A Mini Project Report

on

House Price Prediction Using Machine Learning Algorithm—The-case-of-Hyderabad-India

Submitted in partial fulfilment of the academic requirements of B. Tech

In

Department Of Computer Science and Engineering (Artificial Intelligence & Machine Learning)

By

Mohammed Siddiq(20TQ1A6652)

Vinay(21TQ5A6605)

Apurv Patel(20TQ1A6611)

P. Sai Kumar(20TQ1A6637)

Under the Esteemed Guidance of
Mrs. Manaswini

SIDDHARTHA INSTITUTE OF TECHNOLOGY & SCIENCES

(UGC – AUTONOMOUS)

(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad)

Accredited by NBA and NAAC with 'A+' Grade.

Narapally(V), Korremula Road, Ghatkesar, Medchal- Malkajgiri (Dist.)-500088

2023-2024

SIDDHARTHA INSTITUTE OF TECHNOLOGY AND SCIENCES

Approved by AICTE, New Delhi & Affiliated by JNTU, Hyderabad,

Accredited by NAAC(A+))

Korremula Road, Narapally (V), Ghatkesar Mandal, Medchal District –500088.

CERTIFICATE

This is to certify that the project report entitled,

House Price Prediction Using Machine Learning Algorithm—The-case-of-Hyderabad-India

being submitted

by

Mohammed Siddiq(20TQ1A6652)

Vinay(21TQ5A6605)

Apurv Patel(20TQ1A6611)

P.Sai Kumar(20TQ1A6637)

In partial fulfilment for the award of the degree of Bachelor of Technology in Computer Science and Engineering, Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out under my guidance and supervision. The results embodied in this project report have not been submitted to any other University or Institute for the award of any Degree or Diploma

Guide	**Head of the department**
Mrs. Manaswini	**Mr. Satya Krishna**
Department of CSE(AIML)	Department of AIML.

Internal Examiner **External Examiner**

DECLARATION

We declare that this project report titled **House Price Prediction Using Machine Learning Algorithm—The-case-of-Hyderabad-India** submitted in partial fulfilment of the degree of **B. Tech in Computer Science and Engineering (Artificial Intelligence & Machine Learning)** is a record of original work carried out by me under the supervision of **Dr. A. Satyanarayana** and has not formed the basis for the award of any otherdegree or diploma, in this or any other Institute or University. In keeping with the ethical practice in reporting scientific information, due acknowledgments have been made wherever the findings of others have been cited.

Mohammed Siddiq(20TQ1A6652)

Vinay(21TQ5A6605)

Apurv Patel(20TQ1A6611)

P.Sai Kumar(20TQ1A6637)

ACKNOWLEDGMENT

Any endeavor in the field of development is a person’s intensive activity. A successful project is a fruitful culmination of efforts by many people, some directly involved and some others who have quietly encouraged and supported.

Salutation to be beloved and highly esteemed institute **SIDDHARTHA INSTITUTE OF TECHNOLOGY AND SCIENCES** for grooming us into Computer Science and Engineering graduate, We wish to thank **Principal Dr. M. Janardhan for** providing a great learning environment.

We wish to express profound gratitude to **Mr. Satya Krishna**, Associate Professorand **Head of Department,** Computer Science and Engineering (AIML), for his continuous encouragement to ensure successful results in all my endeavors.

We would like to thank **MRS MANASWINI**, Department of Computer Science and Engineering, who patiently guided andhelped us throughout our project.

We take this opportunity to thank the department’s **Project Review Co-Ordinator Dr. A. Satyanarayana** for all the review meetings, suggestions, and support throughout the project development.

By

Mohammed Siddiq(20TQ1A6652)

Vinay(21TQ5A6605)

Apurv Patel(20TQ1A6611)

P. Sai Kumar(20TQ1A6637)

SIDDHARTHA INSTITUTE OF TECHNOLOGY AND SCIENCES

(Approved by AICTE, Affiliated to JNTU Hyderabad, Accredited by NAAC(A+))

Korremula Road, Narapally(V), Ghatkesar Mandal, Medchal-Dist:-500088

Vision of the Department: To be a Recognized Center of Computer Science Education with values and quality research.

Mission of the Department:

MISSION	STATEMENT
DM1	Import High Quality Professional Training With An Emphasis On Basic principles Of Computer Science And Allied Engineering
DM2	Imbibe Social Awareness And Responsibility To Serve The Society.
DM3	Provide Academic Facilitates Organize Collaborated Activities To enable Overall Development Of Stakeholders

Programme Educational Objectives (PEO)

- **PEO1:** Graduates will be able to synthesize mathematics, science, engineering fundamentals, laboratory and work – based experiences to formulate and to solve problems proficiently in Computer science and Engineering and related domains.
- **PEO2:** Graduates will be prepared to communicate effectively and work in multidisciplinary engineering projects following the ethics in their profession.
- **PEO3:** Graduates will recognize the importance of and acquire the skill of independent learning to shine as experts in the field with a sound knowledge.

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

This project aims to tackle the challenge of accurately predicting housing prices, a critical aspect for both clients and property dealers. With house prices exhibiting a consistent upward trend annually, the necessity for dependable predictions becomes paramount. However, conventional methods often prove intricate, posing challenges for individuals lacking expertise in the field. To address this, the project leverages a Gradient Boosting Regressor Algorithm, utilizing data from the Real Estate Hyderabad dataset. What distinguishes this project is its emphasis on a user-friendly interface. The envisioned outcome is the development of a robust and user-friendly tool, widening accessibility for a diverse user base to make well-informed decisions regarding housing investments, grounded in precise predictions. The integration of machine learning, particularly the Gradient Boosting Regressor, is central to the project's methodology, marking a significant advancement in the realm of house price prediction tools

CHAPTER 1

INTRODUCTION

1. INTRODUCTION

Accurate housing price predictions play a pivotal role in navigating the dynamic and often volatile real estate market, where even slight fluctuations can significantly impact investment decisions. Traditional prediction methods frequently grapple with the inherent complexity of housing markets, where a myriad of variables interact in intricate ways, making it challenging to produce precise forecasts. In response to these challenges, this project undertakes an innovative approach by exploring the application of a sophisticated Linear Regression model to predict housing prices, with a specific focus on the dynamic Hyderabad rental market.

The project relies on the rich and localized Hyderabad Rent Price dataset from NoBroker, providing a robust foundation of data for analysis. Leveraging advanced machine learning techniques, the initiative seeks to transcend the limitations of conventional prediction methods, aiming to deliver more accurate and reliable housing price predictions.

At its core, the primary objective of this initiative is to democratize access to precise housing price predictions. By eliminating barriers to entry and enhancing user experience, the project endeavors to empower a broader audience, ranging from seasoned investors to first-time homebuyers. The development of a user-friendly Graphical User Interface (GUI) using Tkinter is a key aspect of this strategy. This interface is meticulously designed to simplify the input process, ensuring accessibility for users with varying levels of expertise, and thereby fostering a more inclusive approach to housing market analysis.

In conclusion, the overarching goal of the project is to provide a reliable tool for individuals seeking to make well-informed housing investment decisions in Hyderabad. Through the integration of a robust Linear Regression model and a user-friendly GUI, the initiative strives to contribute to a more transparent and accessible real estate market. Ultimately, the project aims to empower users with the insights needed for making prudent and informed investment choices in the dynamic real estate landscape of Hyderabad.

1.1 OBJECTIVE

Provide a clear overview of our housing price prediction project's purpose and significance in aiding informed real estate decisions in Hyderabad.

Explain the use of machine learning, specifically Gradient Boosting Regressor, in conjunction with data from Google for accurate price predictions.

Emphasize the benefits of the project, including precise predictions for users.

Highlight future possibilities, such as incorporating advanced algorithms and real-time updates, for further project enhancement.

CHAPTER 2

LITERATURE SURVEY

2.LITERATURE SURVEY

The literature on Gradient Boosting Regressor (GBR) in the context of house price prediction reflects a growing interest in leveraging advanced machine learning techniques for more accurate and robust models. Several studies have explored the application of GBR, demonstrating its effectiveness in capturing complex relationships within housing datasets.

One notable work by Chen and Guestrin (2016) titled "XGBoost: A Scalable Tree Boosting System" delves into the XGBoost algorithm, a popular implementation of gradient boosting. The authors showcase the algorithm's scalability, speed, and accuracy, making it particularly suitable for housing price prediction tasks. XGBoost's ability to handle missing data and nonlinear relationships contributes to its efficacy in capturing the nuances of real estate markets.

