# "House Rent Prediction System Using Machine Learning & Deep Learning"

**A Project Report Submitted to**
**Rajiv Gandhi Proudyogiki Vishwavidyalaya**

**Towards Partial Fulfillment for the Award of**
**Bachelor of Engineering in *Computer Science & Engineering***

***Submitted by:***

**Aeshna Jain(0827CS201017)**

***Guided by:***

**Prof. Priyanka Jangde**
**Associate Professor**
**Computer Science & Engineering**

***Acropolis Institute of Technology & Research, Indore***

**Jan – June 2023**

# EXAMINER APPROVAL

The Project entitled ***"House rent prediction system using machine learning and deep learning"*** submitted by **Aeshna Jain (0827CS201017)** has been examined and is hereby approved towards partial fulfillment for the award of Bachelor of Engineering degree in Computer Science & Engineering discipline, for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein, but approve the project only for the purpose for which it has been submitted.

**(Internal Examiner)**

**Date:**

**(External Examiner)**

**Date:**

# GUIDE RECOMMENDATION

This is to certify that the work embodied in this project ***“House rent prediction system using machine learning and deep learning”*** submitted by **Aeshna Jain (0827CS201017)** is a satisfactory account of the bonafide work done under the supervision of **Prof. Priyanka Jangde** are recommended towards partial fulfillment for the award of the Bachelor of Engineering (Computer Science & Engineering) degree by Rajiv Gandhi Proudyogiki Vishwavidhyalaya, Bhopal.

**(Project Guide)** **(Project Coordinator)**

# STUDENTS UNDERTAKING

This is to certify that project entitled **"*"House rent prediction system using machine learning and deep learning*"** has developed by us under the supervision of **Prof. Priyanka Jangde**. The whole responsibility of work done in this project is ours. The sole intension of this work is only for practical learning and research.

We further declare that to the best of our knowledge; this report does not contain any part of any work which has been submitted for the award of any degree either in this University or in any other University / Deemed University without proper citation and if the same work found then we are liable for explanation to this.**)**

**Aeshna Jain (0827CS201017)**

# Acknowledgement

We thank the almighty Lord for giving me the strength and courage to sail out through the tough and reach on shore safely. There are number of people without whom this projects work would not have been feasible. Their high academic standards and personal integrity provided me with continuous guidance and support. We owe a debt of sincere gratitude, deep sense of reverence and respect to our guide and mentors **Prof. Priyanka Jangde**, Associate Professor, AITR, for their motivation, sagacious guidance, constant encouragement, vigilant supervision and valuable critical appreciation throughout this project work, which helped us to successfully complete the project on time.

We express profound gratitude and heartfelt thanks to **Dr Kamal Kumar Sethi**, HOD CSE, AITR Indore for his support, suggestion and inspiration for carrying out this project. I am very much thankful to other faculty and staff members of CSE Dept, AITR Indore for providing me all support, help and advice during the project. We would be failing in our duty if do not acknowledge the support and guidance received from **Dr S C Sharma**, Director, AITR, Indore whenever needed. We take opportunity to convey my regards to the management of Acropolis Institute, Indore for extending academic and administrative support and providing me all necessary facilities for project to achieve our objectives.

We are grateful to **our parent** and **family members** who have always loved and supported us unconditionally. To all of them, we want to say, “Thank you”, for being the best family that one could ever have and without whom none of this would have been possible.

**Aeshna Jain(0827CS201017)**

# Executive Summary

***"House rent prediction system using machine learning and deep learning"***

This project is submitted to Rajiv Gandhi Proudyogiki Vishwavidhyalaya, Bhopal (MP), India for partial fulfillment of Bachelor of Engineering in Computer Science & Engineering branch under the sagacious guidance and vigilant supervision of **Prof. Priyanka Jangde.**

The project is based on Deep Learning, which is a sub field of machine learning, concerned with algorithms inspired by the structure and function of the brain called artificial neural networks. In this project, PixelLib is used which is a python library. PixelLib is a library used for easy implementation of semantic and instance segmentation of objects in images and videos. It makes it possible to train a custom segmentation model and also supports background editing of images and videos using few lines of code.

**Key words**: Image segmentation, PixelLib

*“Where the vision is one*
*year, cultivate flowers;*
*Where the vision is ten years,*
*cultivate trees;*
*Where the vision is eternity,*
*cultivate people.”*
*- Oriental Saying*

# List of Figures

# List of Abbreviations

***Abbr1:*** AI-Artificial Intelligence

***Abbr2:*** API-Application Programming Interface

***Abbr3:*** BHK-Bedroom Hall Kitchen

***Abbr4:*** CSV-Comma Separated Values

***Abbr5:*** GPU-Graphics Processing Unit

***Abbr6:*** HTML-Hypertext Markup Language

***Abbr7:*** JSON-JavaScript Object Notation

***Abbr8:*** *ML-Machine Learning*

***Abbr9:*** *OS-Operating System*

***Abbr10:*** *RAM-Random Access Memory*

***Abbr11:*** *ROI-Region of Interest*

***Abbr12:*** *SQL -Structured Query Language*

***Abbr13:*** *URL-Uniform Resource Locator*

# Table of Contents

# Chapter 1. Introduction

# Introduction

Apart from food, sleeping and water, houses are one of the most important needs. The demand for rented houses is growing, due to an increase in the population. Among the population, there are people who make houses their investment and property yet most of the people buying the house for their livelihood and shelter. The rent for a house is dependent primarily on its furnishing, i.e. whether it is fully furnished, partially furnished or unfurnished. House prices are a major contributor to the national economy, as the highest percentage of a person’s salary goes to renting a house for a living, people buy furniture for their houses, builders and contractors buy raw materials for building houses, and indirectly contribute to the national economy. It is a tedious and time-consuming task of collecting information about rents at different locations for major cities with hundreds of thousands of inhabitants. Hence, it’s important to establish an automated and online easy to access system that can predict price accurately.

The project uses image segmentation and regression model to get the desired output from the system. Here, Machine learning is used to train the machine for particular set of objects so that a system can be implemented for counting those objects by detecting and then recognizing them in real-time.

## 1.1 Overview

The project is based on image segmentation and regression testing on the trained model. A model is built that can predict the rent of a house in the Mumbai region by providing an image of the interior of the house and the needs of the user in relation to the house. The purpose of the system is to provide users with reliable rental prices for housing, in accordance with this model.

## 1.2 Background and Motivation

As the holy grail of computer vision research is to tell a story from a single image or a sequence of images, object detection and recognition has been studied for more than four decades. Significant efforts have been paid to develop representation schemes and algorithms aiming at recognizing generic objects in images taken under different imaging conditions (e.g., viewpoint, illumination, and occlusion).

Object Detection and recognition becomes a necessity when there is a need of automation, where the identification is done by machines instead of doing it manually for better performance and reliability. Image segmentation comes under the head object detection where each pixel value of an image is categorized to a particular class. As the population grows every year, rental rates rise as well. Studies have shown that for people who wish to sell their own homes and rent them out, the fluctuations in housing rental prices are a major concern. A system should therefore be put in place to predict the rental of a house according to its location, features, number of bedrooms, kitchen and dining room and floor area. This project focuses on developing a model which can predict house rent based on various factors and interior image of house.

## 1.3 Problem Statement and Objectives

Every now and then, thousands of people migrate to different cities for different purposes. In this case, getting a residence at an affordable rate is a complex task specially when the person is new to the place and unaware about the ongoing property rates. On the other hand, the people who wish to rent out their houses also find it difficult to estimate the correct rental price. Hence a system is created under this project to automate the house rent prediction by studying the particulars of the house to be rented.

Objective: The goal of this project is to create a model that can forecast the rent of a home in the Mumbai area using information about the user's wants and the interior of the home.

## 1.4 Scope of the Project

This system provides an easy-to-use interface for all users who wish to predict the rental price of their house/flats/apartment so that they can sell it.

- A lender may set an appropriate price for a house by using the prediction system.
- Users who have moved to Mumbai may look for rentals in particular places that are the price they will pay if they take up residence in a certain location with special facilities.
- By using this system, users will be able to compare prices and make up their own mind which items they are going to add or ignore.
- The future scope of this idea is to scale up with the websites which provide information about rented house in a particular society so that we can use their updated list of houses and update our dataset frequently to serve users with best and accurate value every time.

## 1.5 Team Organization

**Aeshna Jain:**

I studied about the topic and its scope and surveyed various research papers related to the object detection and the technology that is to be used.I also worked on the implementation of pandas and numpy library.

I worked on backend and assisted in creating the machine learning model. Documentation is also a part of the work done by me in this project.

## 1.6 Report Structure

The project ***House Rent Prediction System*** is primarily concerned with the **Object Detection and Regression model** and whole project report is categorized into five chapters.

Chapter 1: Introduction- introduces the background of the problem followed by rationale for the project undertaken. The chapter describes the objectives, scope and applications of the project. Further, the chapter gives the details of team members and their contribution in development of project which is then subsequently ended with report outline.

