms, scatter plots, bar plots, and more

## 3.4 What is Decision Tree Regressor?

The Decision Tree Regressor is a supervised machine learning algorithm that is used for regression tasks. It builds a decision tree model based on the training data, where each internal node represents a feature or attribute, each branch represents a decision rule, and each leaf node represents the predicted value.
Here's an overview of how the Decision Tree Regressor works:

**Data Splitting:** The algorithm starts with the entire training dataset and selects a feature that best splits the data into homogeneous subsets. It uses various metrics such as mean squared error (MSE) or mean absolute error (MAE) to evaluate the quality of the splits.

**Recursive Splitting:** Once the initial split is made, the process is recursively repeated for each resulting subset or child node. The algorithm selects the best feature and split point for each child node based on the same splitting criteria.

**Leaf Node Creation:** The splitting process continues until a stopping criterion is met. This criterion can be a maximum depth of the tree, a minimum number of samples required to split a node, or a minimum reduction in the impurity measure.

**Value Assignment:** When a leaf node is reached, the Decision Tree Regressor assigns a predicted value to the samples in that node. This value is typically the average or median of the target variable values of the samples in that leaf.

**Prediction:** To make predictions for new data points, the algorithm traverses the decision tree based on the feature values of the input data. It follows the decision rules at each node until it reaches a leaf node, and then assigns the predicted value of that leaf node to the input data.

The Decision Tree Regressor has several advantages, including its interpretability, ability to capture non-linear relationships, and handling of both numerical and categorical features. However, it can suffer from overfitting, especially if the tree depth is too large or there are too few samples

## 3.5 What is a Flask Server?

Flask is a lightweight and popular web framework in Python used for developing web applications. A Flask server refers to the web server that is created using the Flask framework to host and serve your web application.
Here's an overview of how a Flask server works:

**Installation**: First, you need to install Flask on your machine using the pip package manager. You can install Flask by running the following command in your terminal or command prompt:
pip install flask

**Creating the Flask App:** To create a Flask server, you typically start by defining a Flask application instance. This is done by creating a Python file and importing the Flask module.

**Defining Routes:** Routes in Flask define the different URLs or endpoints that your web application will respond to. You can define routes using the @app.route decorator. Each route corresponds to a specific URL, and you can associate functions with these routes to handle requests and generate responses. For example, you can define a route that responds to the root URL ("/") as follows:
@app.route('/') def index(): return 'Hello, Flask!'

**Running the Server:** To run the Flask server and make your web application accessible, you need to call the run() method on the Flask application instance. This method starts the development server provided by Flask. You can then run your Flask server by executing the Python script. The server will start, and you can access your web application by visiting the specified URL (usually http://localhost:5000/) in your web browser.

**Handling Requests:** Flask provides various methods to handle different types of HTTP requests such as GET, POST, PUT, DELETE, etc. You can define functions for each route that process incoming requests and generate responses based on the request data. Flask provides request and response objects that allow you to access request data, headers, form data, and construct response objects.

**Templates:** Flask allows you to use template engines like Jinja2 to generate dynamic HTML pages. You can define HTML templates with placeholders for dynamic content, and Flask will render these templates with the provided data.

Flask provides a flexible and customizable framework for building web applications in Python. It supports various features like URL routing, request handling, form processing, session management, authentication, and more. Additionally, Flask has a large ecosystem of extensions and plugins that can be integrated to add additional functionality to your web application.

## 3.6 What is VS Code IDE?

VS Code (Visual Studio Code) is a popular integrated development environment (IDE) developed by Microsoft. It is a lightweight, cross-platform code editor that provides a rich set of features for coding, debugging, and version control. VS Code is widely used by developers for various programming languages and frameworks.
Here are some key features of VS Code:

**Cross-Platform Support:** VS Code is available for Windows, macOS, and Linux operating systems, allowing developers to use the same IDE across different platforms.

**Intuitive User Interface:** VS Code has a clean and intuitive user interface with a minimalistic design. It provides a sidebar for file navigation, a powerful editor with syntax highlighting and IntelliSense (code completion), and a customizable layout that allows users to arrange panels according to their preferences.