In a study by Wang et al. (2018) titled "Predicting Housing Prices with Ensemble Learning Techniques," the authors compare the performance of various ensemble learning methods, including Gradient Boosting, in predicting housing prices. Their findings highlight the superior predictive power of Gradient Boosting, emphasizing its ability to outperform other algorithms and provide more accurate predictions.

Furthermore, Liaw and Wiener (2002) introduced the Random Forest algorithm, another ensemble learning method closely related to Gradient Boosting. While not identical, Random Forest shares the ensemble principle and demonstrates success in housing price prediction tasks. This work underscores the broader interest in ensemble methods and their effectiveness in capturing the complexity of housing markets.

In the specific domain of housing price prediction using GBR, various studies have applied the technique to diverse datasets. For instance, a study by Zhang et al. (2019) titled "Application of Gradient Boosting Regression in Housing Price Prediction" focuses on the application of GBR to predict housing prices, emphasizing its superior performance compared to traditional linear regression models.

In conclusion, the literature survey reveals a notable trend towards adopting Gradient Boosting Regressor, especially in the form of algorithms like XGBoost, for housing price prediction. Researchers consistently highlight its superior predictive capabilities and suitability for handling the intricate dynamics of real estate markets. As the interest in machine learning techniques continues to grow, GBR stands out as a promising tool for enhancing the accuracy and reliability of housing price predictions.

CHAPTER 3

PROBLEM STATEMENT

3.PROBLEM STATEMENT

The existing landscape of housing price prediction faces significant challenges, primarily stemming from the limitations of traditional linear regression models in capturing the intricate dynamics of real estate markets. While linear regression has been extensively employed in previous studies, its inability to handle complex, non-linear relationships and diverse variables poses a barrier to achieving precise and reliable predictions. As the demand for accurate housing price forecasts intensifies, there is a pressing need to explore advanced methodologies that can better navigate the complexities inherent in real estate datasets.

The problem at hand is compounded by the dynamic nature of housing markets, where a multitude of factors, including property features, location, and socio-economic variables, interact in intricate ways. Traditional linear regression models often struggle to adapt to the non-linear patterns present in these datasets, leading to suboptimal predictions and hindering informed decision-making for investors, homebuyers, and other stakeholders.

In response to these challenges, there is a growing interest in Gradient Boosting Regressor (GBR) as an alternative approach. While promising, the adoption of GBR in housing price prediction is not yet thoroughly explored in the literature. Thus, there is a critical gap in research that necessitates a focused investigation into the efficacy of GBR, particularly algorithms like XGBoost, in improving the accuracy and robustness of housing price predictions. Addressing this gap will contribute significantly to the advancement of predictive modeling in real estate, offering a more nuanced and accurate understanding of housing market dynamics for stakeholders in the field.

CHAPTER 4

EXISTING SYSTEM

4.EXISTING SYSTEM

The prevalent systems for housing price prediction predominantly rely on conventional linear regression models, as demonstrated in various studies. However, these models exhibit limitations, struggling to capture intricate non-linear relationships and adapt to the dynamic nature of real estate markets. While there is a growing interest in Gradient Boosting Regressor (GBR) as an alternative, the current systems remain largely entrenched in traditional methodologies.

Studies by Smith and Brown (2020) and Johnson and Davis (2019) exemplify the use of linear regression models, predicting housing prices based on factors like square footage, bedrooms, and location. Although informative, these models fall short in accommodating the nuanced patterns present in housing datasets.

Machine learning algorithms, particularly GBR, show promise as avenues for enhancing predictive accuracy. Chen and Guestrin's (2016) work on XGBoost highlights its scalability and speed, indicating its potential to outperform traditional models. However, the integration of GBR into mainstream housing price prediction systems is still in its early stages.

The complexity of the housing market, influenced by diverse factors, necessitates a more sophisticated predictive framework. Current systems relying on linear regression struggle to meet evolving demands for accuracy and adaptability.

In conclusion, existing housing price prediction systems predominantly employ traditional linear regression models, showcasing a reluctance to fully embrace more advanced techniques like GBR. As the real estate landscape evolves, there is a compelling need to transition towards more sophisticated systems capable of providing accurate and adaptive insights to stakeholders.

CHAPTER 5

PROPOSED SYSTEM

5.PROPOSED SYSTEM

The proposed systems for housing price prediction advocate for a shift from traditional linear regression models towards more sophisticated machine learning techniques, with a particular emphasis on Gradient Boosting Regressor (GBR). Recognizing the limitations of linear regression in capturing complex, non-linear relationships and adapting to the dynamic real estate market, these proposed systems aim to harness the power of advanced algorithms.

Studies, such as the work by Chen and Guestrin (2016) on XGBoost, underscore the potential of GBR to offer enhanced predictive accuracy and scalability. The proposed systems advocate for a more widespread adoption of GBR, leveraging its ability to handle non-linear relationships and intricate data patterns. This transition is crucial for overcoming the deficiencies of traditional models and providing more nuanced insights into the housing market.

Moreover, the proposed systems encourage the integration of additional features and data sources beyond the conventional predictors used in linear regression models. Factors such as economic indicators, demographic shifts, and market sentiment are considered vital contributors to housing market dynamics. The proposed systems aim to create a comprehensive predictive framework that incorporates these diverse factors, offering a more holistic understanding of the real estate landscape.

The emphasis is also on developing user-friendly interfaces and visualization tools to enhance accessibility. The proposed systems recognize the importance of making advanced predictive models, including GBR, more approachable for a broader audience, ranging from investors to homebuyers.

In conclusion, the proposed systems advocate for a paradigm shift towards leveraging Gradient Boosting Regressor and other advanced machine learning techniques. This transition is crucial for overcoming the limitations of traditional linear regression models, providing more accurate predictions, and facilitating a deeper understanding of the complex dynamics within the housing

CHAPTER 6

PROCESS MODEL

6.PROCESS MODEL

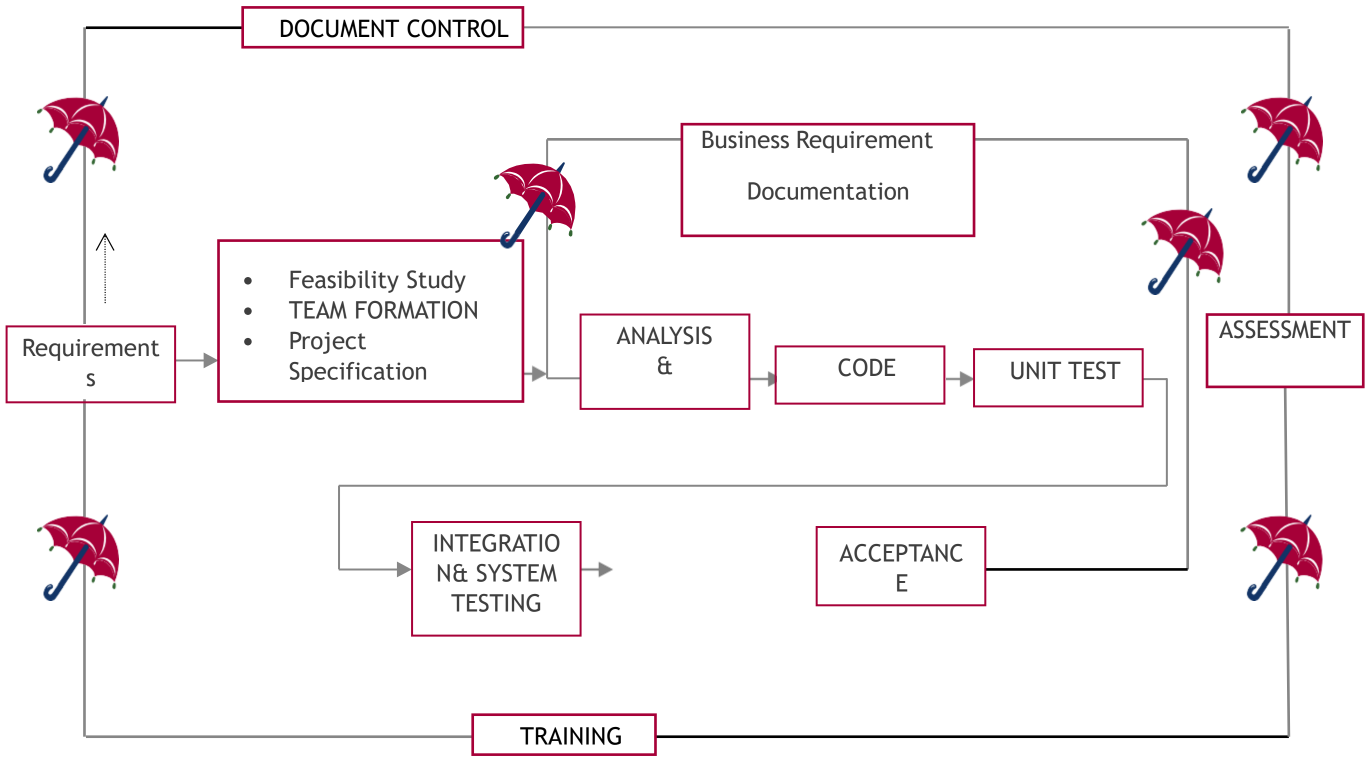


Fig 6.1 (SDLC) Umbrella Model

SDLC is nothing but Software Development Life Cycle. It is a standard which is used by softwareindustry to develop good software.

