RAJALAKSHMI ENGINEERING COLLEGE

An AUTONOMOUS institution

Affiliated to ANNA UNIVERSITY, Chennai

HOUSE PRICE PREDICTION

LAB RECORD

CS19442 – SOFTWARE ENGINEERING CONCEPTS

Submitted by

Kiruthik Rokan B	220701133
Kishore K	220701134
Kishore Kaarthik S	220701135
Krishnavarthini K H	220701136
Krithika B	220701137
Krithika R	220701138

TABLE OF CONTENT

CHAPTER 1

OVERVIEW OF THE PROJECT

The “House Price Prediction” project is a machine learning-based project that aims to predict house prices based on various influencing factors. The project involves the use of regression models to predict the price of a house given a set of features.

Problem Statement:

The main challenge that the “House Price Prediction” project addresses is the accurate prediction of house prices. House prices can fluctuate significantly based on a variety of factors. It can be difficult for individuals, real estate agents, and investors to accurately determine the fair market value of a property. This can lead to overpricing, where a house is listed at a price higher than its actual worth, or under-pricing, where a house is listed at a price lower than its actual worth. Both scenarios can have negative consequences. Overpricing can deter potential buyers, leading to the property staying on the market for a long time. Under-pricing, on the other hand, can result in a financial loss for the seller.

Data:

The project heavily relies on data related to houses and their prices. This data can come from various sources such as public records, real estate websites, or proprietary databases. The data includes features that could potentially influence the price of a house. These features include the number of rooms (which can indicate the size and capacity of the house), location (certain locations may be more desirable due to factors like safety, proximity to city centres, scenic views, etc.), size of the house (larger houses are generally more expensive), proximity to schools (houses near good schools are often more expensive), hospitals (access to healthcare facilities can increase a property’s value), etc. The quality and quantity of this data are crucial for the success of the project. A diverse and comprehensive dataset can lead to a more accurate and robust prediction model.

Benefits to Users:

Implementing the “House Price Prediction” system can provide several benefits to users:

1. **Fair Pricing:** By accurately predicting the price of a house, users can ensure they are neither overpaying nor under-pricing a property. This can be particularly beneficial for buyers, who want to ensure they are getting a good deal, and sellers, who want to make sure they are getting a fair price for their property. Real estate agents can also benefit as it can help them price their listings more accurately.
2. **Efficient Decision Making:** With a reliable prediction model, users can make quicker and more informed decisions about buying or selling a property. Instead of relying on guesswork or potentially biased human appraisals, users can use the prediction model to get an objective estimate of a property’s value.
3. **Market Trends:** Over time, the prediction model can help users understand market trends and the factors that most significantly impact house prices. This can be useful for investors looking to invest in real estate, as it can help them identify potentially undervalued properties or areas where property values are likely to increase.
4. **Personalized Recommendations:** If integrated with a user’s preferences, the system can also provide personalized recommendations. For example, if a user prefers houses with a large garden, the system could prioritize such properties in its recommendations. This can further enhance the user experience by making the house hunting process more efficient and tailored to the user’s preferences.

Key Components:

1. **Data Collection:** The first step in this project is to gather data. This could be from a public dataset like the Boston Housing dataset or a private dataset. The dataset should include features that could potentially influence the price of a house, such as the number of rooms, location, size of the house, proximity to schools, hospitals, etc.
2. **Data Preprocessing:** This step involves cleaning the data, handling missing values, outliers, and converting categorical data into numerical data. Feature scaling might also be necessary to normalize the range of input features.
3. **Exploratory Data Analysis (EDA):** This involves understanding the dataset, finding patterns and relationships between different features, and visualizing the data to gain insights.
4. **Model Building:** This is where we train our machine learning model. We could use various regression models like Linear Regression, Decision Tree Regression, Random Forest Regression, etc. The choice of model would depend on the nature of our dataset.
5. **Model Evaluation**: After training the model, we need to evaluate its performance. We could use metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), or Root Mean Squared Error (RMSE) for evaluation.
6. **Model Optimization**: Based on the evaluation, we might need to optimize our model by tuning hyperparameters or using different regression models.
7. **Deployment:** Once we are satisfied with our model's performance, we can deploy it to a web server or create a user-friendly interface where users can input the features of a house and get a predicted price.