**Extensions and Marketplace:** VS Code has a rich extension ecosystem that enables developers to enhance their coding experience. There are thousands of extensions available in the VS Code marketplace for various programming languages, frameworks, and tools. These extensions add functionalities such as language support, linters, code formatters, Git integration, and more.

**Integrated Terminal:** VS Code includes an integrated terminal within the IDE, allowing developers to execute commands and interact with the shell without switching to an external terminal window.

**Debugging Support:** VS Code provides built-in debugging capabilities for several programming languages and frameworks. It allows users to set breakpoints, step through code, inspect variables, and monitor the execution flow of their programs.

**Version Control Integration:** VS Code has seamless integration with version control systems such as Git. It provides features like source control management, visual diffing, and easy access to common Git operations directly from the IDE.

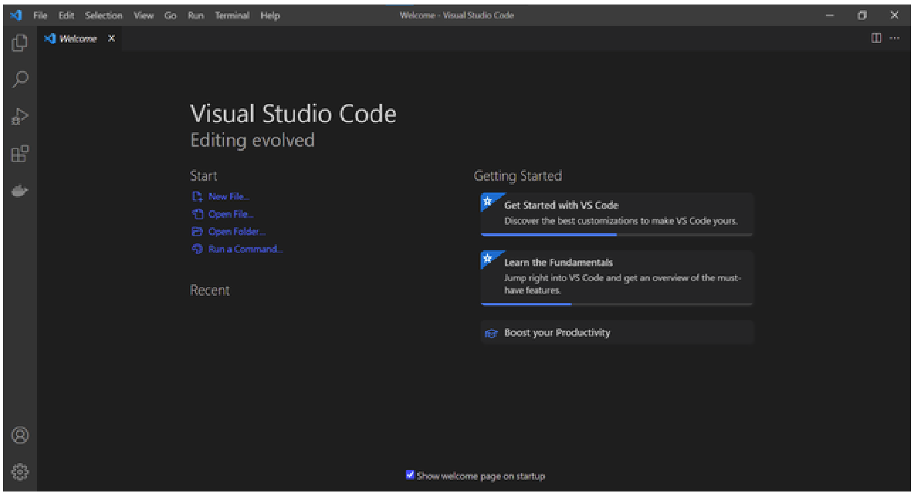
**Task Automation:** VS Code supports task automation through the use of tasks and the integrated task runner. Developers can define custom tasks to execute build processes, run tests, or perform other automation tasks.

**IntelliSense and Code Navigation:** VS Code offers intelligent code completion and suggestions (IntelliSense) based on the context of the code being written. It also provides features like Go to Definition, Find All References, and Peek Definition, which enable easy code navigation and exploration.

**Customization:** VS Code is highly customizable to suit individual preferences. Users can customize the editor's theme, keybindings, and settings according to their needs. They can also install different UI themes and icons to personalize the look and feel of the IDE.

**Integrated Terminal:** VS Code includes an integrated terminal within the IDE, allowing developers to execute commands and interact with the shell without switching to an external terminal window.

VS Code has gained popularity due to its lightweight nature, extensibility, and ease of use. It has a large and active community of users and developers who contribute to the extension ecosystem, creating a vibrant and supportive ecosystem around the IDE.



Discover the best customizations to make VS Code yours.

Jump right into VS Code and get an overview of the must-have features.

## 3.7 What is Postman?

Postman is a popular API development and testing tool that allows you to send HTTP requests to APIs and receive responses. It provides a user-friendly interface for constructing requests, setting headers and parameters, and analyzing the responses. Here's an overview of how you can use Postman for API building and testing:

**Installing Postman:** You can download and install Postman from the official Postman website based on your operating system. Postman is available for Windows, macOS, and Linux.

**Creating a Request:** Launch Postman and start by creating a new request. You can specify the HTTP method (GET, POST, PUT, DELETE, etc.) and the URL of the API endpoint you want to test. You can also add headers, query parameters, request body, and other relevant information.

**Sending a Request:** After configuring the request details, you can click the "Send" button to send the request to the API server. Postman will display the response received from the server, including the status code, headers, and the response body.

