A PROJECT REPORT

On

“REAL ESTATE PROPERTY VALUE ESTIMATION USING ML”

Submitted in partial fulfillment of the requirements to

KKR & KSR INSTIUE OF TECHNOLOGY & SCIENCES

For the award of the degree

BACHELOR** of **TECHNOLOGY

In

CSE -DATA SCIENCE

Submitted by

YADLAPALLI DURGA PRASAD	(22JR5A4412)
VIDELA PAVAN KALYAN	(21JR1A44C9)
YAMARAJULA TARUN	(21JR1A44D0)
MOHAMMED KHALID ZAID	(21JR1A44A4)
MALAPATI MANOHAR	(21JR1A44A0)
MALLEBOINA TRINADH	(21JR1A44A1)

Under the guidance of

Dr. B. Bhanu Prakash,
MTech ., PhD., PDF(Malaysia)
Head of the Department, CSD

DEPARTMENT OF CSE – DATA SCIENCE

KKR & KSR INSTITUTE OF TECHNOLOGY AND SCIENCES

Autonomous

(APPROVED BY AICTE AND PERMANENTLY AFFILIATED TO JNTUK)
Accredited by NBA and NAAC with ‘A’ Grade
Vinjanampadu (V), Vatticherukuru (M), GUNTUR – 522017
April 2023

DEPARTMENT OF CSE – DATA SCIENCE

KKR & KSR INSTITUTE OF TECHNOLOGY AND SCIENCES

Autonomous

(APPROVED BY AICTE AND PERMANENTLY AFFILIATED TO JNTUK)
Accredited by NBA and NAAC with 'A' Grade
Vinjanampadu (V), Vatticherukuru (M), GUNTUR – 522017
APRIL – 2023

CERTIFICATE

his is to certify that this project work titled "REAL ESTATE PROPERTY VALUE ESTIMATION USING ML" is the bonafidework of **YADLAPALLI DURGA PRASAD(22JR5A4412)**, **VIDELA PAVAN KALYAN(21JR1A44C9), YAMARAJULA TARUN (21JR1A44D0), MOHAMMED KHALID ZAID(21JR1A44A4), MALAPATI MANOHAR (21JR1A44A0) and MALLEBOINA TRINADH (21JR1A44A1)** who carried out the work under supervision and submitted in the partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in CSE-DATA SCIENCE from **KKR & KSR INSTITUTE OF TECHNOLOGY AND SCIENCES.**

HEAD OF THE DEPARTMENT PROJECTGUIDE

EXTERNAL EXAMINER

DECLARATION

We hereby inform that this main project entitled "REAL ESTATE PROPERTY VALUE ESTIMATION USING ML" has been Carried out and submitted in partial fulfillment for the award to the degree of **Bachelor of Technology in Computer Science and Engineering** to **KKR & KSR INSTITUTE OF TECHNOLOGY & SCIENCES** under the guidance of Dr.B.BHANU PRAKASH, Head of the Department, CSE-DATA SCIENCE. The work embodied in this project work is original and has not been submitted in part or full for any degree of this or any degree of any other university.

YADLAPALLI DURGA PRASAD	22JR5A4412
VIDELA PAVAN KALYAN	21JR1A44C9
YAMARAJULA TARUN	21JR1A44D0
MOHAMMED KHALID ZAID	21JR1A44A4
MALAPATI MANOHAR	21JR1A44A0
MALLEBOINA TRINADH	21JR1A44A1

ACKNOWLEDGEMENT

We would like to express our profound gratitude towards **Dr. B.Bhanu Prakash Head of the Department,** CSE– DATA SCIENCE**,** who played a supervisory role to utmost perfection, enabled us to seek through our B.Tech mini project and for guiding as an internal guide methodically and meticulously.

We are highly indebted to **Dr.B. Bhanu prakash**, **Head of the Department**, CSE-DATA SCIENCE for providing us all the necessary support. We are very much thankful to the **college management for** their continuous support and facilities provided.

We render our deep sense of gratitude to **Dr. P. BABU, Principal,** for permitting us to carry out our main project works. We would like to express our sincere thanks to Computer Science and Engineering staff for lending us their time to help us and complete the work successfully.