IMPORTANT FEATURES

1. **Feature Engineering:** Feature engineering is a crucial aspect of this project. It involves creating new features from existing ones to improve the model's predictive power. For example, creating a feature that represents the age of a house from the year it was built, or creating a feature that represents the proximity to amenities from the location data.
2. **Model Selection:** The project involves testing various machine learning models to find the one that provides the most accurate predictions. This could include simple linear regression models, decision tree models, or more complex models like random forest or gradient boosting.
3. **Cross-Validation:** To ensure the model's robustness and avoid overfitting, the project uses cross-validation techniques. This involves splitting the dataset into a training set and a validation set. The model is trained on the training set and then tested on the validation set to evaluate its performance.
4. **User Interface:** The project could also include the development of a user-friendly interface where users can input the features of a house and get a predicted price. This could be a web application or a mobile app, depending on the target users.
5. **Continuous Learning:** As new data becomes available, the model should be updated to reflect the latest trends in the housing market. This could be achieved through continuous learning techniques, where the model is periodically retrained with the latest data.
6. **Privacy and Security:** Given that the project deals with potentially sensitive data, it's important to ensure that all data is handled securely and that user privacy is respected. This could involve anonymizing data, using secure data storage solutions, and ensuring that the application is secure from potential cyber threats.

CHAPTER 2

BUSINESS ARCHITECTURE

CURRENT PROCESS

The current process of house price estimation is often manual and relies heavily on real estate agents or appraisers. They use their experience and knowledge of the local market to estimate a house's price. They consider various factors such as location, size, age, condition of the house, proximity to amenities, and compare it with similar properties that have recently sold in the same area. However, this process can be subjective and may lead to inconsistencies in pricing.

The "House Price Prediction" project aims to improve the current process by using machine learning techniques to predict house prices. The process involves collecting data related to houses and their prices, preprocessing the data, training a machine learning model on this data, and then using the model to predict the prices of houses. This process can be more accurate and efficient than traditional methods, as it can analyse a large amount of data and identify complex patterns that humans might miss.

PERSONAS

1. **Home Buyers:** Home buyers are individuals or families looking to purchase a house. They can use the system to get an estimate of a house's price before making an offer. This ensures they don't overpay for the property. For example, if a buyer is interested in a house listed at $500,000, but the system predicts the house's value at $450,000, the buyer can negotiate the price down or look for other options, saving them money.
2. **Home Sellers:** Home sellers are individuals or families looking to sell their house. They can use the system to price their property appropriately, ensuring they don't underprice and lose potential profit. For instance, if a seller is unsure about how to price their property, they can input the details of their house into the system and get an estimated price. This helps them to set a competitive price that reflects the true value of their property.
3. **Real Estate Agents:** Real estate agents are professionals who help individuals buy, sell, or rent properties. They can use the system to price their listings more accurately and provide better service to their clients. For example, if an agent is representing a seller, they can use the system to set a competitive listing price. If they're representing a buyer, they can use it to ensure their client is getting a fair deal.
4. **Investors:** Investors are individuals or entities that purchase properties as an investment, with the goal of earning a return through rent, resale, or both. They can use the system to identify potentially undervalued properties for investment. For example, if the system predicts a higher price for a property than its listing price, it could indicate that the property is undervalued and could be a good investment opportunity.
5. **Appraisers:** Appraisers are professionals who estimate the value of real property. They can use the system as a reference when appraising properties. For instance, an appraiser could use the system's prediction as a starting point and then adjust the value based on their professional judgment and factors the system might not have considered.

BUSINESS PROBLEM

1. **Inaccurate Pricing:** One of the most significant issues in the real estate industry is the difficulty of accurately pricing properties. Incorrect pricing can lead to properties being overpriced, which can deter potential buyers and cause the property to remain on the market for a long time. On the other hand, underpricing can result in a financial loss for the seller. This project aims to provide a more accurate method of predicting house prices, reducing the risk of overpricing or underpricing.
2. **Time-Consuming Appraisals:** Traditional methods of property valuation, such as manual appraisals, are time-consuming and often require the expertise of a professional appraiser. This project aims to automate the process, making it quicker and more efficient.
3. **Subjectivity in Pricing:** Traditional methods of pricing properties can be subjective, with different real estate agents or appraisers potentially arriving at different valuations for the same property. The “House Price Prediction” project aims to provide a more objective method of pricing properties.
4. **Lack of Transparency:** Buyers and sellers often lack access to the information used to price a property, leading to a lack of transparency in the process. This project aims to make the factors influencing a property’s price more transparent, allowing buyers and sellers to make more informed decisions.
5. **Inefficiencies in the Market:** Inaccurate pricing can lead to inefficiencies in the real estate market, with properties either being undervalued or overvalued. By providing more accurate price predictions, this project can help to reduce these inefficiencies, leading to a more efficient market.