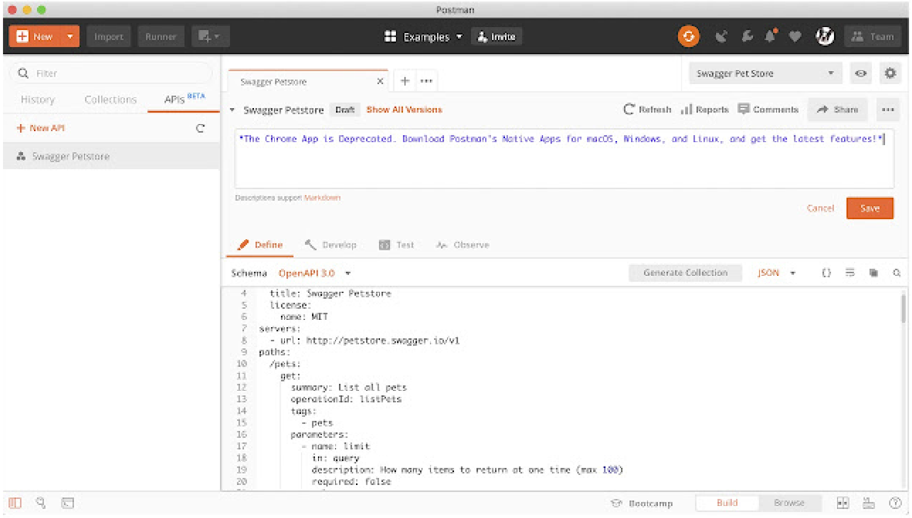
**Analyzing the Response:** Postman provides various features to analyze and inspect the API response. You can view the response in different formats (e.g., raw, JSON, XML), format the response for better readability, and access specific fields or values within the response.

**Saving and Organizing Requests:** Postman allows you to save your requests as collections, which are a group of related requests. You can organize your requests into folders and subfolders within collections to keep them well-structured and easily accessible.

**Testing and Automation:** Postman provides a testing framework that allows you to write tests to verify the API's behavior. You can write tests in JavaScript using the built-in Postman sandbox, and these tests can be executed automatically whenever a request is sent. This helps ensure the API's functionality remains intact during development and future updates.

**API Documentation:** Postman can generate documentation for your APIs based on the requests and collections you've created. This documentation can be shared with team members or published for external users to understand and consume your APIs easily.

Postman also offers advanced features like environment variables, pre-request scripts, response validation, and collaboration features that facilitate teamwork and streamline the API development and testing process.



"The Chrome App is Deprecated. Download Postman's Native Apps for macOS, Windows, and Linux, and get the latest features!"

```
  title: Swagger Petstore
  license:
    name: MIT
servers:
  - url: http://petstore.swagger.io/v1
paths:
  /pets:
    get:
      summary: List all pets
      operationId: listPets
      tags:
        - pets
      parameters:
        - name: limit
          in: query
          description: How many items to return at one time (max 100)
          required: false
```

## 3.8 Methodology

Firstly,a real time dataset of house prices along with various parameters is taken on which a modell using Linear Regression is trained to predict the price after giving the input for various parameters like Total Area, location, BHK etc. then that created model is converted into the form of a pickle file using which a Flask server is created and through that server a request is sent to the model to get location names and predivt the price of the property. After that a User friendly interface is made using varius Front-end Development tools which works on the Request-Response methodology and gives us the output.

# 4 Implementation, Testing and Maintenance

Implementation is one of the most important tasks in project implementation is the phase, in which one has to be cautious, because all the efforts undertaken during the project will be fruitful only if the software is properly implemented according to the plans made.The implementation phase is less creative than system design. It is primarily concerned with usertraining, site preparation and file-sites, the tests of the network along with the systems are also included under implementation.Depending on the nature of the systems extensive user training may be required. Programming is itself a design works. The initial parameters of the management information systems should be modified as a result of programming efforts. Programming provides a real test for the assumption made by the analyst. System testing checks the readiness and accuracy of the system to access, update and retrieve data from the new files. Once the programs become available the test data are read into the computer and processed. In most conventions parallel run was conducted to establish the efficiency of the system.Implementation is used here to mean the process of converting a new or revised system in to an operational one. Conversion is one aspect of one implementation

## 4.1 Hardware Development

A computer or laptop with adequate RAM and HDD (Intel Core i3 or equivalent, 20 GB free space on HDD for installing various software), any operating system (Windows, Linux, MacOS), installed with the most recent version of any web browsers that support contemporary JavaScript (Brave, Chrome, Mozilla Firefox), as well as a text editor such as notepad, notepad++, etc., are required. Just a basic setup is required in terms of hardware.