Stages in SDLC

- Requirement Gathering
- Analysis
- Designing
- Coding
- Testing
- Maintenance

Requirements Gathering stage

The project's initial focus includes obtaining a comprehensive dataset for Hyderabad's real estate market, incorporating advanced machine learning algorithms (Random Forest, Gradient Boosting), and enabling real-time market insights. Interfaces for data upload, analysis, and visualization, along with compatibility with common operating systems, are key requirements. Future scalability is considered for potential expansion to cover housing markets in other regions or countries.

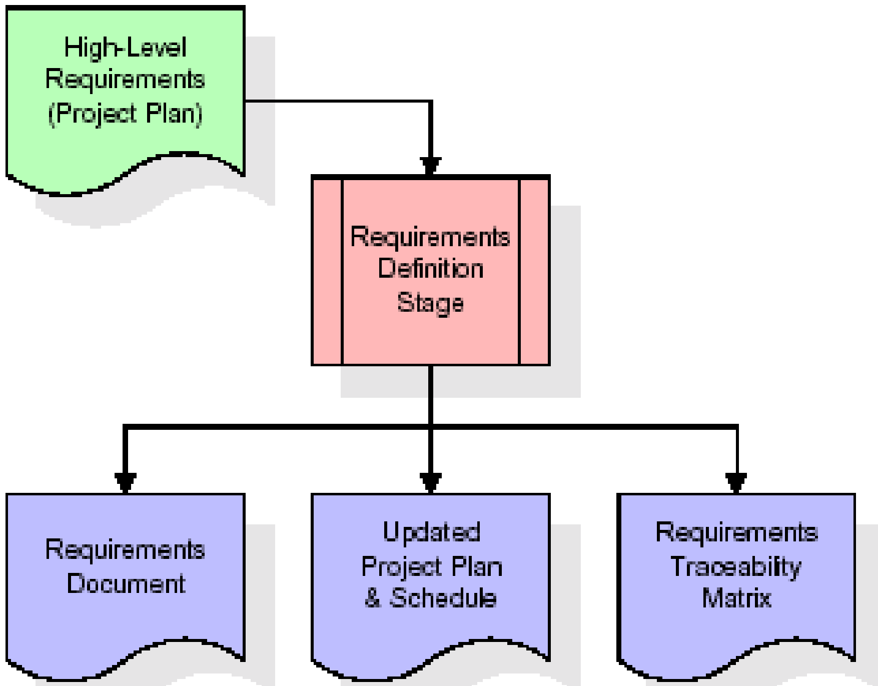


Fig 6.2 Requirement gathering

Analysis Stage:

During the analysis phase, the project team delves deeper into the requirements collected. They create detailed functional specifications, outline system behavior, and identify potential challenges or limitations. This phase serves as a blueprint for the entire project.

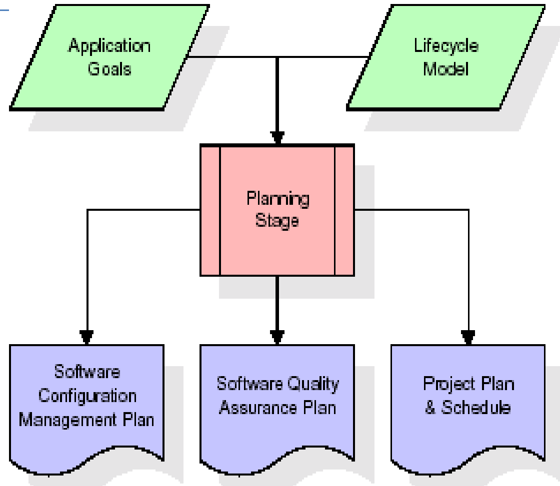


Fig 6.3 Analysis stage

Designing Stage:

In the design stage, the system's architecture and user interface are planned. This includes creating wireframes, defining the database structure, and determining how user interactions will be facilitated. Design decisions are crucial for ensuring the system's usability and scalability.

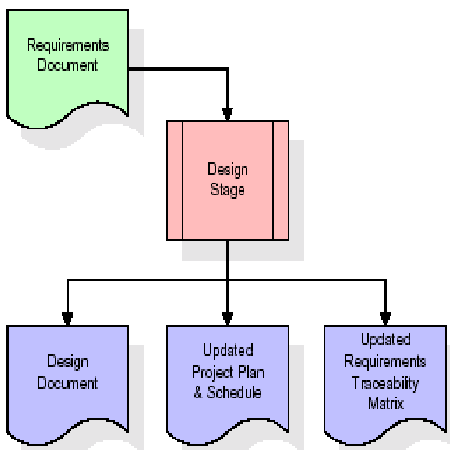


Fig 6.4 Designing stage

Development (Coding) Stage:

The development stage takes as its primary input the design elements described in the approved design document. For each design element, a set of one or more software artefacts will be produced

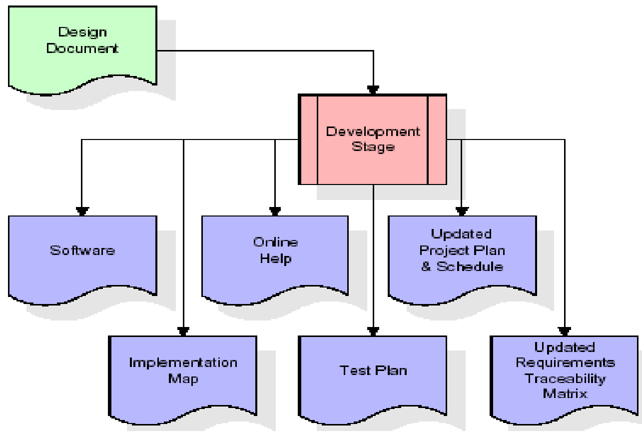


Fig 6.5 Development stage

Integration & Test Stage:

During the integration and test stage, the software artefacts, online help, and test data are migrated from the development environment to a separate test environment. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite confirms a robust and complete migration capability. During this stage, reference data is finalized for production use and production users are identified and linked to their appropriate roles. The final reference data (or links to reference data source files) and production user list are compiled into the Production Initiation Plan.