Chapter 2: Review of Literature- explores the work done in the area of Project undertaken and discusses the limitations of existing system and highlights the issues and challenges of project area. The chapter finally ends up with the requirement identification for present project work based on findings drawn from reviewed literature and end user interactions.

Chapter 3: Proposed System - starts with the project proposal based on requirement identified, followed by benefits of the project. The chapter also illustrate software engineering paradigm used along with different design representation. The chapter also includes and details of major modules of the project. Chapter also gives insights of different type of feasibility study carried out for the project undertaken. Later it gives details of the different deployment requirements for the developed project.

Chapter 4: Implementation - includes the details of different Technology/ Techniques/ Tools/ Programming Languages used in developing the Project. The chapter also includes the different user interface designed in project along with their functionality. Further it discusses the experiment results along with testing of the project. The chapter ends with evaluation of project on different parameters like accuracy and efficiency.

Chapter 5: Conclusion - Concludes with objective wise analysis of results and limitation of present work which is then followed by suggestions and recommendations for further improvement.

# Chapter 2. Review of Literature

## Review of Literature

Machine learning has become an important tool for data analysis and decision-making in various fields. In particular, Lasso regression has gained popularity as a powerful method for feature selection and regularization in linear regression models. Several studies have compared Lasso regression with other regularization techniques such as Ridge regression and Elastic Net. One of the main advantages of Lasso regression is its ability to perform feature selection, which is crucial in high-dimensional data analysis. In conclusion, Lasso regression is a powerful tool for feature selection and regularization in linear regression models. It has been used in various applications in different fields and has shown promising results.

Object Detection and Recognition is being studies for more than four decades now and significant efforts have been paid to develop representation schemes and algorithms aiming at recognizing generic objects in images taken under different imaging conditions. Within a limited scope of distinct objects, such as handwritten digits, fingerprints, faces, and road signs, substantial success has been achieved. Object recognition is also related to content-based image retrieval and multimedia indexing as a number of generic objects can be recognized. In addition, significant progress towards object categorization from images has been made in the recent years. Object recognition has also been studied extensively in psychology, computational neuroscience and cognitive science.

# 2.1 Preliminary Investigation

## 2.1.1 Current System

Still there are no online systems available that can rent price of any house/flat/apartment. Now also people use traditional approaches to consult any broker for detecting and researching for price of properties that they want to lend or buy it on rent. This system can be proven as a revolution to estimate precise value for houses/flats and apartments. This project uses image detection and classification conditions of houses which is not used by any software or company till now.

# 2.2 Requirement Identification and Analysis for Project

One of the major challenges in developing house rent prediction systems using machine learning and object detection is the lack of standardized data. Rental prices may vary significantly between different regions, and the factors that influence rental prices may also vary. Additionally, the accuracy of the predictions may be affected by the quality of the data and the availability of images of the houses.

- Creating a dataset from scratch for a machine learning project can be a challenging task. The main challenge is collecting and labeling a large enough set of data that is representative of the problem being solved. In the context of a house rent prediction system, this would require collecting rental data from various sources and manually labeling them with relevant features such as location, number of rooms, amenities, and furniture present in the house.
- Data collection can be a time-consuming process, and it may be difficult to ensure the data is of high quality and representative of the problem being solved. Additionally, labeling data can be a subjective process, and different people may have different interpretations of what constitutes a particular feature.
- Furthermore, the dataset may need to be cleaned and pre-processed to remove outliers, missing values, or irrelevant features. This can be a tedious and error-prone process that requires careful attention to detail.
- Web scraping is the process of automatically extracting data from websites. For the house rent prediction system, web scraping techniques can be used to collect data from various rental listing websites. The data collected may include information such as the location of the house, the number of rooms, the amenities, and the price.

- The web scraping process can be automated using programming tools such as Python, Beautiful Soup, and Selenium. These tools can be used to extract information from the HTML code of web pages and convert it into a structured format such as a CSV or JSON file.
- It is important to note that web scraping can be a complex process, as websites may use different structures and formats to present information. Additionally, web scraping can raise legal and ethical concerns, as it may violate website terms of service or infringe on user privacy. Therefore, it is important to use web scraping techniques responsibly and with caution.
- Cleaning a dataset created from web scraping is an essential step in preparing the data for machine learning. The web scraping process can introduce errors and inconsistencies into the data, such as missing values, duplicate entries, or incorrect formatting. These errors can negatively impact the performance of the machine learning model.
- To clean the dataset, several steps may be necessary. First, missing values and duplicates must be identified and removed. This can be done using statistical methods, such as calculating the mean or median of a feature and imputing missing values with that value. Duplicates can be removed by identifying records with identical feature values.
- Next, irrelevant features can be removed to reduce the dimensionality of the dataset. This can be done using feature selection methods, such as Principal Component Analysis (PCA) or Lasso regression. These methods can identify features that have little predictive power and remove them from the dataset.
- The dataset may need to be normalized or standardized to ensure that all features are on a similar scale. This can improve the performance of machine learning algorithms that are sensitive to differences in feature scales.
- Cleaning a dataset created from web scraping can be a time-consuming process that requires careful attention to detail. However, it is an essential step in preparing the data for machine learning and can greatly improve the performance of the final model.
- To detect furniture in images taken by a user, a computer vision system would need to perform several steps. First, the system would need to receive the user's image as input. This can be done through an interface that allows the user to upload an image or take a photo using a camera connected to the system.
- Next, the image would be processed using image segmentation to identify regions of the image that may contain furniture. This can be done using

algorithms such as k-means clustering or graph-based segmentation. These techniques divide the image into segments based on color, texture, or other visual features, which can help to identify regions that may contain furniture.

- Once the image has been segmented, the system would need to perform object detection to identify specific furniture items within the image. This can be done using machine learning algorithms, such as convolutional neural networks (CNNs), that have been trained on a large dataset of furniture images. The system would analyze each segment of the image and determine whether it contains a specific furniture item, such as a sofa or a table.
- Finally, the system would output the results of the object detection process, indicating which furniture items were detected in the user's image. The output could be displayed to the user, or used as input for further processing or analysis.
- Integrating the output from the number of furniture items detected with a dataset prepared from web scraping involves adding the count of furniture items detected to the existing data. This process can provide additional insights and help in better analysis of the dataset.
- The first step in integrating the output from furniture detection with a dataset prepared from web scraping is to extract the number of furniture items detected from the output. This could be done by analyzing the output of the furniture detection algorithm or by counting the number of furniture items identified in the image.
- Once the number of furniture items detected is extracted, it can be matched with the existing dataset using common fields or attributes. For example, if the dataset contains information about the furniture in a house, the number of items detected could be matched with the house address or unique identifier.
- The integrated dataset can then be used for further analysis or modeling, such as predicting the rent of a house based on the number of furniture items detected or analyzing the relationship between the number of furniture items detected and the house's location or size.
- Creating a lasso regression model to predict house rent based on the prepared dataset from web scraping and furniture detection using object detection involves using a statistical technique to identify the most significant variables or features that impact the house rent.
- The first step in creating the lasso regression model is to preprocess the dataset by cleaning the data, handling missing values, and scaling the features

appropriately. This step is crucial to ensure that the data is suitable for modeling and to prevent any biases or inaccuracies in the model.

- Next, the lasso regression algorithm is applied to the preprocessed dataset to identify the most important features that impact the house rent. The lasso regression algorithm works by adding a penalty term to the regression equation, which shrinks the coefficients of the less important variables to zero, effectively removing them from the model. This results in a more parsimonious model that is less prone to overfitting and provides better predictions.
- Overall, creating a lasso regression model to predict house rent based on the prepared dataset from web scraping and furniture detection using object detection is a powerful tool that can help in making informed decisions in real estate and related industries.

### 2.3.1 Conclusion

This chapter reviews the literature surveys that have been done during the research work. In conclusion, house rent prediction systems using machine learning and object detection have the potential to provide a valuable tool for both renters and landlords. While there are still challenges to overcome, the research in this field is promising and suggests that accurate rental price predictions can be achieved. Future research should focus on developing more accurate models and standardizing the data used to train them.

# Chapter 3. Proposed System

# Proposed System

## 3.1 The Proposal

This project aims at developing a system which can predict prices of rent of houses, flats and apartments in Mumbai based on conditions, location and amenities of properties. To achieve above stated goal, this project is developed by training machine learning model and uses deep learning concepts on images to extract information like type and condition of furniture present in the house.

This system is a web-based application which takes a form from user in which they feed necessary information like locations, amenities and components like number of rooms halls and washrooms, presence of balcony or not and its specific locations. This system is a web-based application which takes a form from user in which they feed necessary information like locations, amenities and components like number of rooms halls and washrooms, presence of balcony or not and its specific locations. This form also asks user to feed in necessary images of interior of houses/flats and apartments that is to be rented to predict their conditions.