CHAPTER 3

REQUIREMENTS AS USER STORY

FUNCTIONAL REQUIREMENTS

Functional requirements define the basic system behavior. In other words, they specify what the system should do. For the “House Price Prediction” project, the functional requirements could be:

1. **Data Collection:** The system should be able to collect and store data related to houses and their prices.
2. **Data Preprocessing:** The system should be able to preprocess the data, including handling missing values, outliers, and categorical data.
3. **Price Prediction:** The system should be able to predict the price of a house based on its features.
4. **User Interface:** The system should provide a user-friendly interface where users can input the features of a house and get a predicted price.
5. **Continuous Learning:** The system should be able to update the prediction model as new data becomes available.

NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements define the system’s properties or qualities, such as performance, security, usability, etc. For the “House Price Prediction” project, the non-functional requirements could be:

1. **Performance:** The system should return the predicted price within a few seconds of the user inputting the details of a house.
2. **Usability:** The system should be user-friendly, with a simple and intuitive interface that users can navigate easily.
3. **Security:** The system should ensure that all user data is stored securely and that user privacy is respected.
4. **Scalability:** The system should be able to handle an increasing amount of work by adding resources.
5. **Reliability:** The system should function without failure under stated conditions for a specified period of time.

USER STORIES

1. As a home buyer, I want to input the features of a house and get an estimated price so that I can make an informed offer.
2. As a home seller, I want to input the details of my house and get an estimated price so that I can list it at a competitive price.
3. As a real estate agent, I want to input the details of a house and get an estimated price so that I can accurately price my listings.
4. As an investor, I want to input the details of a house and get an estimated price so that I can identify potentially undervalued properties.
5. As a user, I want to see the factors that the model considers when predicting prices so that I can understand how the prediction is made.
6. As a user, I want to see how accurate the model's predictions are so that I can trust its estimates.
7. As a user, I want the system to remember my previous searches so that I can easily refer back to them.
8. As a user, I want to compare the predicted prices of multiple houses so that I can make a comparative analysis.
9. As a user, I want to see the prediction in my local currency so that I can easily understand the price.
10. As a user, I want to receive updates when there are significant changes in the predicted price of a saved search so that I can stay informed.

POKER PLANNING ESTIMATIONS

1. Story 1: Medium.

   This involves creating a user interface for input and displaying the estimated price. It's a medium complexity task.

2. Story 2: Medium

   This is similar to the first user story and also involves creating a user interface for input and displaying the estimated price.

3. Story 3: Medium

   This is similar to the first two user stories.

4. Story 4: Medium

   This is similar to the first three user stories.

5. Story 5: Hard

   This involves modifying the user interface to display the factors considered in the prediction. It's a medium complexity task.

6. Story 6: Hard

   This involves implementing a method to calculate and display the accuracy of the model. It's a medium complexity task.

7. Story 7: Hard

   This involves implementing a feature to save and retrieve previous searches. It's a high complexity task due to the need for data persistence.

8. Story 8: Hard

   This involves modifying the user interface to allow for comparison of multiple houses. It's a high complexity task.