## 4.2 Programming Language

1. Python: Python is a high-level programming language that is widely used for various purposes, including web development, data analysis, artificial intelligence, machine learning, scripting, and more. Python emphasizes code readability and simplicity, making it easier to write and understand compared to other programming languages.

2. HTML: Hypertext Markup Language revision 5 (HTML5) is markup language for the structure and presentation of World Wide Web contents. HTML5 supports the tra-

ditional HTML and XHTML-style syntax and other new features in its markup, New APIs, XHTML and error handling.

3. CSS: Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable. CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs,variations in display for different devices and screen sizes as well as a variety of other effects. CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

4. JavaScript: : JavaScript is a lightweight, cross-platform, and interpreted scripting language. It is well-known for the development of web pages, many non-browser environments also use it. JavaScript can be used for client-side as well as Server side developments.JavaScript contains a standard library of objects, like Array and Math, and a core set of language elements like operators, control structures, and statements.

## 4.3 Operating System

Unlike some iOS or some types of application development, the tools for web development and model creation are all operating system agnostic. You can write HTML, CSS, and JavaScript on Windows, Mac, Linux, and even ChromeOS if you're willing to get a bit creative. So while you won't find a definitive answer in this article, here are a few pros and cons for each of the three major operating systems, Windows, Mac, and Linux.

## 4.4 Test Environment

### Hardware and Software

1. Laptop/PC

2. VS Code IDE

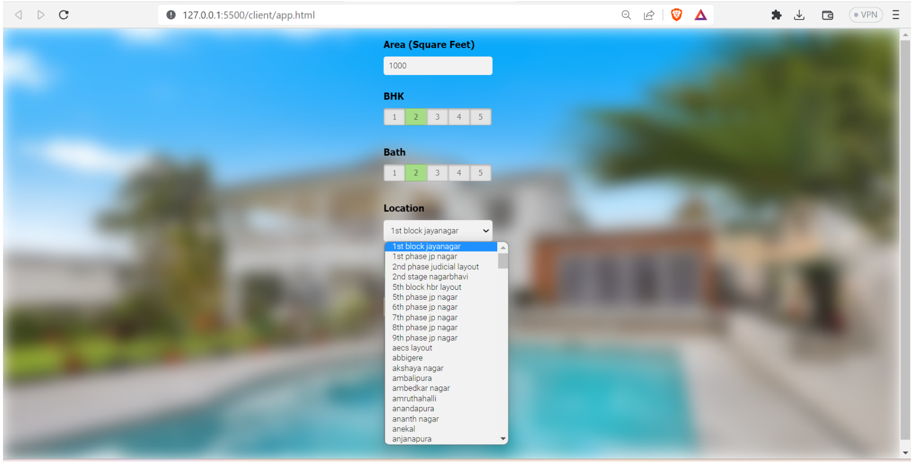
3. Python and required Libraries

4. APIs

# 5 Project Output

Price prediction Model using Linear regression along with Supervised learning technique is ready for generating the output.

1. The first output which we will check will be that our Model is not just based on the static HTML code which is written but also sends the Request and fetches a Response for the required Query.