## 3.2 Benefits of the Proposed System

The proposed house rent prediction system that uses object detection to detect furniture present in the house and web scraping to generate a dataset for training a machine learning model to predict rent using lasso regression offers several benefits. Some key points of explanation are:

- **Accurate predictions:** The use of object detection and web scraping helps to create a comprehensive and accurate dataset, which can improve the accuracy of the rent predictions. The lasso regression model can identify the most important variables that impact the rent, resulting in more accurate predictions.
- **Efficient and automated**: The proposed system can automate the process of rent prediction, which can save time and resources for landlords, property managers, and real estate agents. By using object detection and web scraping, the system can gather data quickly and efficiently.

- **Customizable:** The system can be customized to suit specific needs and preferences. For example, users can input their preferred location, number of rooms, and other criteria to generate personalized rent predictions.
- **Insights for decision-making:** The system can provide valuable insights for decision-making in the real estate industry. For example, by analyzing the relationship between the number of furniture items and the rent, the system can identify trends and patterns that can help in making informed decisions.
- **Scalable:** The system can be scaled to handle large volumes of data, making it suitable for use in large-scale real estate operations.

Overall, the proposed house rent prediction system offers several benefits, including improved accuracy, efficiency, customizability, insights for decision-making, and scalability.

## 3.3 Feasibility Study

A feasibility study is an important step in determining whether the proposed house rent prediction system is viable and practical. Here is an overview of the feasibility study for the proposed system:

- **Technical feasibility:** The proposed system is technically feasible as it uses established technologies such as object detection, web scraping, and machine learning algorithms. The use of these technologies has been proven to be effective in similar applications.
- **Economic feasibility:** The economic feasibility of the system depends on factors such as the cost of hardware, software, and labor. The system requires a computer, a camera or other device for image capture, and software for object detection, web scraping, and machine learning. The cost of these components can be significant, but the potential cost savings from automated rent prediction could outweigh the initial investment.
- **Legal feasibility:** The system must comply with relevant legal and ethical standards, including data privacy and security regulations. The use of personal information in the dataset and the handling of sensitive data must be addressed.
- **Operational feasibility:** The system's operational feasibility depends on the availability of necessary resources, such as trained personnel, sufficient data storage, and reliable internet connectivity. The use of automated tools such as web scraping and object detection can streamline operations and make the system more efficient.

- **Schedule feasibility:** The development and implementation of the system will require time and resources. The project must be appropriately planned and scheduled to ensure timely delivery.

Based on the feasibility study, the proposed house rent prediction system is technically feasible, economically viable, legally compliant, operationally feasible, and can be delivered within a reasonable schedule.

### 3.3.1 Technical

The proposed house rent prediction system is technically feasible, as all the required technologies and resources for its development and implementation are well-established. Object detection algorithms such as PixelLib can detect furniture present in images, web scraping tools can be used to collect data on rental properties, machine learning algorithms like Lasso Regression can predict rent based on the collected data, and database management tools can store and manage the data. However, the main challenges are related to data quality, availability, and computational resources, which can be addressed through careful planning and optimization. Overall, the technical feasibility study indicates that the proposed system is a practical solution for automated rent prediction based on image data and machine learning algorithms.

### 3.3.2 Economical

The economic feasibility study of the proposed house rent prediction system aims to determine the financial viability of the project. The study examines the costs associated with the development and implementation of the system and compares them to the potential benefits that the system can provide.

The main costs of the system include the development and implementation of the software, data collection and processing, computational resources, and maintenance costs. The costs associated with software development and implementation include the cost of hiring developers, purchasing necessary software and hardware, and any other development-related expenses. Data collection and processing costs include the cost of web scraping tools and the time and effort required to collect and process the data.

On the other hand, the potential benefits of the system include increased accuracy in rent prediction, reduced human error, improved efficiency in the rental process, and reduced operational costs. The system can automate the process of rent prediction, allowing real estate businesses to make faster and more accurate decisions, leading to increased profitability and productivity.

The economic feasibility study indicates that the proposed system is economically feasible, as the potential benefits outweigh the costs. The system can help real estate businesses streamline their operations, save time and resources, and increase profitability. Therefore, the proposed house rent prediction system is a practical and cost-effective solution for automated rent prediction based on image data and machine learning algorithms.

### 3.3.3 Operational

The operational feasibility study of the proposed house rent prediction system aims to determine the system's ability to meet its objectives and whether it can be integrated into the current workflow of real estate businesses.

The operational feasibility study examines the current rental process of the businesses and identifies any potential bottlenecks, gaps, or limitations that the proposed system can address. It also assesses the availability of resources required for the system, including personnel, technology, and data.

The study also considers the training needs for the employees who will use the system and assesses their ability to adapt to the new technology. Additionally, it evaluates the system's usability, reliability, and performance.

The operational feasibility study indicates that the proposed house rent prediction system is operationally feasible. The system can easily integrate into the current workflow of real estate businesses, and it can address potential bottlenecks in the rental process by automating the rent prediction process. The system's usability, reliability, and performance can be enhanced by designing a user-friendly interface, implementing robust testing procedures, and ensuring the availability of required computational resources.

Moreover, the system's integration with existing technologies, such as web scraping and machine learning algorithms, ensures that the system can meet its objectives and provide accurate rent predictions based on image data. The training needs for the employees can be addressed by providing proper training and support materials.

Therefore, the operational feasibility study concludes that the proposed house rent prediction system is operationally feasible and can provide real estate businesses with an efficient and accurate automated rent prediction solution.