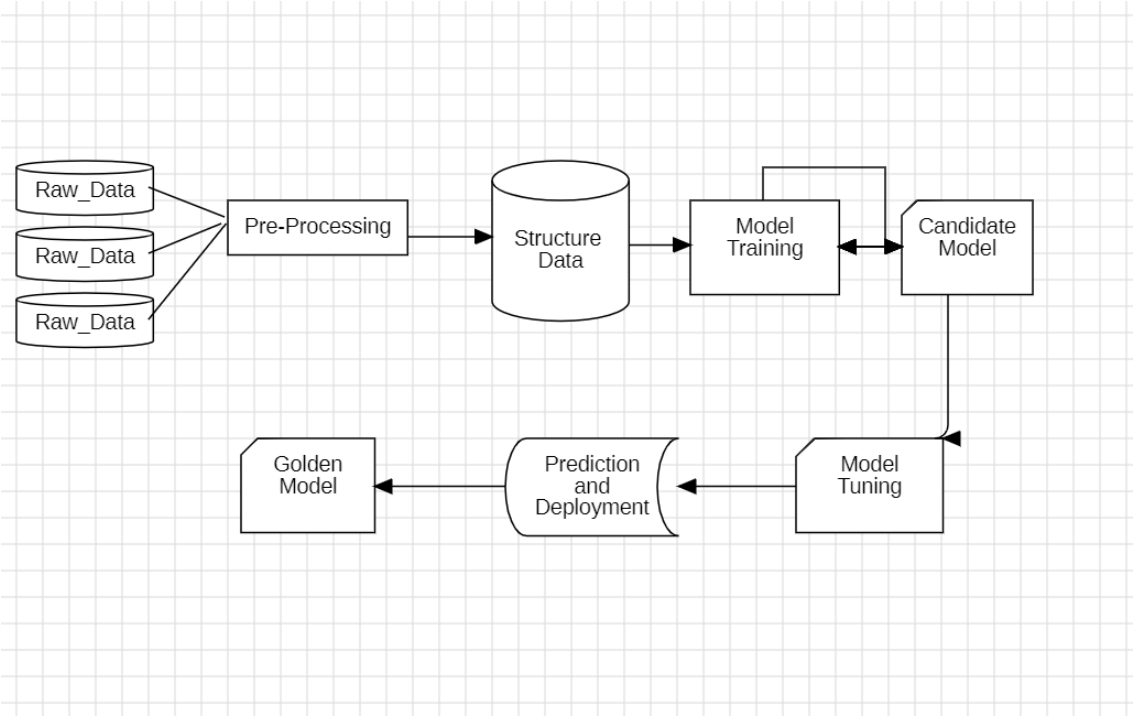
9. Story 9: Medium

   This involves implementing a feature to convert the predicted price into different currencies. It's a high complexity task due to the need for accurate and up-to-date exchange rates