127.0.0.1:5500/client/app.html

Area (Square Feet)

1000

BHK

1 2 3 4 5

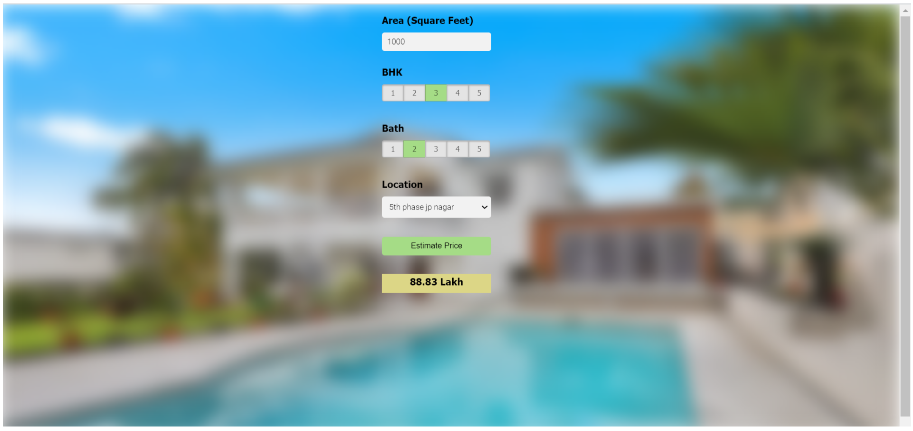
Bath

1 2 3 4 5

Location

1st block jayanagar

2. The second and the main thing which we will check is that it is requesting the predicting price function in the server to make a request to the model for the predicting the price based on the training dataset.



Area (Square Feet)

1000

BHK

1 2 3 4 5

Bath

1 2 3 4 5

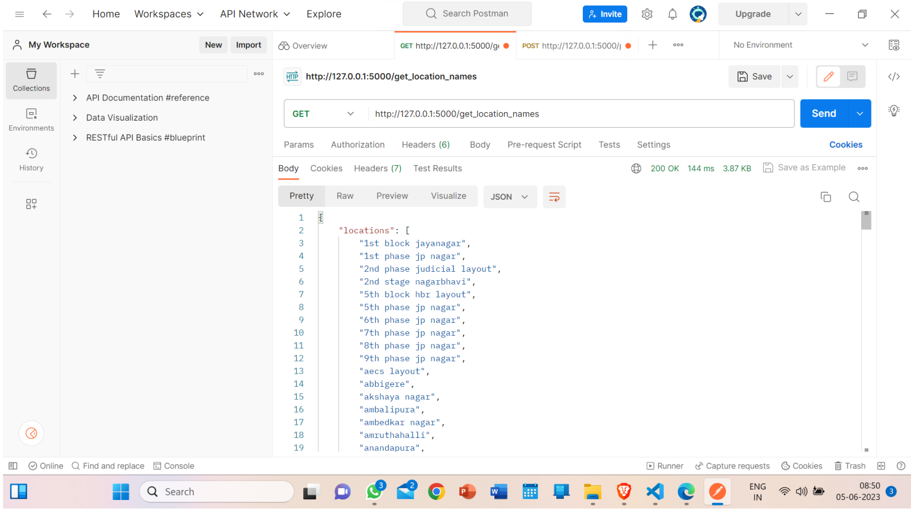
Location

5th phase jp nagar

88.83 Lakh

This was the main outcome of the project but now let's have have a look on the APIs Requests.

The Request for getting all location names.