## 3.4 Design Representation

### 3.4.1 Use Case Diagram

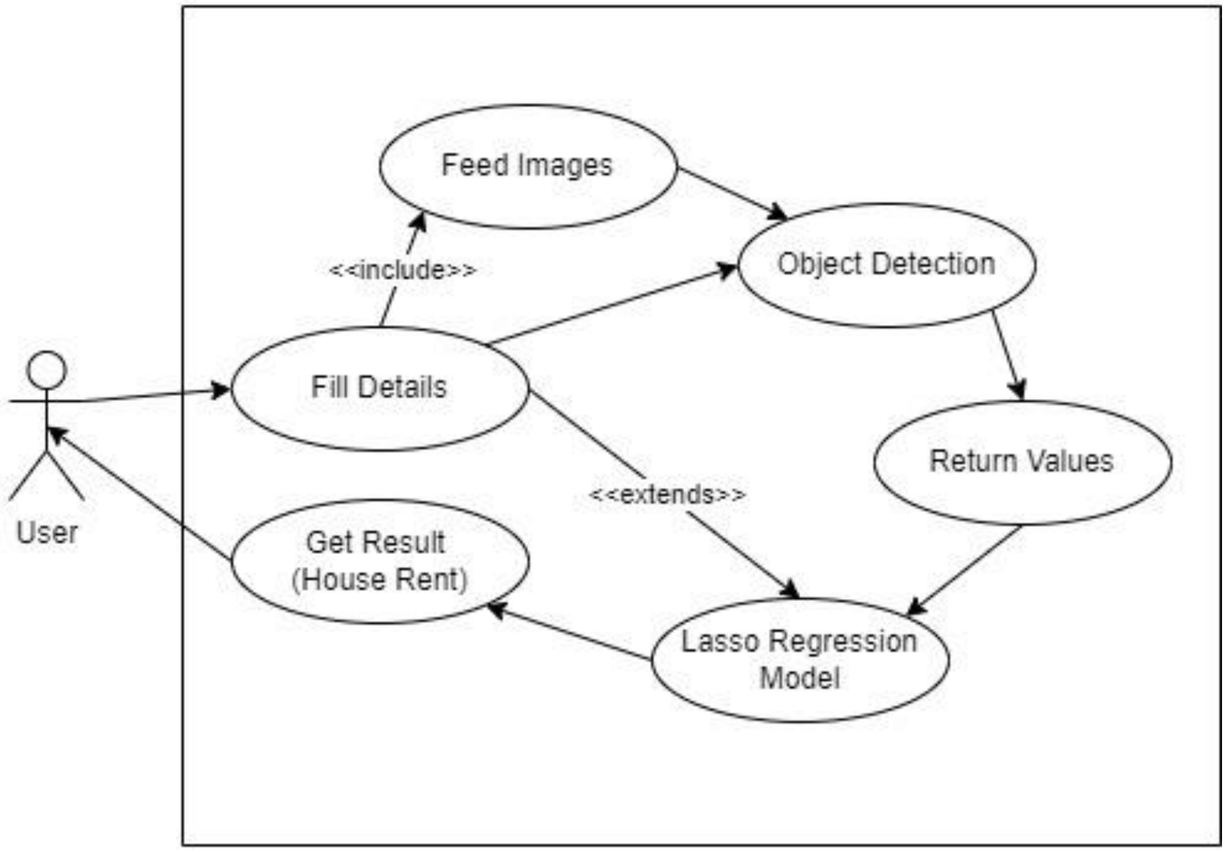


**Figure 3-1: Use Case diagram**

## 3.4.2 Sequence Diagram

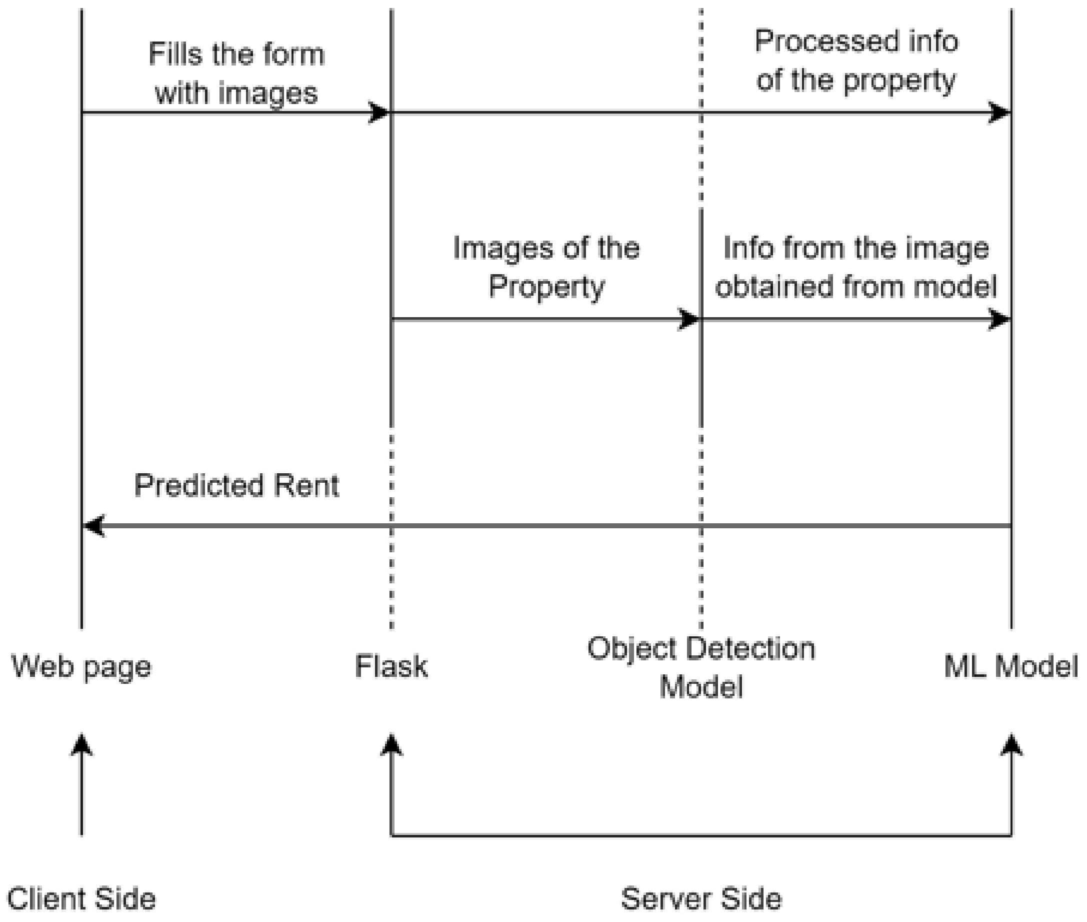


**Figure 3-2 sequence Diagram**

## 3.4.3 Data Flow Diagrams

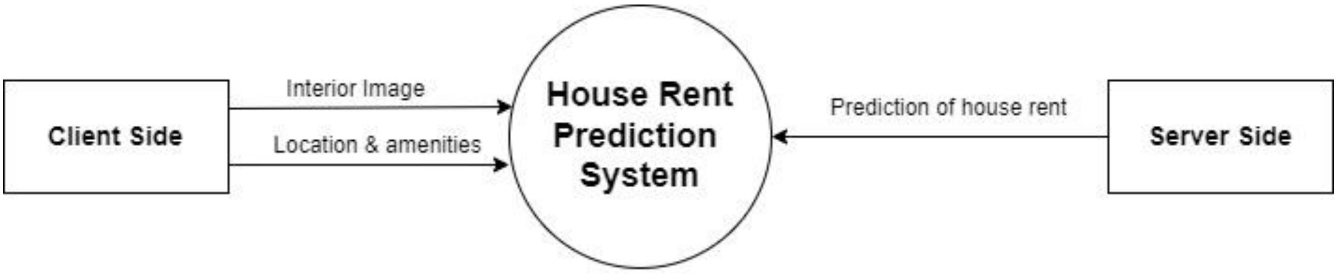


Figure 3-3 Data Flow Diagram Level 0

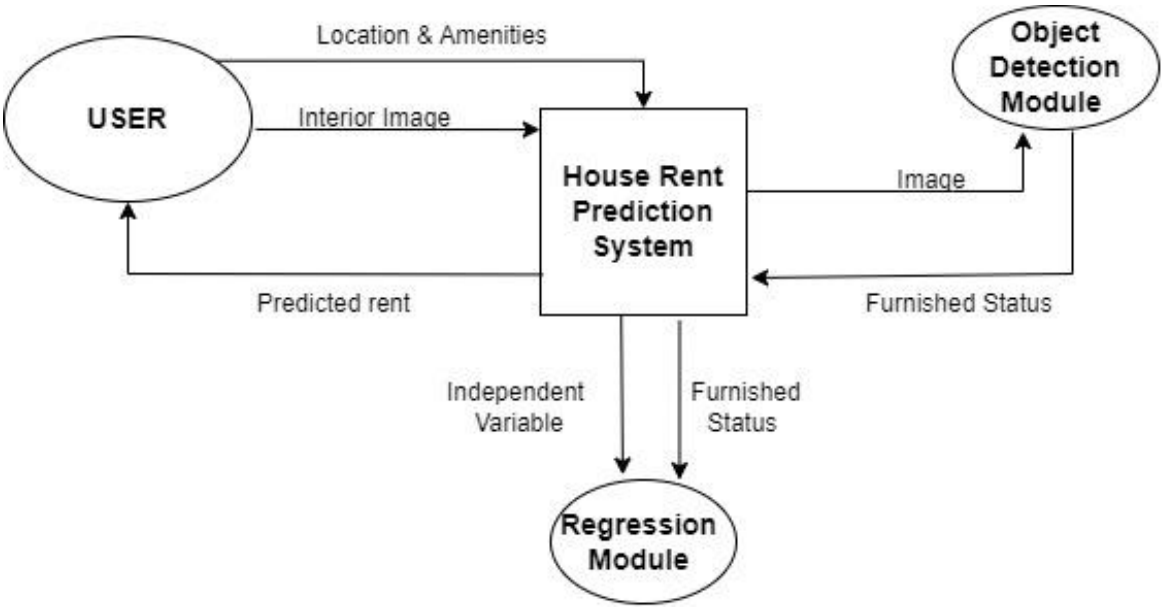


Figure 3-4 Data Flow Diagram Level 1

## 3.4.4 Object Detection Using Pixellib

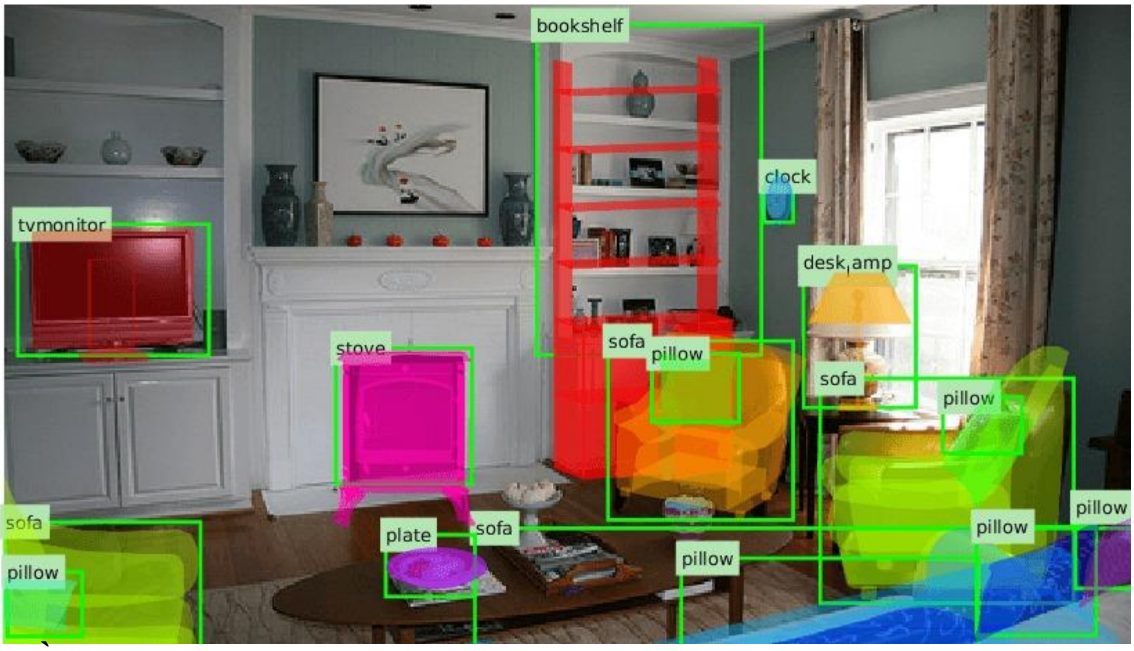


**Figure 3-5: Image segmentation Using Pixellib**

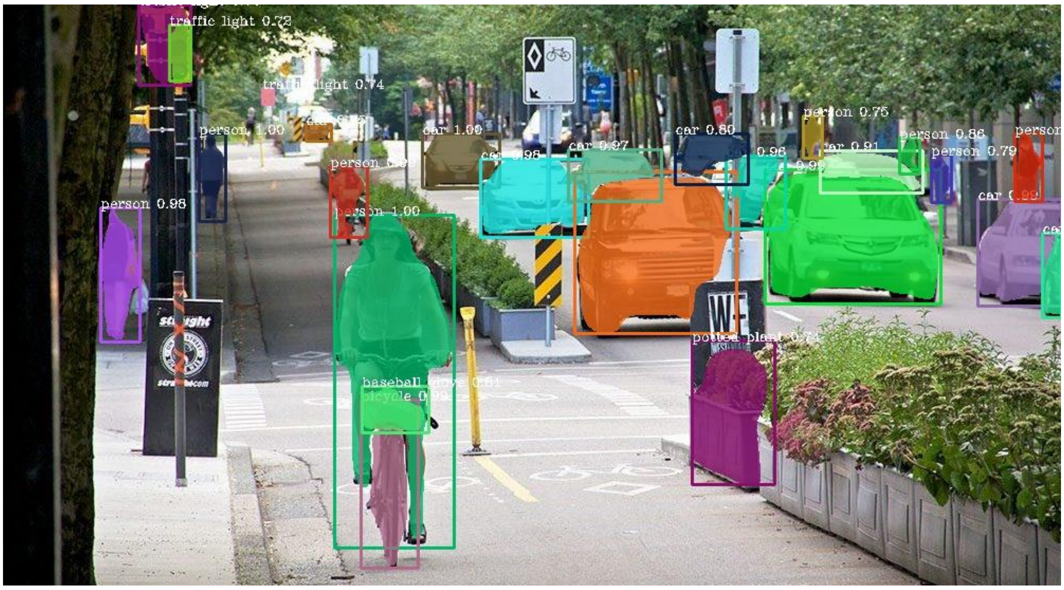


**Figure 3-6: Image segmentation Using Pixellib**

### 3.4.5 Dataset Structure

Dataset is the training and testing data on which our machine learns and test its accuracy. For this we split the data into 80:20 ratio i.e. 80% data is utilized for training and 20% of data is used for testing purpose. This dataset is prepared using web-scraping of magicbricks.com site where information of rental properties are specified.

| Name | Data Type | Description |
| --- | --- | --- |
| **Location** | String | The name of the residential society where the property is located |
| **Rent** | integer | The monthly rent of the property in INR |
| **Carpet area** | integer | The size of the property in square feet |
| **Per square feet rate** | integer | The per square feet rate of the property in INR |
| **BHK** | integer | The number of bedrooms, hall, and kitchen in the property |
| **Furnished status** | 0,1,0.5 | Whether the property is furnished, unfurnished or semi-furnished |
| **Amenities** | Binary type | Amenities include availability of gym, swimming pool and power backup. |

**Figure 3-7: DataSet Structure**

## 3.5 Deployment Requirements

The deployment hardware and software requirements for the proposed house rent prediction system include the following:

### 3.5.1 Hardware

- Processor: Intel Core i5 or higher
- RAM: 8 GB or higher
- Hard Disk Space: 1 TB or higher
- Graphics Card: NVIDIA or AMD with a minimum of 2 GB memory
- Internet Connection: Broadband or higher

### 3.5.2 Software

- Operating System: Windows 10 or Ubuntu 18.04 or higher
- Python 3.7 or higher
- Pixellib Library
- Scikit-learn Library
- BeautifulSoup Library
- Django Framework
- MySQL Database

The software components must be installed and configured properly for the system to function correctly. The Python environment must have all the necessary packages installed, and the MySQL database must be set up and configured to store the scraped data, furniture images, and trained machine learning model.

The proposed system also requires a web server, such as Apache or Nginx, to host the web application. Additionally, a domain name and SSL certificate are required to ensure secure communication between the client and server.

Overall, the hardware and software requirements for the proposed house rent prediction system require careful consideration to ensure that the system functions correctly and provides users with a satisfactory experience.

# Chapter 4. Implementation

## Implementation

The implementation process would involve the following steps:

- Develop the code for data preprocessing, object detection, feature extraction, and model training using the chosen programming language and frameworks.
- Train the machine learning model on the preprocessed dataset and evaluate its performance using appropriate metrics.
- Develop the web-based application using a suitable web framework and deploy it to a cloud service provider.
- Test the application and refine it as necessary based on user feedback.
- Monitor the performance of the application and update the machine learning model periodically as new data becomes available.

Overall, implementing this model requires a thorough understanding of machine learning, web development, and cloud computing, as well as expertise in the specific technologies used.

## 4.1 Technique Used

### 4.1.1 Web Scrapping

Web scraping is the process of extracting data from websites using automated software. It involves downloading and parsing the HTML or XML content of web pages to extract the desired information. In this house rent prediction system, web scraping will be used to collect data from various real estate websites to create a dataset for training the machine learning model.

Web scraping can be performed using various tools and libraries such as Beautiful Soup, Scrapy, Selenium, and many more. These tools allow developers to extract data from websites in a structured manner, making it easy to collect the required information.