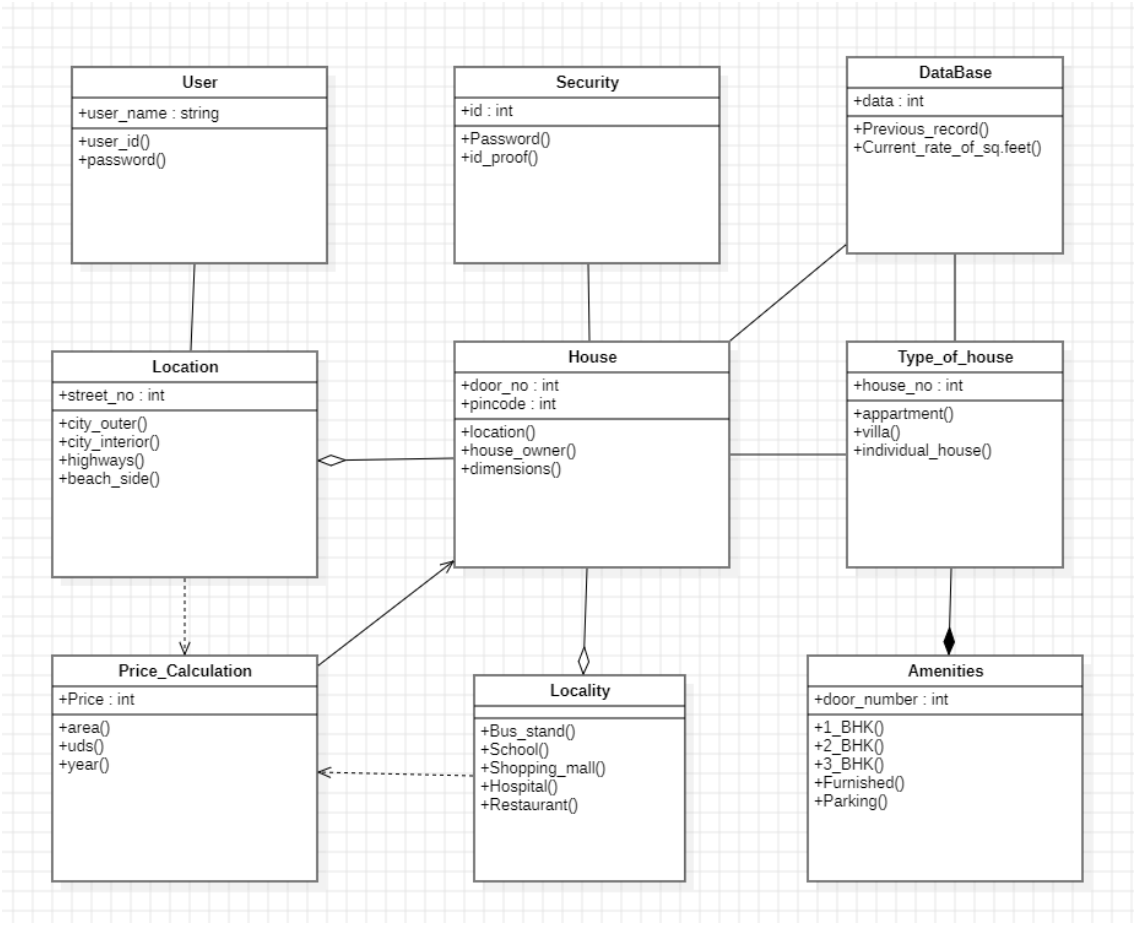
CHAPTER 4

UML DIAGRAM

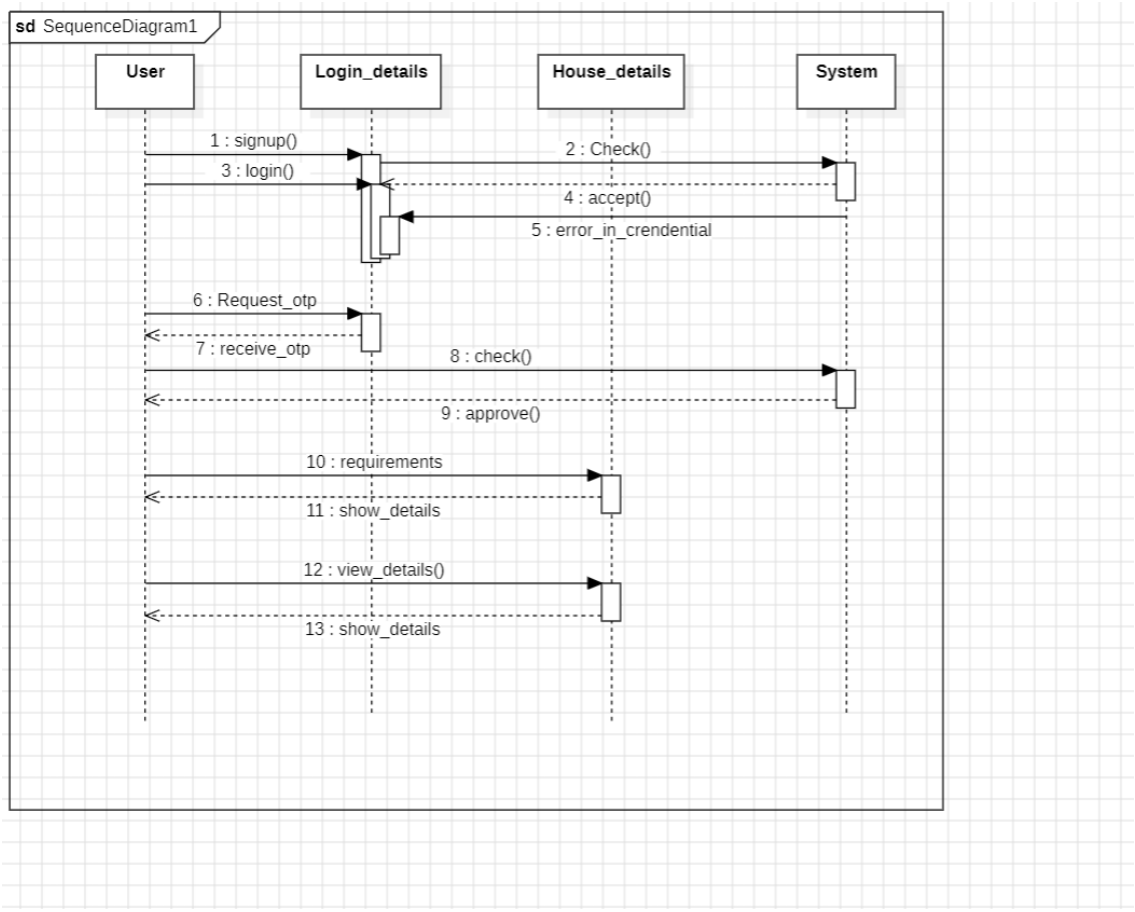
ARCHITECTURE DIAGRAM



CLASS DIAGRAM



SEQUENCE DIAGRAM



CHAPTER 5

TEST STRATEGY

Test Plans

A test plan for the “House Price Prediction” project could include the following elements:

1. **Introduction:** A brief overview of the system and the purpose of the test.
2. **Test Items:** A list of features to be tested.
3. **Features to be Tested:** Detailed descriptions of the system features to be tested.
4. **Features not to be Tested:** Detailed descriptions of the system features not to be tested.
5. **Approach:** The overall strategy and methodology that will be used to conduct the tests.
6. **Item Pass/Fail Criteria:** The criteria that will be used to determine whether a test item has passed or failed.
7. **Test Deliverables:** A list of all the documents, tools, and other components to be developed and maintained in support of the testing effort.
8. **Testing Tasks:** A list of tasks to be performed during testing.
9. **Environmental Needs:** Any necessary infrastructure, tools, or equipment.
10. **Responsibilities:** The roles and responsibilities of each member of the testing team.
11. **Schedule:** The timeline for testing activities.
12. **Risks and Contingencies:** Potential risks and contingency plans.

Test Cases

Here are some example test cases for the user stories mentioned earlier:

1. **User Story 1 – Home Buyers:**
    - Happy Path: Input valid house features and verify that the system returns a reasonable price estimate.
    - Error Scenario: Input invalid house features (e.g., negative number of rooms) and verify that the system returns an error message.
2. **User Story 2 – Home Sellers:**
    - Happy Path: Input valid house features and verify that the system returns a reasonable price estimate.
    - Error Scenario: Input invalid house features (e.g., extremely large house size) and verify that the system returns an error message.
3. **User Story 3 – Real Estate Agents:**
    - Happy Path: Input valid house features and verify that the system returns a reasonable price estimate.
    - Error Scenario: Input invalid house features (e.g., non-existent location) and verify that the system returns an error message.
4. **User Story 4 – Investors:**
    - Happy Path: Input valid house features and verify that the system returns a reasonable price estimate.