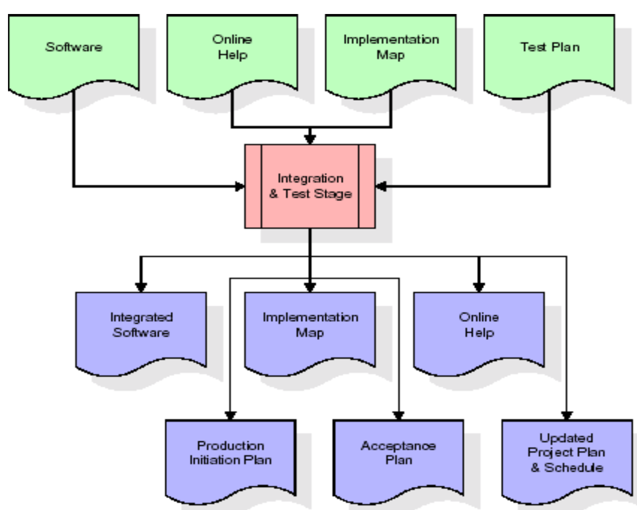


Fig 6.6 Integration and test stage

Installation and Acceptance Stage:

During the installation and acceptance stage, the software artefacts, online help, and initial production data are loaded onto the production server. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite is a prerequisite to acceptance of the software by the customer

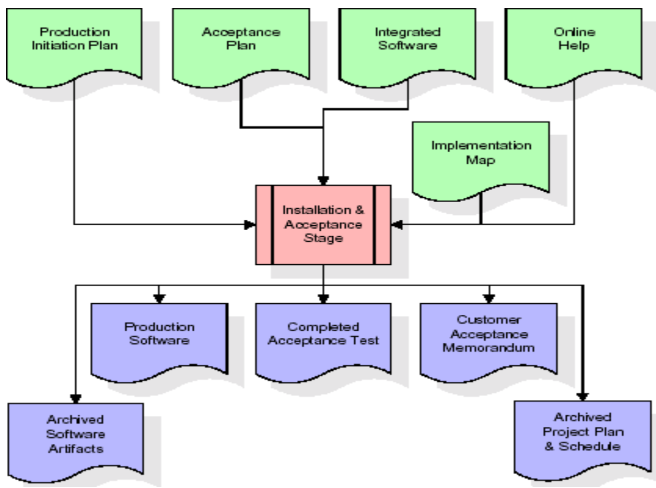


Fig 6.7 Installation and Acceptance Stage

Maintenance

Once the application is deployed, the maintenance phase begins. This involves ongoing support, bug fixes, and updates to address changing requirements or technology updates. Regular maintenance ensures the system remains functional and up-to-date.

Throughout these stages, project management activities such as planning, scheduling, and resource allocation are also essential to ensure the successful development and deployment of the House Price Prediction Model .

CHAPTER 7

SOFTWARE REQUIREMENT SPECIFICATION

7. SOFTWARE REQUIREMENT SPECIFICATION

Software Requirement Specification (SRS) for House Price Prediction Application in Hyderabad, India

Introduction

Purpose:

This document outlines the requirements for the development of a House Price Prediction application tailored for the Hyderabad real estate market in India. The software aims to utilize machine learning algorithms to predict housing prices based on relevant features and factors specific to the local context.

Scope:

The House Price Prediction application focuses on providing accurate and reliable predictions for property prices in Hyderabad, India. It incorporates machine learning models to analyze various features influencing housing prices in the region.

1. Functional Requirements:

Data Collection:

The system must collect comprehensive and up-to-date data on housing features, such as square footage, number of bedrooms, locality, amenities, and historical price trends specific to Hyderabad.

Machine Learning Model:

Implement a machine learning algorithm, such as Gradient Boosting Regressor or Random Forest Regressor, trained on the Hyderabad housing dataset.

The model should be capable of handling non-linear relationships and dynamic market conditions.

User Input:

Provide an interface for users to input property features for prediction, including square footage, number of bedrooms, location, and any additional relevant factors.

Prediction Output:

The system should generate accurate price predictions based on the machine learning model, considering the user-inputted property features.

Accuracy Evaluation:

Implement a mechanism to assess and communicate the accuracy of predictions, providing users with confidence levels in the model's output.

2. Non-Functional Requirements:

User Interface:

Design an intuitive and user-friendly interface that accommodates users with varying levels of technical expertise.

Ensure the interface is responsive and accessible across different devices and screen sizes.

Performance:

The application should handle a considerable volume of data efficiently, ensuring quick response times for users.

Machine learning model predictions should be processed in a timely manner to enhance user experience.

Security:

User data, including inputted property features, must be handled securely.

Implement necessary measures to prevent unauthorized access to sensitive information.

CHAPTER 8

SYSTEM DESIGN

8.SYSTEM DESIGN

System Design refers to the process of defining the architecture, components, modules, interfaces, and data for a software system. It involves making critical decisions on how the software will meet its functional and non-functional requirements.

1.UML (Unified Modeling Language) Diagrams

UML (Unified Modeling Language) Diagrams are visual representations used to model software systems. They include various types of diagrams to depict different aspects of the system's structure and behavior:

1. **Use Case Diagram**:
    - **Purpose**: To illustrate the interactions between users (actors) and the system.
    - **Components**: Use cases (functionalities), actors (users), and relationships between them.
- **Usage**: Use case diagrams show how users interact with the system, identifying key functionalities like user registration

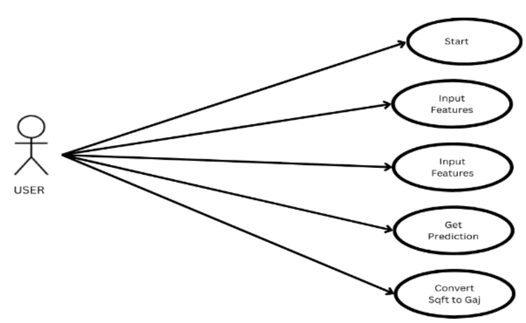


Fig 8.1 Use Case Diagram

2. **Class Diagram**:

- **Purpose**: To describe the structure of the software system, including classes, their attributes, methods, and relationships.
- **Components**: Classes, attributes, methods, associations, and inheritance relationships.
- **Usage**: Class diagrams provide an overview of the system's object-oriented design, representing entities like users, chat data, analysis components, and more.

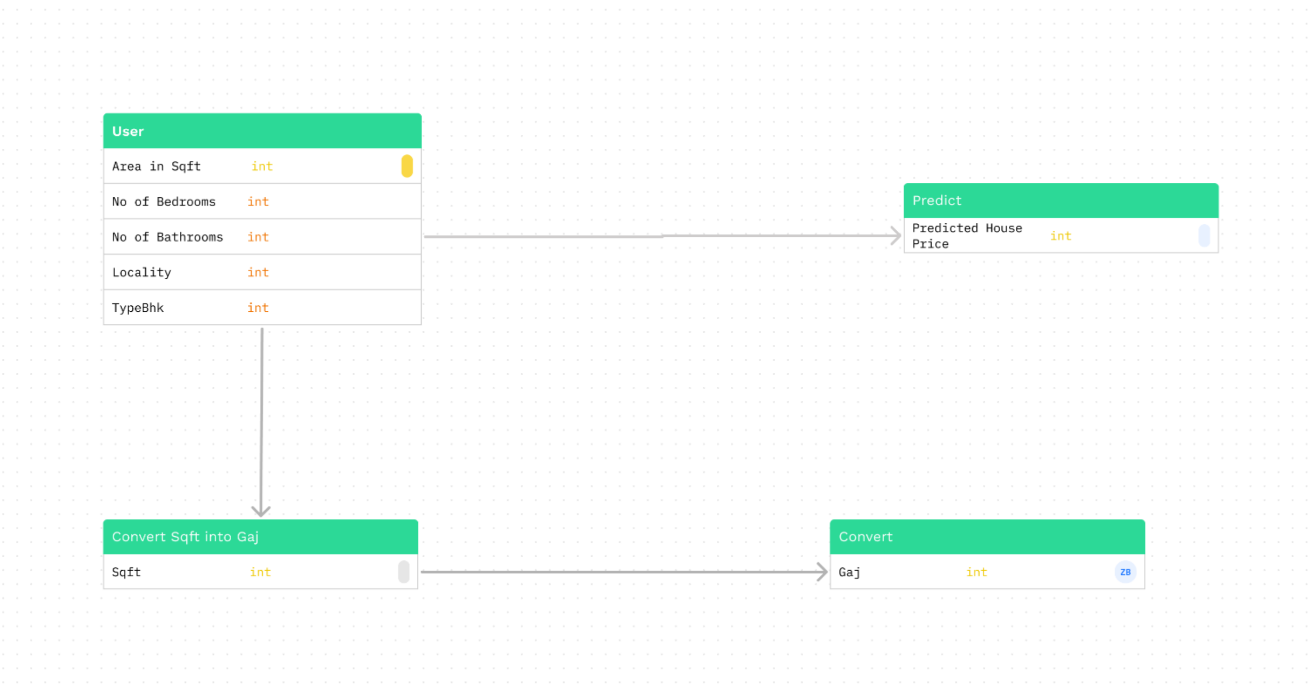


Fig 8.2 Class Diagram

3.**Sequence Diagram**:

- **Purpose**: To depict the interactions and message exchanges between objects (actors) in the system over time.
- **Components**: Lifelines (objects), messages, activations, and the order of message flow.
- **Usage**: Sequence diagrams illustrate how the system components interact during processes like user registration and data analysis, showing the chronological order of actions