In this house rent prediction system, web scraping will be used to extract relevant information from real estate websites such as property features, location, rent, and images. This data will be used to create a dataset for training the machine learning model. Once the dataset is prepared, it will be used to train the machine learning model to predict the rent of a house based on various factors such as location, number of rooms, and property features.

Web scraping can be a complex process, and it is important to ensure that it is performed ethically and legally. It is important to respect the website's terms of service and not to overload the server with too many requests. Proper data cleaning and processing should also be performed on the extracted data to ensure that it is accurate and usable.

### 4.1.2 Lasso Regression

Lasso regression is a type of linear regression that is used for feature selection and regularization. It can be used to predict the rent of a house based on various factors such as location, number of rooms, and property features.

In this house rent prediction system, lasso regression will be used in conjunction with object detection techniques to predict the rent of a house. The object detection technique will detect the furniture present in the input images and use this information as an additional feature in the machine learning model.

The machine learning model will be trained on a dataset that includes the location, number of rooms, property features, and furniture present in the house. Lasso regression will be used to select the most important features from this dataset and to regularize the coefficients of the model.

Once the machine learning model is trained, it can take input images of a house as well as other relevant information such as location and number of rooms, and predict the rent based on the learned coefficients and selected features. The object detection technique will be used to extract information about the furniture present in the house, which will be used as an additional feature in the prediction.

### 4.1.3 Image Segmentation

Image segmentation is a technique of dividing an image into multiple segments or regions based on similar characteristics such as color, texture, or intensity. In this model, image segmentation is used to detect furniture in the input images provided by the user. The input image is first divided into multiple segments, and then the segments are analyzed to identify furniture using object detection techniques.

The PixelLib library is used for object detection, which has pre-trained models for detecting common objects like chairs, tables, sofas, etc. The identified furniture is then used to augment the dataset created through web scraping. This augmented dataset is then used to train the Lasso regression model to predict the rent of the house.

In summary, image segmentation is a crucial step in detecting furniture from input images and augmenting the dataset. It helps in improving the accuracy of the Lasso regression model and provides a more reliable rent prediction.

## 4.2 Tools Used

### 4.2.1 Beautiful Soup

Beautiful Soup is a Python library that is used for web scraping purposes. It is used to extract the data from HTML and XML files by parsing the content of the files. Beautiful Soup provides a simple and intuitive interface for working with HTML and XML files, making it easy to extract the required data.

In this house rent prediction system, Beautiful Soup will be used to extract data from various real estate websites. The library will be used to parse the HTML files of the web pages and extract the relevant information such as property features, location, rent, and images. The data extracted by Beautiful Soup will be used to create a dataset for training the machine learning model.

Beautiful Soup can be installed using pip or conda package manager.

### 4.2.2 Pixellib

Pixellib is a Python library used for object detection and segmentation in images and videos. It offers pre-trained models for detecting common objects like cars, people, furniture, etc. In this model, Pixellib is used for detecting furniture in the input images provided by the user.

First, the input image is passed to Pixellib's object detection function, which identifies the regions in the image that contain objects. Then, using a pre-trained model, Pixellib can detect common furniture objects like chairs, tables, and sofas in these regions. Once the furniture objects are detected, they are segmented from the image, and their characteristics are extracted and used to augment the dataset created through web scraping.

### 4.2.2 Scikit learn

Scikit-learn is a machine learning library in Python that provides various algorithms and tools for classification, regression, clustering, and dimensionality reduction. In this model, scikit-learn will be used for implementing the Lasso regression algorithm.