    - Error Scenario: Input invalid house features (e.g., negative house size) and verify that the system returns an error message.
5. **User Story 5 – Appraisers:**
    - Happy Path: Input valid house features and verify that the system returns a reasonable price estimate.
    - Error Scenario: Input invalid house features (e.g., non-numeric input for numeric fields) and verify that the system returns an error message.

CHAPTER 6

DEPLOYMENT ARCHITECTURE

The architecture for a "House Price Prediction" project could be designed using a Machine Learning (ML) pipeline, which typically includes the following components:

1. **Data Collection:** This is the first step in the pipeline where you gather data related to houses and their prices. The data could come from various sources such as public records, real estate websites, or proprietary databases.
2. **Data Preprocessing:** In this step, the raw data is cleaned and transformed into a format that can be used by the machine learning model. This could involve handling missing values, outliers, and converting categorical data into numerical data.
3. **Feature Engineering:** This involves creating new features from existing ones to improve the model's predictive power. For example, creating a feature that represents the age of a house from the year it was built, or creating a feature that represents the proximity to amenities from the location data.
4. **Model Training:** This is where you select a machine learning algorithm and train it on your pre-processed data. You could use various regression models like Linear Regression, Decision Tree Regression, Random Forest Regression, etc.
5. **Model Evaluation:** After training the model, you need to evaluate its performance. You could use metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), or Root Mean Squared Error (RMSE) for evaluation.
6. **Model Optimization:** Based on the evaluation, you might need to optimize your model by tuning hyperparameters or using different regression models.
7. **Model Deployment:** Once you are satisfied with your model's performance, you can deploy it to a web server or create a user-friendly interface where users can input the features of a house and get a predicted price.
8. **Monitoring and Updating:** After deployment, it's important to monitor the model's performance and update it as new data becomes available.