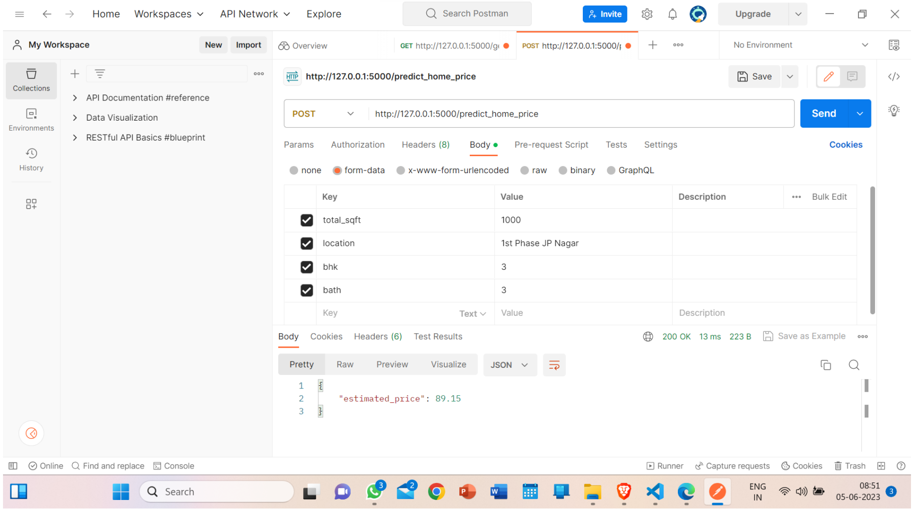


GET http://127.0.0.1:5000/get_location_names

200 OK 144 ms 3.87 KB

```
{
    "locations": [
        "1st block jayanagar",
        "1st phase jp nagar",
        "2nd phase judicial layout",
        "2nd stage nagarbhavi",
        "5th block hbr layout",
        "5th phase jp nagar",
        "6th phase jp nagar",
        "7th phase jp nagar",
        "8th phase jp nagar",
        "9th phase jp nagar",
        "aecs layout",
        "abbigere",
        "akshaya nagar",
        "ambalipura",
        "ambedkar nagar",
        "amruthahalli",
        "anandapura",
```

The Reuest for predicting the price of the desired location based on various parameters.



POST http://127.0.0.1:5000/predict_home_price

○ none ● form-data ○ x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

| | Key | Value | Description |
|---|---|---|---|
| ☑ | total_sqft | 1000 | |
| ☑ | location | 1st Phase JP Nagar | |
| ☑ | bhk | 3 | |
| ☑ | bath | 3 | |

200 OK 13 ms 223 B

```
{
    "estimated_price": 89.15
}
```

# 6 Conclusion and Future Scope

## 6.1 Conclusion

The real estate price prediction model is a valuable tool that can assist in estimating property prices based on various features and historical data. By leveraging machine learning techniques, such as regression algorithms, the model can provide valuable insights and aid in decision-making for buyers, sellers, and real estate professionals. Throughout the development process, several steps were followed, including data collection, preprocessing, feature engineering, model training, and evaluation. The model's ability to generalize and make accurate predictions on unseen data is crucial in determining its effectiveness.

## 6.2 Future Scope

The real estate price prediction model has several potential areas for future development and enhancement:

**Incorporating Additional Features:** Consider including more features that might impact property prices, such as proximity to schools, hospitals, transportation hubs, crime rates, or environmental factors. Expanding the feature set can improve the model's predictive capabilities.

**Exploring Advanced Machine Learning Techniques:** Explore advanced algorithms and techniques beyond linear regression, such as ensemble methods (e.g., random forests, gradient boosting), deep learning models, or support vector machines. Experimenting with different models can potentially enhance the accuracy and robustness of predictions.

# 7 Refrences

1. Fred. Drake [Python Documentation] https://docs.python.org/3/ (Accessed Mar. 15,2023)

2. Brian. Granger [Jupyter Documentation] https://jupyter.org/ (Accessed Mar. 20,2023)

3. Abhinav. Asthana [Postman Documentation] https://www.postman.com/api-documentation-tool/ (Accessed Mar. 26,2023)

4. Amurtha. K [Feature Engineering]https://www.analyticsvidhya.com/blog/2021/03/step-by-step-process-of-feature-engineering-for-machine-learning-algorithms-in-data-science/ (Accessed Mar. 30,2023)

5. David. Cournapeau [scikit-learn Documentation] https://scikit-learn.org/stable/(Accessed Apr 5,2023)

6. Adam. Smith [Linear Regression Model]https://studygyaan.com/data-science-ml/linear-regression-machine-learning-project-for-house-price-prediction (Accessed Apr. 17,2023)

7. Claudia. Boone [Standard Deviation] https://examples.yourdictionary.com/examples-of-standard-deviation.html(Accessed Apr.20,2023)

8. Armin. Rocharer [Flask Documentation] https://flask.palletsprojects.com/en/2.3.x/(Accessed Apr. 27,2023)

9. Abhinav. Asthana [Postman Documentation] https://www.postman.com/api-documentation-tool/ (Accessed May 15,2023)