Lasso regression is a linear regression algorithm used to perform variable selection by adding a penalty term to the loss function of the regression. The penalty term is a multiple of the absolute value of the regression coefficients. This penalty term shrinks the coefficients of irrelevant features to zero, thereby performing feature selection.

In the proposed house rent prediction system, scikit-learn will be used to implement the Lasso regression algorithm on the dataset generated from web scraping and furniture detection. The Lasso regression algorithm will learn the relationship between the input features and the rent, and predict the rent based on new input features. The scikit-learn library provides various tools for model evaluation and hyperparameter tuning, which can be used to improve the accuracy of the model.

### 4.2.4 Pandas

Pandas is an open-source data manipulation and analysis library in Python. It provides data structures for efficiently storing and manipulating tabular data, such as data frames and series.

In the house rent prediction system, Pandas can be used for various data-related tasks, such as:

- **Data cleaning:** Pandas provides various functions to clean and preprocess data, such as handling missing values, removing duplicates, and converting data types.
- **Data manipulation:** Pandas provides functions for filtering, sorting, grouping, and aggregating data.
- **Data analysis:** Pandas provides statistical and mathematical functions to analyze data, such as mean, median, standard deviation, and correlation.

In the context of the system, Pandas can be used for cleaning and preprocessing the data scraped from websites to generate a dataset, as well as manipulating and analyzing the dataset to prepare it for training the machine learning model. Additionally, Pandas can be used to load the trained model and apply it to new data to predict the rent.

## 4.3 Language Used

Python language is used in the system due to the following Characterstics :

**Simple:**

Python is a simple and minimalistic language. Reading a good Python program feels almost like reading English (but very strict English!). This pseudo-code nature of Python is one of its greatest strengths. It allows you to concentrate on the solution to the problem rather than the syntax i.e. the language itself.

**Free and Open Source:**

Python is an example of a FLOSS (Free/Libre and OpenSource Software). In simple terms, you can freely distribute copies of this software, read the software's source code, make changes to it, use pieces of it in new free programs, and that you know you can do these things. FLOSS is based on the concept of a community which shares knowledge. This is one of the reasons why Python is so good - it has been created and improved by a community who just want to see a better Python.

**Object Oriented:**

Python supports procedure-oriented programming as well as object-oriented programming. In procedure-oriented languages, the program is built around procedures or functions which are nothing but reusable pieces of programs. In object-oriented languages, the program is built around objects which combine data and functionality. Python has a very powerful but simple way of doing object-oriented programming, especially, when compared to languages like C++ or Java.

**Extensive Libraries:**

The Python Standard Library is huge indeed. It can help you do various things involving regular expressions, documentation generation, unit testing, threading, databases, web browsers, CGI, ftp, email, XML,

## 4.4 Screenshots

The Following are the screenshots of the result of the project:

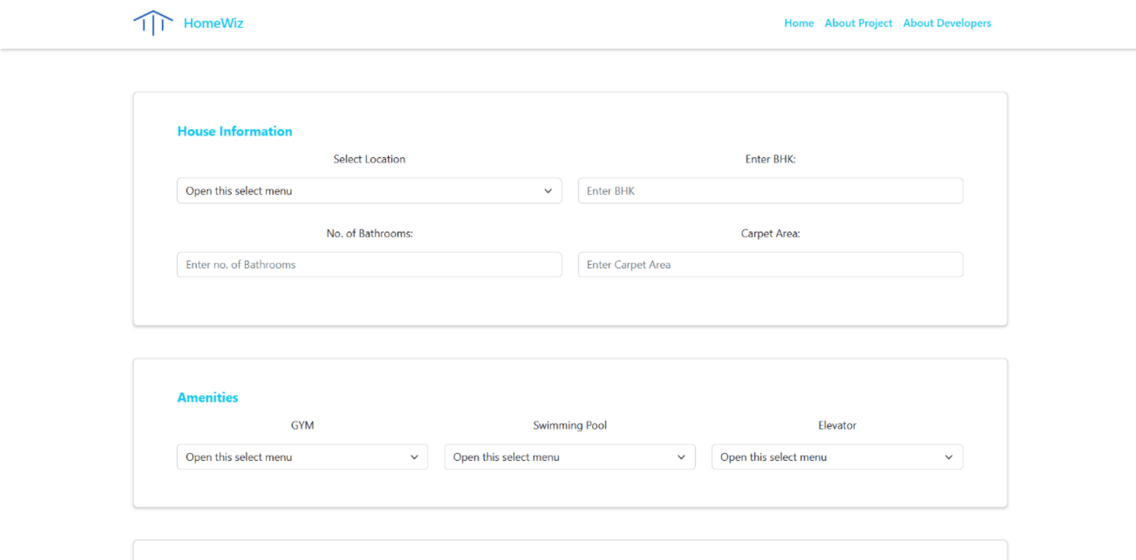


**Figure 4-1: Screenshot 1**

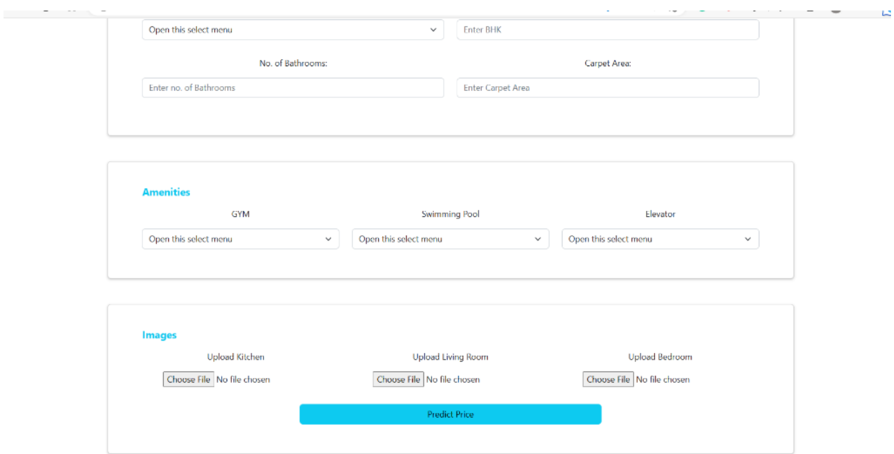


**Figure 4-2: Screenshot 2**

## 4.5 Testing

Testing is the process of evaluation of a system to detect differences between given input and expected output and also to assess the feature of the system. Testing assesses the quality of the product. It is a process that is done during the development process.

### 4.5.1 Strategy Used

Tests can be conducted based on two approaches –

- Functionality testing
- Implementation testing

The texting method used here is Black Box Testing. It is carried out to test functionality of the program. It is also called 'Behavioral' testing. The tester in this case, has a set of input values and respective desired results. On providing input, if the output matches with the desired results, the program is tested 'ok', and problematic otherwise.
The house rent prediction system has undergone various testing methods to ensure its effectiveness and accuracy. One of the testing methods used is unit testing, which involves testing individual components of the system to ensure they are working correctly. Integration testing was also performed to verify that the different components of the system work seamlessly together.

In addition, performance testing was carried out to assess the system's ability to handle a large volume of data and processing demands. This was done to ensure that the system does not become slow or unresponsive when handling multiple requests from users.

Finally, user acceptance testing was conducted to ensure that the system meets the expectations and requirements of end-users. This involved a group of users interacting with the system to verify that it is easy to use, user-friendly, and provides accurate and reliable results.