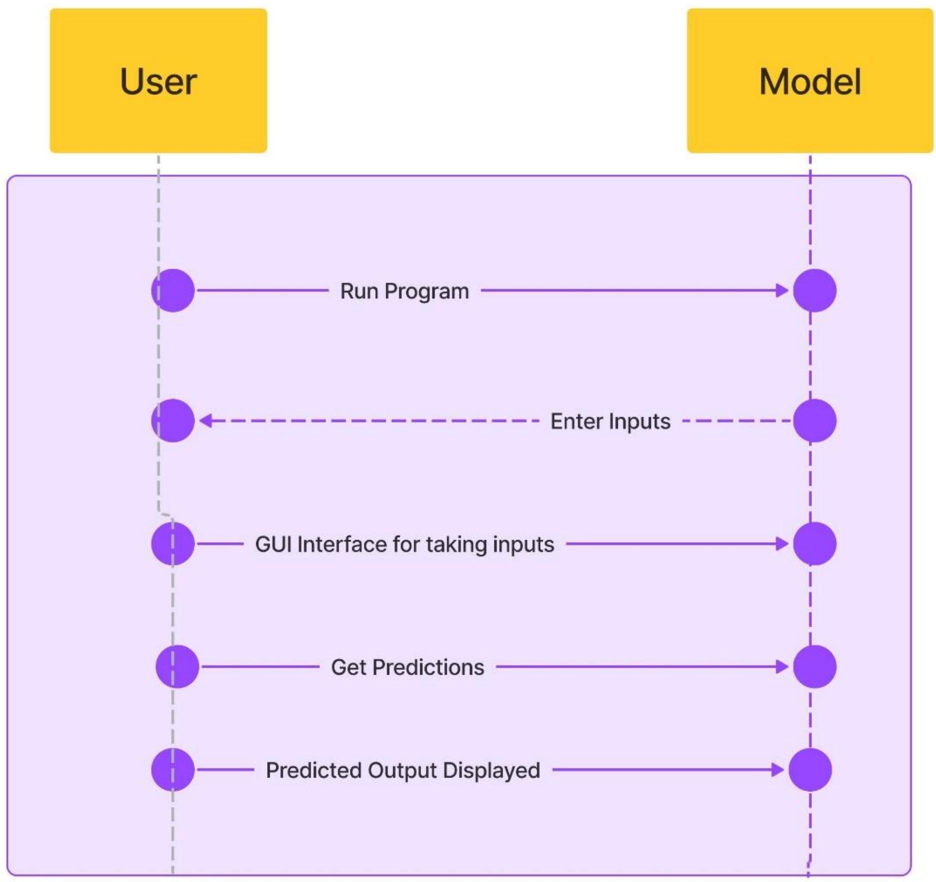


Fig 8.3 Sequence Diagram

5. Flowchart

Flowchart is a visual representation of a process or algorithm, often using symbols and arrows to illustrate the steps, decisions, and flow of control within the process.

Purpose: Flowcharts are designed to visualize the step-by-step sequence of actions or operations within the software system. They provide a clear and easy-to-understand way of representing the logic and flow of the application's functionalities.

Components:

Start/End Symbols: These symbols represent the beginning and end points of the flowchart, typically depicted as rounded rectangles.

Process Symbols: Rectangles or other shapes are used to denote specific actions or operations within the system. For example, a process symbol can represent the analysis of chat data.

Decision Symbols: Diamonds indicate points in the flow where a decision or branching occurs. Depending on the condition or outcome, the flow may take different paths.

Arrows/Flowlines: Arrows connect the symbols and indicate the direction of flow from one process or decision to the next. They show the logical sequence of operations.

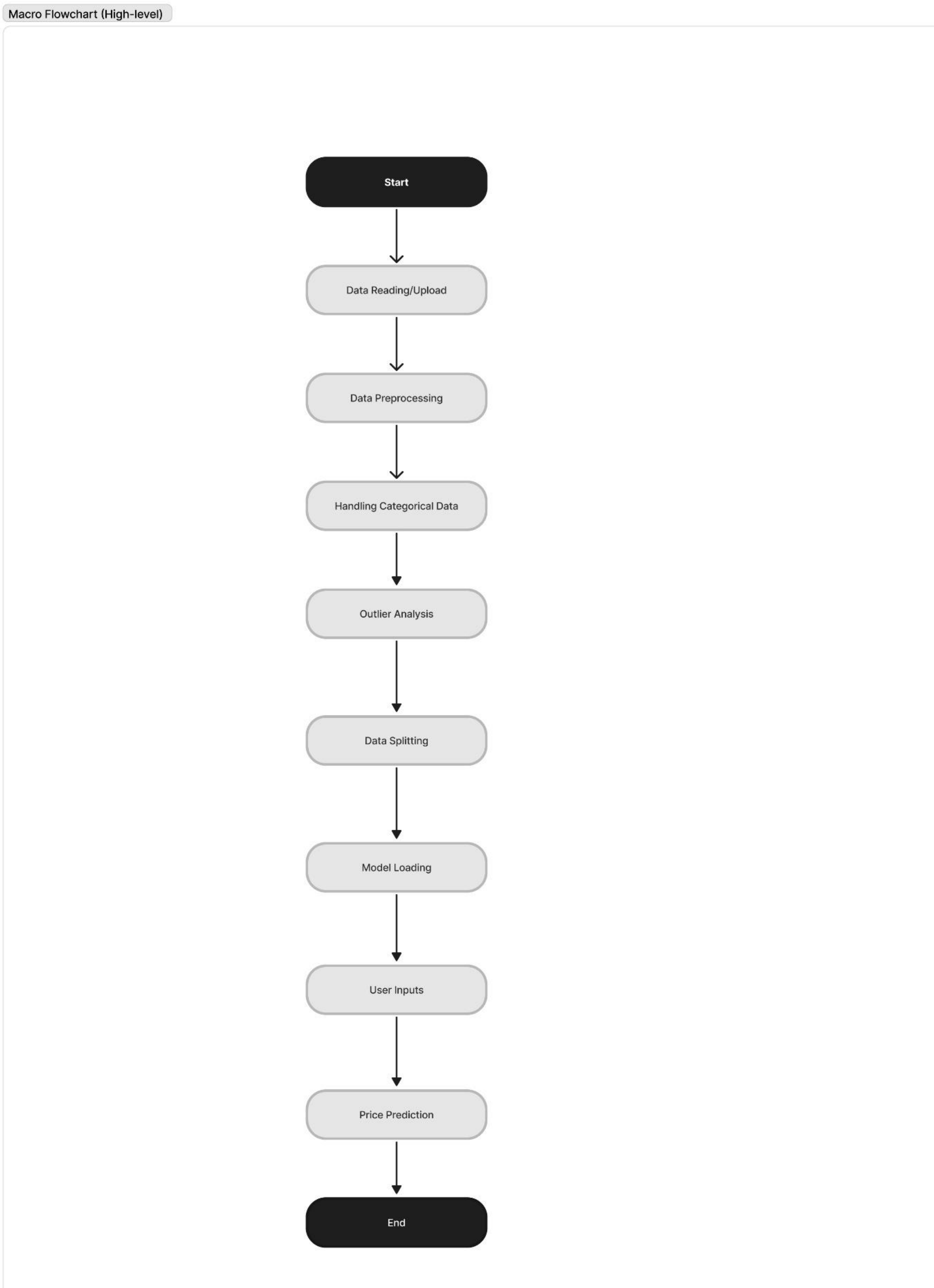


Fig 8.4 Flowchart

Usage: Flowcharts in this project could be used to illustrate various processes, such as user registration, data analysis, or result export. For instance, a flowchart might detail the steps involved in uploading chat data, parsing it, performing sentiment analysis, generating visualizations, and allowing users to export the results.

Flowcharts serve as valuable tools for planning, documentation, and communication among project stakeholders. They provide a visual roadmap of how the "WhatsApp Chat Analyzer" processes and manages data, making it easier to understand and implement the system's functionalities.

CHAPTER 9

IMPLEMENTATION

9.IMPLEMENTATION

PYTHON

Python is a general-purpose language. It has wide range of applications from Machine Learning (like: Pandas, Numpy etc), The syntax of the language is clean and length of the code is relatively short. It's fun to work in Python because it allows you to think aboutthe problem rather than focusing on the syntax.

Why the name Python?