## 4.5.2 Results

**Sample Input:**

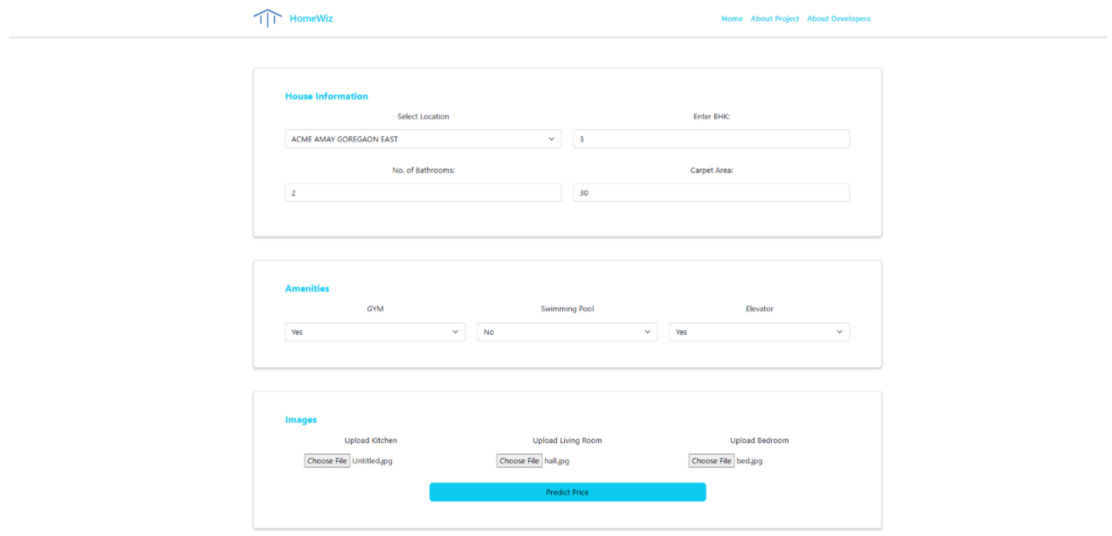


**Figure 4-3: Sample Input**

**OUTPUT:**

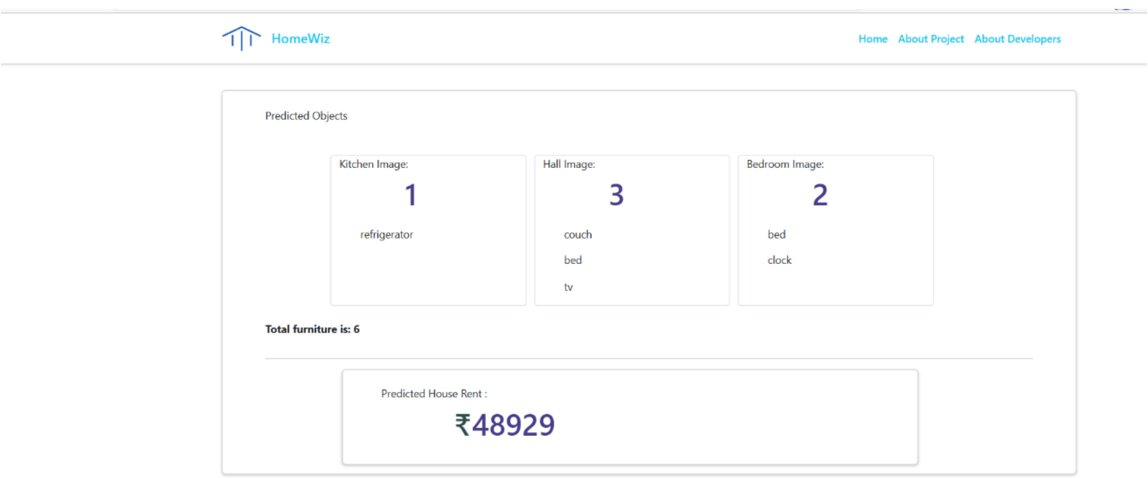


**Figure 4-4: Output**

# Chapter 5. Conclusion

# Conclusion

## 5.1 Conclusion

The most significant need for human survival next to food is shelter, which on a daily basis becomes more in demand as the population grows. There are fluctuations in house prices, which is why the choice of price for a rental home is so important. This system aims to make it easier and more secure for them to obtain rents on their flat, apartment or house. It can therefore affect their choice in a positive way. This system is intended to avoid devious acts of fraud being carried out by contractors. In order to eliminate manual effort and time when looking for houses in the area of our description, it is very useful to use a prediction system based on machine learning. The aim of the project was to predict the rent of the houses accurately which can be helpful for both – the one who wants to rent their property and the one who wants to purchase the house on rent. The project uses PixelLib to identify the objects and regression model to get the output.

## 5.2 Limitations of the Work

- Since no task can be 100% perfect. The same applies to this project as the predicted house rent is not 100% accurate.
- The models that we are using for identifying the objects are pre-trained models. So, if we want to train our own model, it takes a lot of time and processing.
- The system is created such that it can predict the house rent for the properties situated in Mumbai only. Also the user needs to enter the correct details in the form in order to get the rent predicted correctly.
- One of the limitations of this model is that it heavily relies on the dataset generated from web scraping. Since the dataset is generated using a specific set of websites, it may not represent the entire housing market. Additionally, the data may become outdated over time, which could result in inaccurate predictions. The model may also face limitations in predicting rental prices for houses that have unique features or furnishings that are not present in the dataset. Therefore, the model should be updated periodically with new data to ensure accurate predictions.

- Furthermore, the accuracy of the model's predictions depends on the quality of the object detection algorithm used to identify furniture in the images. If the algorithm is not accurate, the model's predictions may be affected. Similarly, the accuracy of the Lasso regression model also depends on the quality of the input data.

- However, the accuracy and reliability of the dataset depend on the quality of the sources from which it is scraped. The scraped data may contain outdated or incorrect information, leading to errors in the model's predictions. Moreover, as the dataset is older, it may not reflect the current real estate market trends, and hence, the predictions may not be accurate.

## 5.3 Suggestion and Recommendations for Future Work

The future scope of this idea is to scale up with the websites which provide information about rented house in a particular society so that we can use their updated list of houses and update our dataset frequently to serve users with best and accurate value every time.

•The Model would be trained for detecting a greater number of objects.

•The scope of the project can be extended to more cities other than Mumbai.

•In future, the system can be modified in such a way that enables the customers to rent their houses and integrate it with a payment gateway as well.

# Bibliography


- Mr. Rushikesh Naikare, Mr. Girish Gahandule, Mr. Akash Dumbre, Mr. Kaushal Agrawal, Prof. Chaitanya Manka, "House Planning and Price Prediction System using Machine Learning", International Engineering Research Journal, Vol:3 Issue: 3, 2019.
- Aswin Sivam Ravikumar, Thibaut Lust, "Real Estate Price Prediction Using Machine Learning", 2016.
- H. Wu et al., “Influence factors and regression model of urban housing prices based on internet open access data,” Sustain., vol. 10, no. 5, pp. 1–17, 2018, doi: 10.3390/su10051676.
- Ayush Varma, Abhijit Sarma, Sagar Doshi, Rohini Nair - “Housing Price Prediction Using Machine Learning and Neural Networks” 2018, IEEE.
  5.https://www.irejournals.com/formatedpaper/1702692.pdf
- Anirudh Kaushal, Achyut Shankar, House Price Prediction Using Machine Learning, 2021
- CH. Raga Madhuri, G. Anuradha, M. Vani Pujitha -” House Price Prediction Using Regression Techniques: A Comparative Study” 2019 in (ICSSS),IEEE.

# Log Report

# Guide Interaction Sheet

| | | | | | |
|---|---|---|---|---|---|
| **Computer Science and Engineering Department** | | | | | |
| **PROJECT - LOG BOOK** | | | | | |
| **Project Title:** | | House rent prediction system using ML & DL | | | |
| **Team Name:** | | Bots | **Team Id:** | **CS1-T1** | |
| **Coordinator 1 Name:** | | | **Semester:** | **VI** | |
| **Coordinator 2 Name:** | | Prof. Preeti Shukla | **Section:** | **CS1** | |
| **Technology:** | | | **Domain:** | | |
| **S No** | **Enrollment** | **Team Member Name** | **Mobile Number** | **Email Id** | **Role** |
| 1 | | | | | |
| 2 | | | | | |
| 3 | | | | | |
| 4 | | | | | |
| **S No** | **Meeting Date** | **Summary of Work & Discussion** | **Member Present** | **Remarks/ Suggestions Given** | **Guide Sign** |
| 1 | | | | | |
| 2 | | | | | |
| 3 | | | | | |
| 4 | | | | | |
| 5 | | | | | |
| 6 | | | | | |
| 7 | | | | | |
| 8 | | | | | |
| 9 | | | | | |
| 10 | | | | | |
| 11 | | | | | |
| 12 | | | | | |
| **S No** | **Due Date** | **Particular** | **Submission Date** | **Observations** | **Guide Sign** |
| 1 | | Team Formation | | | |
| 2 | | Project Title | | | |
| 3 | | Synopsis | | | |
| 4 | | Synopsis Presentation | | | |
| 5 | | Design Diagrams | | | |
| 6 | | Paper Publication | | | |
| 7 | | Presentation-II | | | |
| 8 | | Video | | | |
| 9 | | Technical Poster | | | |
| 10 | | Report | | | |

# Source Code

## 1. app.py

```
# pip install torch
# pip install torchvision
# pip install pixellib
# pip install pycocotools

import os
import pickle
import pandas as pd
from flask import Flask, request, render_template, redirect, url_for, flash
from werkzeug.utils import secure_filename
from sklearn import linear_model

import pixellib
from pixellib.torchbackend.instance import instanceSegmentation

UPLOAD_FOLDER = 'upload'
ALLOWED_EXTENSIONS = {'png', 'jpg', 'jpeg'}

app = Flask(__name__)
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER

ins = instanceSegmentation()
ins.load_model("pointrend_resnet50.pkl")
target_classes =
ins.select_target_classes(couch=True,dining_table=True,bench=True,refrigerator=True,bed=
True,oven=True,microwave=True,toaster=True,clock=True,tv=True)

#loading lasso model
data = pd.read_csv('Cleaned_goregaon.csv')
pipe = pickle.load(open('LassoModel.pkl', 'rb'))


def allowed_file(filename):
```

```
    return '.' in filename and \
        filename.rsplit('.', 1)[1].lower() in ALLOWED_EXTENSIONS



@app.route('/', methods=['GET', 'POST'])
def index():
    societies = sorted(data['Society'].unique())
    locations = sorted(data['Loc'].unique())
    return render_template('index.html', societies=societies, locations=locations)
    #return render_template('index.html')



@app.route('/predict', methods=['GET', 'POST'])
def predict():



    if request.method == 'POST':
        gym = request.form['gym']
        lift= request.form['lift']
        swimming_pool = request.form['swimming_pool']

        kitchen = request.files['kitchen']
        hall = request.files['hall']
        bedroom = request.files['bedroom']

        # if user does not select file, browser also
        # submit an empty part without filename
        if kitchen.filename == '':
            flash('No selected file')
            return redirect(request.url)
        if kitchen and allowed_file(kitchen.filename):
            filenameOfKitchen = secure_filename(kitchen.filename)
            pathOfKitchenFile = os.path.join(app.config['UPLOAD_FOLDER'],
filenameOfKitchen)
            print(pathOfKitchenFile)
            kitchen.save(pathOfKitchenFile)
```

```
    if hall.filename == '':
      flash('No selected file')
      return redirect(request.url)
    if hall and allowed_file(hall.filename):
      filenameOfHall = secure_filename(hall.filename)
      pathOfHallFile = os.path.join(app.config['UPLOAD_FOLDER'], filenameOfHall)
      hall.save(pathOfHallFile)

    if bedroom.filename == '':
      flash('No selected file')
      return redirect(request.url)
    if bedroom and allowed_file(bedroom.filename):
      filenameOfBedroom = secure_filename(bedroom.filename)
      pathOfBedroomFile = os.path.join(app.config['UPLOAD_FOLDER'],
filenameOfBedroom)
      bedroom.save(pathOfBedroomFile)

    print("All pics Saved")

    resultsOfKitchenImage, output1 = ins.segmentImage("{}".format(pathOfKitchenFile),
segment_target_classes= target_classes,show_bboxes=True,
output_image_name="{}".format(pathOfKitchenFile))
    li1 = resultsOfKitchenImage['class_names']
    print(type(li1))
    countk=len(li1)
    resultsOfHallImage, output2 =
ins.segmentImage("{}".format(pathOfHallFile),segment_target_classes= target_classes,
show_bboxes=True, output_image_name="{}".format(pathOfHallFile))
    li2 = resultsOfHallImage['class_names']
    counth=len(li2)
    resultsOfBedroomImage, output3 =
ins.segmentImage("{}".format(pathOfBedroomFile),segment_target_classes= target_classes,
show_bboxes=True, output_image_name="{}".format(pathOfBedroomFile))
    li3 = resultsOfBedroomImage['class_names']
    countb=len(li3)
    total_furniture=counth+countk+countb
    furni=0
```

```
        if total_furniture >15:
            furni=1
        elif total_furniture<=15 and total_furniture>=2:
            furni=0.5
        elif total_furniture==0:
            furni=0



        if request.method == 'POST':
            location = request.form['Location']
            bhk = request.form['bhk']
            sqft = request.form['sqft']
            bath = request.form['bath']
            # furni = request.form['Furnished']

            powe = 1
            # lift = 1
            # swim = 1
            # gym = 0
            input = pd.DataFrame([[location, sqft,
bhk,bath,furni,lift,swimming_pool,gym,powe]],columns = ['Loc','Carpet
Area','BHK','Bath','Furnished','Lift','Swimming Pool','Gym','Power Back Up'])
            prediction = pipe.predict(input)[0]
            print(prediction)
        return render_template('predict.html', li1 = li1, li2 = li2, li3 = li3,
swimming_pool=swimming_pool,lift=lift,gym=gym,countk=countk,counth=counth,countb=c
ountb,total_furniture=total_furniture,prediction=prediction)



    return render_template('predict.html')

if __name__ == '__main__':
    app.run(debug=True)
```

2. Demo.ipynb (for lasso regression implementation)

```
import pandas as pd
import numpy as np
data = pd.read_csv('Goregaon_extracted.csv')
```

```
data
```

```
data = data.drop(columns=['Unnamed: 0'], axis = 1)
```

```
data.info()
```

```
data
```

```
data.describe()
```

```
data.isnull().sum()
```

```
data = data.drop(columns =['Balcony','Description'], axis = 1)
```

```
data
```

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

```
data['Society'] = data['Society'].apply(lambda x: str(x).strip())
society_count = data['Society'].value_counts()
```

```
society_count

society_count_nan = society_count[society_count > 100]
society_count_nan

data['Society'] = data['Society'].apply(lambda x: 'Other' if x in society_count_nan else
x)
data['Society'] = data['Society'].apply(lambda x: x.upper())

data['Location'] = data['Location'].apply(lambda x: str(x).strip())
data['Location'] = data['Location'].apply(lambda x: x.upper())

power = []
for li in data['Amenities']:
    if 'Power Back Up' in li:
        power.append(1)
    else:
        power.append(0)



data['Power Back Up'] = power

data['Amenities'] = data['Amenities'].apply(lambda x: x.replace("'Power Back
Up',",""))

lift = []
for li in data['Amenities']:
    if 'Lift' in li:
        lift.append(1)
    else:
        lift.append(0)
data['Lift'] = lift
data['Amenities'] = data['Amenities'].apply(lambda x: x.replace("'Lift'",""))
```

```
data['Amenities'] = data['Amenities'].apply(lambda x: x.replace(","," "))

data['Amenities'] = data['Amenities'].apply(lambda x: x.replace("]",""))

data['Amenities'] = data['Amenities'].apply(lambda x: x.strip())
data['Amenities']

Swim = []
for li in data['Amenities']:
    if 'Swimming Pool' in li:
        Swim.append(1)
    else:
        Swim.append(0)
data['Swimming Pool'] = Swim
data['Amenities'] = data['Amenities'].apply(lambda x: x.replace("'Swimming
Pool'",""))
data['Amenities'] = data['Amenities'].apply(lambda x: x.strip())

gym = []
for li in data['Amenities']:
    if 'Gymnasium' in li:
        gym.append(1)
    else:
        gym.append(0)
data['Gym'] = gym
data['Amenities'] = data['Amenities'].apply(lambda x: x.replace("'Gymnasium'",""))
data['Amenities'] = data['Amenities'].apply(lambda x: x.strip())

data['ImpInfo'].value_counts()

data['ImpInfo'] = data['ImpInfo'].apply(lambda x: x.replace("[",""))
data['ImpInfo'] = data['ImpInfo'].apply(lambda x: x.replace("]",""))
data['ImpInfo'] = data['ImpInfo'].apply(lambda x: x.replace("1Bath",""))
```

```
data['ImpInfo'] = data['ImpInfo'].apply(lambda x: x.replace("'Semi-Furnished'",""))
data['ImpInfo'] = data['ImpInfo'].apply(lambda x: x.replace("'Unfurnished'",""))
data['ImpInfo'] = data['ImpInfo'].apply(lambda x: x.replace("'Furnished'",""))
data['ImpInfo'] = data['ImpInfo'].apply(lambda x: x.replace("'",""))
data['ImpInfo'] = data['ImpInfo'].apply(lambda x: x.strip())

data['ImpInfo'] = data['ImpInfo'].apply(lambda x: x.replace(",",""))

data = data.drop(columns =['Amenities','Overviews','More Details','ImpInfo'], axis =
1)

data['Society'] = data['Society'].apply(lambda x: x.split())

data['Location'] = data['Location'].apply(lambda x: x.split())
data['Loc'] = data['Society'] + data['Location']
data['Loc'] = data['Loc'].apply(lambda x: " ".join(x))

data['Loc'].value_counts()

loc_count = data['Loc'].value_counts()

loc_count

loc_count = loc_count[loc_count < 5]
loc_count

data['Society'] = data['Society'].apply(lambda x: " ".join(x))
data['Location'] = data['Location'].apply(lambda x: " ".join(x))

data

data.isnull().sum()

data = data.dropna()
```

```
data.isnull().sum()

data.to_csv('Cleaned_goregaon.csv')

data = data.drop(columns = ['Society','Rate','Location'])

data

x = data.drop(columns =['Rent'])
y = data['Rent']

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,handle_unknown='ignore'), ['Loc']), 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)

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)

ridge = Ridge()
pipe = make_pipeline(column_trans, scaler,ridge)
pipe.fit(x_train,y_train)
y_pred_ridge = pipe.predict(x_test)

lasso = Lasso()
pipe = make_pipeline(column_trans, scaler, lasso)
pipe.fit(x_train,y_train)
y_pred_ridge = pipe.predict(x_test)

r2_score(y_test, y_pred_lasso)
```

```
import pickle

pickle.dump(pipe, open('LassoModel.pkl', 'wb'))

location = "OTHER GOREGAON EAST"
sqft = 1320
bhk = 2
bath = 2
furni = 1
powe = 1
lift = 1
swim = 1
gym = 0
da = pd.DataFrame([[location, sqft, bhk,bath,furni,powe,lift,swim,gym]],columns =
['Loc','Carpet Area','BHK','Bath','Furnished','Power Back Up','Lift','Swimming
Pool','Gym'])

da

prediction = pipe.predict(da)[0]

prediction
```