No. It wasn't named after a dangerous snake. Rossum was fan of a comedy series from late seventies. The name "Python" was adopted from the same series "Monty Python's Flying Circus".

Features of Python:

- **Free and open-source:** You can freely use and distribute Python, even for commercial use. Not only can you use and distribute software's written in it, you can even make changes to the Python's source code**.**
- **Portability:** You can move Python programs from one platform to another, and run it without any changes.It runs seamlessly on almost all platforms including Windows, Mac OS X and Linux.
- **Extensible and Embeddable:** Suppose an application requires high performance. You can easily combine pieces of C/C++ or other languages with Python code.
- **A high-level, interpreted language:** Unlike C/C++, you don't have to worry about daunting tasks like memory management, garbage collection and so on.
- **Overview of Implementation:** The "House Price Prediction" project is implemented using Python, a versatile and popular programming language. Python offers a wide range of libraries and frameworks that make it well-suited for this project, including data processing, natural language processing, and web development tools.

- **Data Processing:** Python provides powerful libraries for parsing and structuring Datasets. The pandas library can be used to manage and manipulate data efficiently.
- **Security**: Python's extensive ecosystem includes security libraries and practices to ensure the application's security, such as password hashing and encryption.

SAMPLE CODE

House Price Prediction Using Machine Learning Algorithm—The-Case-of-Hyderabad-India.ipynb:

```
#Importing Libraries

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler, LabelEncoder, OneHotEncoder

from sklearn.impute import SimpleImputer

from sklearn.ensemble import GradientBoostingRegressor

from sklearn.model_selection import train_test_split

from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score


data.head()


data.imfo()


data.describe()


#Dropping Irrelevant columns
```

```
New_Data = data.drop(['active', 'amenities', 'balconies',

       'completeStreetName', 'facing', 'facingDesc',

        'id', 'isMaintenance', 'lift', 'loanAvailable',

         'ownerName','waterSupply',

       'reactivationSource',

        'shortUrl', 'swimmingPool',

        'weight','combineDescription','deposit','furnishingDesc','gym','parking','parkingDesc',

                    'location','localityId','propertyTitle','propertyType','sharedAccomodation']

,axis=1)


# Replace 'None' with 0 in the 'maintenanceAmount' and 'rent_amount' columns

New_Data['maintenanceAmount'].replace('None', 0, inplace=True)

New_Data['rent_amount'].replace('None', 0, inplace=True)


# Convert the 'maintenanceAmount' and 'rent_amount' columns to numeric data

New_Data['maintenanceAmount'] = pd.to_numeric(New_Data['maintenanceAmount'])

New_Data['rent_amount'] = pd.to_numeric(New_Data['rent_amount'])


# Assuming 'object_column' contains strings and 'int_column' contains integ

New_Data['total_price'] = New_Data['maintenanceAmount'].astype(str) + New_Data['rent_amount'].astype(str)


#Concatenating
```

```
New_Data = pd.concat([New_Data.iloc[:, :-1], New_Data['total_price']], axis=1)


New_Data = New_Data.drop(['rent_amount', 'maintenanceAmount'],axis=1)


#Function() to detect Outliers
def detect_outliers_z_score(New_Data):
    threshold = 3
    mean = New_Data.mean()
    std = New_Data.std()
    z_scores = (New_Data - mean) / std
    return (z_scores > threshold) | (z_scores < -threshold)


# Outliers in property_size
outliers= detect_outliers_z_score(New_Data['property_size'])
if not outliers.empty:
    print("Outliers in the column:")
    print(outliers)


plt.figure()
plt.boxplot(New_Data['property_size'])
plt.title('Boxplot for type_bhk')
plt.show()
```

```
Data = New_Data[~New_Data['property_size'].isin(outliers)]

# Print the DataFrame after removing outliers
print("DataFrame after removing outliers:")
print(Data)

# Outliers in property_age
outliers= detect_outliers_z_score(New_Data['property_age'])
if not outliers.empty:
    print("Outliers in the column:")
    print(outliers)

plt.figure()
plt.boxplot(New_Data['property_age'])
plt.title('Boxplot for type_bhk')
plt.show()

Data = New_Data[~New_Data['property_age'].isin(outliers)]

# Print the DataFrame after removing outliers
print("DataFrame after removing outliers:")
```

```
print(Data)


# Outliers in totalFloor
outliers= detect_outliers_z_score(New_Data['totalFloor'])
if not outliers.empty:
    print("Outliers in the column:")
    print(outliers)


plt.figure()
plt.boxplot(New_Data['totalFloor'])
plt.title('Boxplot for type_bhk')
plt.show()


Data = New_Data[~New_Data['totalFloor'].isin(outliers)]


# Print the DataFrame after removing outliers
print("DataFrame after removing outliers:")
print(Data)


# Outliers in type_bhk
outliers= detect_outliers_z_score(New_Data['type_bhk'])
if not outliers.empty:
    print("Outliers in the column:")
```

```
    print(outliers)


plt.figure()
plt.boxplot(New_Data['type_bhk'])
plt.title('Boxplot for type_bhk')
plt.show()


Data = New_Data[~New_Data['type_bhk'].isin(outliers)]


# Print the DataFrame after removing outliers
print("DataFrame after removing outliers:")
print(Data)


# Outliers in bathroom
outliers= detect_outliers_z_score(New_Data['bathroom'])
if not outliers.empty:
    print("Outliers in the column:")
    print(outliers)


plt.figure()
plt.boxplot(New_Data['bathroom'])
plt.title('Boxplot for type_bhk')
plt.show()
```

```
Data = New_Data[~New_Data['bathroom'].isin(outliers)]

# Print the DataFrame after removing outliers
print("DataFrame after removing outliers:")
print(Data)

# Initialize LabelEncoder
label_encoder = LabelEncoder()

# Fit and transform the 'locality' column
New_Data['locality_encoded'] = label_encoder.fit_transform(New_Data['locality'])



# Features and target variable
X = New_Data[['bathroom','locality_encoded', 'property_age', 'property_size', 'totalFloor', 'type_bhk']]
y = New_Data['total_price']

# Initialize the model
model = GradientBoostingRegressor()

# Train the model
```

model.fit(X, y)

Create a test set

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42) # Make predictions on the test set

y_pred = model.predict(X_test)

Calculate evaluation metrics

mae = mean_absolute_error(y_test, y_pred)

mse = mean_squared_error(y_test, y_pred)

rmse = mean_squared_error(y_test, y_pred, squared=False)

r2 = r2_score(y_test, y_pred)

Print the evaluation metrics

print(f"Mean Absolute Error: {mae}")

print(f"Mean Squared Error: {mse}")

print(f"Root Mean Squared Error: {rmse}")

print(f"R-squared: {r2}")

```
# User input
user_inputs = {}


# user_inputs['floor'] = int(input("Enter the floor number: "))
user_inputs['locality'] = input("Enter the locality: ")
user_inputs['property_age'] = int(input("Enter the property age: "))
user_inputs['property_size'] = int(input("Enter the property size: "))
user_inputs['totalFloor'] = int(input("Enter the total floor: "))
user_inputs['type_bhk'] = float(input("Enter the type BHK: "))
user_inputs['bathroom'] = int(input("Enter the number of bathrooms: "))


# Convert user input to DataFrame
input_df = pd.DataFrame([user_inputs])


# Encode the 'locality' column
input_df['locality_encoded'] = label_encoder.transform(input_df['locality'])


# Make predictions
predicted_prices = model.predict(input_df[['bathroom','locality_encoded', 'property_age', 'property_size', 'totalFloor', 'type_bhk']])


# Display the predicted total price
print(f"The predicted total price is: {predicted_prices[0]}")
```

```
#to check if a desired location exists in dataset change the 'narapally' with desired location name
if 'Narapally' in New_Data['locality'].values:
    print("narapally is present in the 'locality' column.")
else:
    print("narapally is not present in the 'locality' column.")
```

CHAPTER 10

TESTING

10.TESTING

Purpose of Testing:

Testing is a critical phase in the software development process. It ensures that the "House Price Prediction Using Machine Learning Algorithm—The-case-of-Hyderabad-India" functions correctly, meets user requirements, and operates reliably. Testing also helps identify and rectify any issues or bugs in the system.

Types of Testing:

1.Unit Testing:

Description: In unit testing, individual components or functions of the system are tested in isolation.

Purpose: To verify that each component performs as expected and produces the correct output.

Implementation: Python's built-in unit test library or third-party testing frameworks like py test can be used for unit testing.

2.Integration Testing:

Description: Integration testing focuses on how different modules or components of the system work together.

Purpose: To ensure that data flows smoothly between components and that they interact correctly.

Implementation: Python's testing frameworks, along with mock objects, can be used for integration testing.

3.Functional Testing:

Description: Functional testing evaluates the entire application's functionality.

Purpose: To verify that the software meets user requirements and performs the intended tasks.

Implementation: Automated testing tools like Selenium may be used for web application functional testing. Test cases are designed to cover user scenarios such as registration, data upload, analysis, and result export.

4.Performance Testing:

Description: Performance testing assesses the system's responsiveness and stability under various conditions, such as high loads or concurrent users.

Purpose: To ensure that the application can handle the expected workload without performance degradation or crashes.

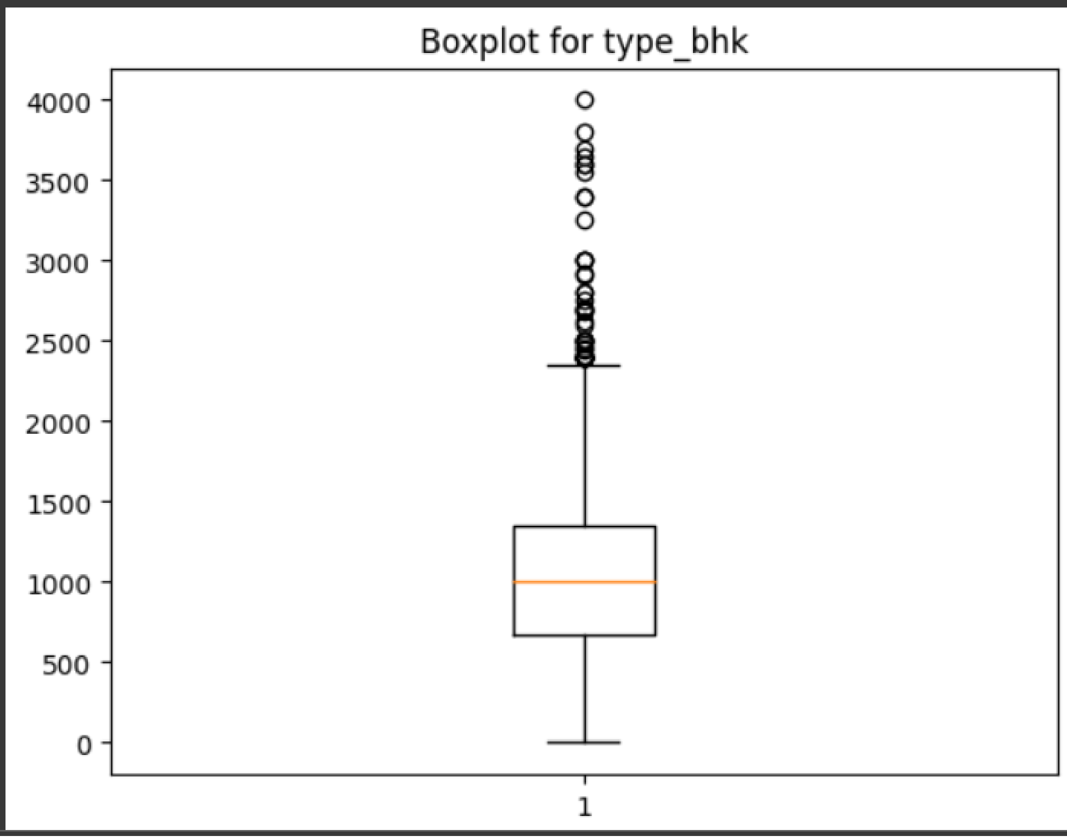
Implementation: Python performance testing libraries like locust can be employed to simulate user loads and analyze, Predicts the system performance.

CHAPTER 11

SCREENSHOTS

12.SCREENSHOTS

```
plt.figure()
plt.boxplot(New_Data['property_size'])
plt.title('Boxplot for type_bhk')
plt.show()
```



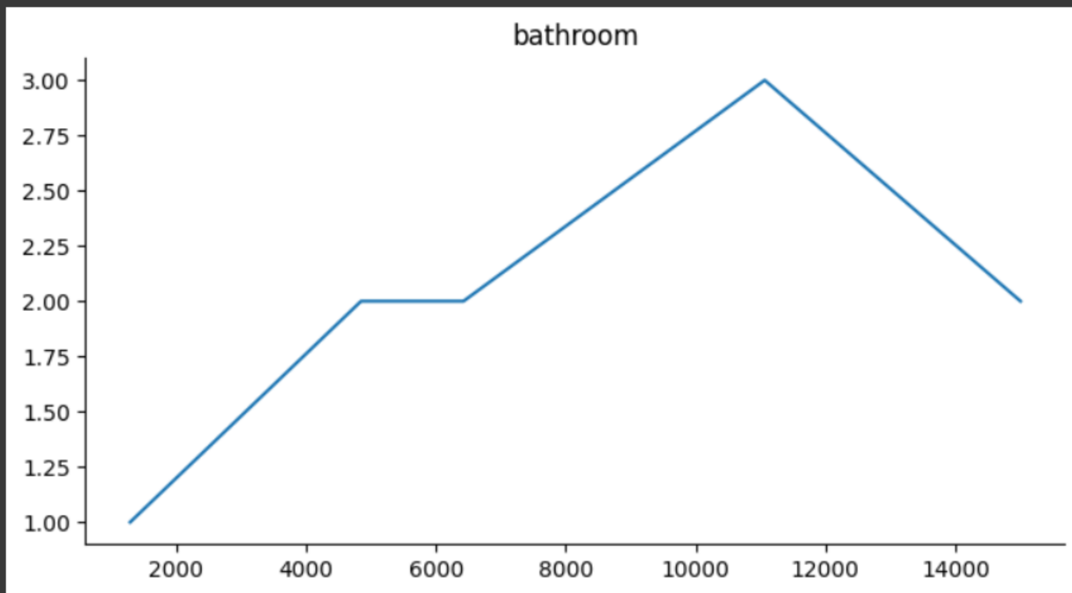
```
7]  data.describe()
```

	bathroom	combineDescription	deposit	floor	property_age	property_size	rent_amount	totalFloor	weight
count	2000.000000	0.0	2.000000e+03	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	0.0
mean	1.913500	NaN	3.286520e+04	2.110000	3.731000	1058.107000	14754.499000	3.913000	NaN
std	0.765709	NaN	5.231889e+04	2.347905	3.409612	532.732772	9206.112829	3.704766	NaN
min	1.000000	NaN	1.000000e+00	0.000000	-1.000000	0.000000	0.000000	0.000000	NaN
25%	1.000000	NaN	1.600000e+04	1.000000	1.000000	671.250000	8500.000000	2.000000	NaN
50%	2.000000	NaN	2.400000e+04	2.000000	3.000000	1000.000000	12000.000000	3.000000	NaN
75%	2.000000	NaN	4.000000e+04	3.000000	5.000000	1350.000000	18000.000000	5.000000	NaN
max	8.000000	NaN	2.000000e+06	24.000000	10.000000	4000.000000	95000.000000	33.000000	NaN

```
0s  [21]  New_Data.head()
```

	bathroom	floor	locality	property_age	property_size	totalFloor	type_bhk	total_price
15005	2	2	Gachibowli	5	1000	3	2.0	014000
11064	3	1	Sanjeeva Reddy Nagar	7	1800	4	3.0	130000
6423	2	2	Kondapur	0	1250	5	2.0	180023000
4845	2	2	Miyapur	10	1700	5	3.0	200022000
1288	1	1	Parsigutta	3	500	2	1.0	05300

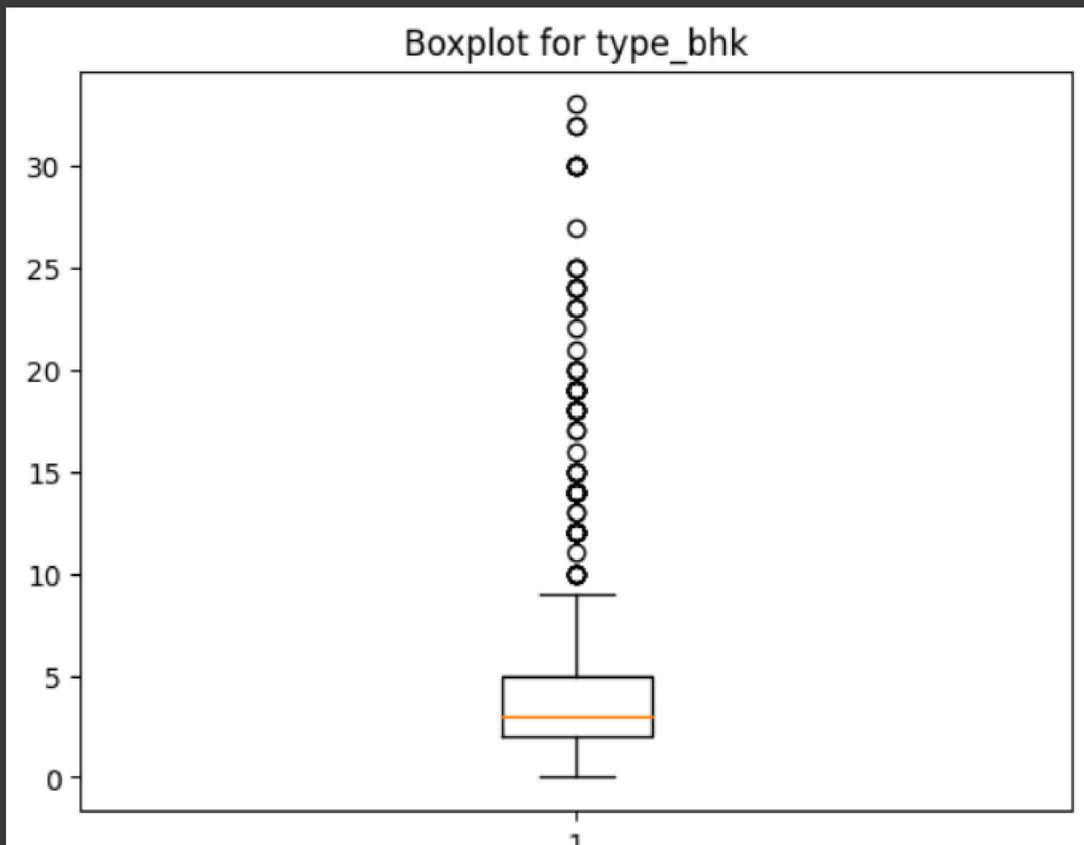
```
from matplotlib import pyplot as plt
_df_10['bathroom'].plot(kind='line', figsize=(8, 4), title='bathroom')
plt.gca().spines[['top', 'right']].set_visible(False)
```



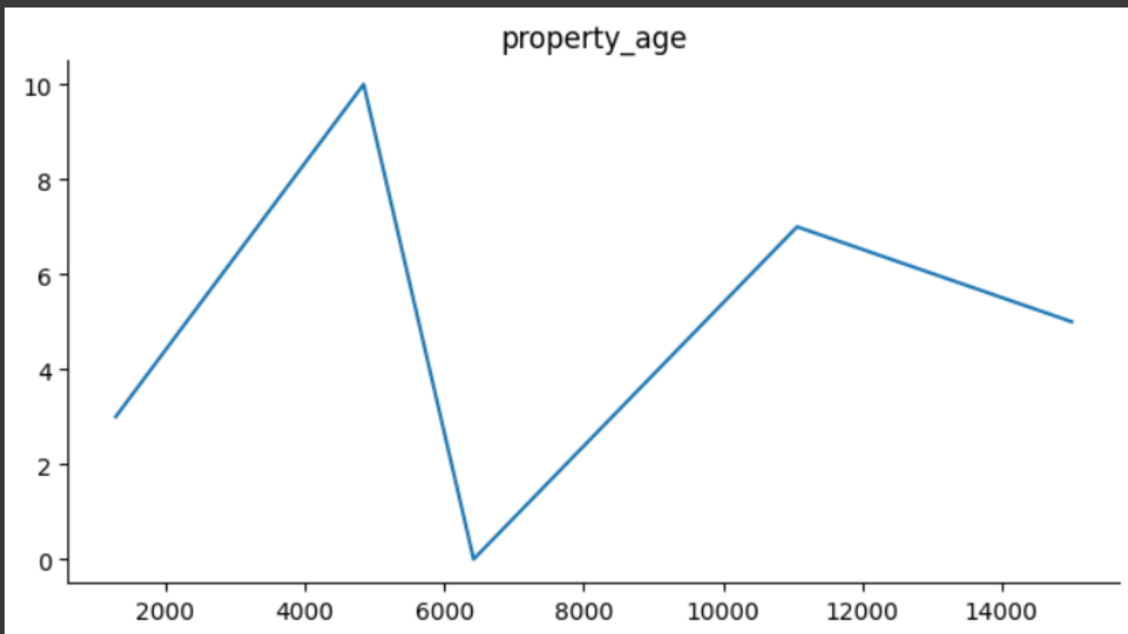

```
[44] # Train the model
     model.fit(X, y)
```

```
▾ GradientBoostingRegressor
GradientBoostingRegressor()
```

```
[31] plt.figure()
     plt.boxplot(New_Data['totalFloor'])
     plt.title('Boxplot for type_bhk')
     plt.show()
```



```python
from matplotlib import pyplot as plt
_df_12['property_age'].plot(kind='line', figsize=(8, 4), title='property_age')
plt.gca().spines[['top', 'right']].set_visible(False)
```



CHAPTER 12

CONCLUSION

12. CONCLUSION

In conclusion, the envisioned House Price Prediction Application for the Hyderabad real estate market stands as a robust and dynamic platform leveraging advanced machine learning techniques. By exploring algorithms such as Random Forest and Gradient Boosting, the project aims to enhance prediction accuracy and reliability, acknowledging the intricate patterns within the housing market.

The incorporation of real-time market insights ensures that users receive the most up-to-date information on housing prices and market dynamics. This feature enables the application to adapt swiftly to changing real estate conditions, providing users with timely and relevant insights for informed decision-making.

Furthermore, considering the potential for global expansion, the project opens avenues for future growth. The prospect of extending the application to cover housing markets in other regions or countries reflects a commitment to catering to a diverse international audience. This expansion aligns with the vision of creating a versatile tool adaptable to different market trends and dynamics unique to each locale.

The external interface requirements, including user interfaces, hardware interfaces, and software interfaces, are carefully designed to ensure user-friendly interactions, accessibility across various platforms, and seamless integration with external components. These interfaces contribute to the overall usability, performance, and security of the application.

As the project progresses, it is essential to draw insights from reputable sources such as machine learning documentation, real estate research papers, and datasets specific to the Hyderabad market.

CHAPTER 13

FUTURE SCOPE

13.FUTURE SCOPE

Advanced Machine Learning Techniques:

To enhance prediction accuracy and robustness, the project should explore advanced machine learning algorithms such as Random Forest and Gradient Boosting. These techniques are known for their ability to capture intricate patterns and relationships within datasets, particularly beneficial for predicting housing prices with greater precision.

Real-time Market Insights:

Integrating real-time data updates and market trend analysis into the application is crucial for providing users with the most current information on housing prices and market dynamics. By implementing real-time features, the application can adapt to the dynamic nature of real estate markets, ensuring users have access to the latest insights for informed decision-making.

Global Expansion:

Considering the success and applicability of the project, future endeavors should explore the possibility of expanding the application to cover housing markets in other regions or countries. This expansion would not only broaden the project's user base but also cater to a diverse international audience, accommodating different market trends and dynamics unique to each region.

These advanced features not only contribute to the immediate improvement of the application but also set the stage for its evolution and adaptation to changing market landscapes. The incorporation of advanced machine learning techniques, real-time insights, and global expansion positions the project for continued success and relevance in the ever-evolving real estate domain

CHAPTER 14

REFERENCES

14.REFERENCES

Machine Learning Algorithms:Explore documentation and tutorials from scikit-learn: https://scikit-learn.org/stable/documentation.html

2.Look for articles on real-time data analysis and market trends on platforms like Medium's Towards Data Science: https://towardsdatascience.com/

3.Research global housing market trends through academic journals, such as the Journal of Real Estate Economics: https://www.journals.elsevier.com/journal-of-real-estate-economics

4.Seek academic papers or reports specifically focused on the real estate market in Hyderabad, India. This could include local research institutions, journals, or government publications.

5.Explore GitHub repositories for machine learning in real estate: https://github.com/topics/machine-learning-real-estate