We would also like to thank our staff, parents and friends for their enduring encouragement and assistance whenever required.

INDEX

Institute Vision and Mission

INSTITUTION VISION

To produce eminent and ethical Engineers and Managers for society by imparting quality professional education with emphasis on human values and holistic excellence.

INSTITUTION MISSION

- To incorporate benchmarked teaching and learning pedagogies in curriculum.
- To ensure all round development of students through judicious blend of curricular, co- curricular and extra-curricular activities.
- To support cross-cultural exchange of knowledge between industry and academy.
- To provide higher/continued education and researched opportunities to the employees of the institution.

Vision of the Department

To achieve academic excellence and to build the leaders with escalating multi – skilled professionals in Data Science Engineering with global competence empowered by technical expertise with in-depth knowledge, innovative research and lifelong learning, promoting employability, higher education with socio-ethical and entrepreneurial values for development.

Mission of the Department

To impart quality technical education and to develop the learners as leaders in Data Science Engineering with fundamental engineering principles and professionally deft, innovative research capabilities to lead and to use technology for the progress of mankind. To changing technological environment by providing the high quality instruction, faculty, modern teaching, socio-ethical and entrepreneurial values as the inner strength for rural development.

Mission 1: Quality Education
Mission 2: Professional career
Mission 3: Entrepreneur and Research.

Program Specific Outcomes (PSOs)

PSO1:

Ability to design and develop applications using various Data Science tools..

PSO2:

The learners will be able to develop the knowledge of the competitive environment in success of globally acclaimed tests like GRE, TOEFL, ILTES, IES, GMAT, CAT, PSUs, and GATE etc.

Program Educational Objectives (PEOs)

1. **Preparation:** The learners of Data Science Engineering can be able to apply the knowledge of mathematics, Applied Science, Computing ,Basic Engineering field to identify, analyze, formulate, design, and develop the practical solutions for industry and academics.

2. **Core Competence:** To enable the learners with core curriculum knowledge in theory and practical's of Data Science Engineering to develop the innovative skills in design, simulation, investigation of complex problems, critical reasoning, development & testing knowledge for offering solutions to real life.

3. **Breadth Knowledge:** To provide the learners with breadth knowledge to build the Data Science Engineering professionals to lead the team work and skills to develop the abilities to communicate, lifelong learning and aptitude of project management, finance with entrepreneurial values.

4. **Leading Professional knowledge:** To practice using a system of multi-faceted disciplinary approach to develop R&D skills by MOUs with premier industries and institutions interacting with training sessions and industrial visits to the learners to have awareness in latest trends of Cloud Computing, Cyber Forensics, Hadoop, Big data, Android etc… in concurring the modern software's and tools of Data Science to lead the escalating needs of society.

5. **Career Improvement and ethics:** To build the learners with the knowledge of real time requirement of cutting edge technologies by intellectually adapt to promote employability, higher education and imbibing ethical, social and eco-friendly.

Program Outcomes

1. **Engineering Knowledge:** Apply the knowledge of mathematics, science, engineering
Fundamentals and an engineering specialization to the solution of complex engineering problems.

2. **Problem Analysis:** Identify, formulate, research literature and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences and engineering sciences.

3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety and the cultural, societal and environmental considerations.

4. **Conduct investigations of complex problems:** Use research based knowledge and research methods including design of experiments, analysis and interpretation of data and synthesis of the information to provide valid conclusions.

5. **Modern tool usage:** Create, select and apply appropriate techniques, resources and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering

practice.

7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts and demonstrate the knowledge of and the need for sustainable development.

8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. Individual and team work: Function effectively as an individual and as a member or leader in diverse teams and in multidisciplinary environments.

10. Communication: Communicate effectively on complex engineering activities with the engineering community and with the society at large, such as, being able to comprehend and write effective reports and design documentations and presentations and also give clear instructions.

11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and as a leader in a team, to manage projects and in multidisciplinary environments.

12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Project Course Outcomes

CO425.1: Analyze the System of Examinations and identify the problem.

CO425.2: Identify and classify the requirements.

CO425.3: Review the related Literature.

CO425.4: Design and modularize the project.

CO2425.5: Construct, integrate, test and implement the project.

CO425.6: Prepare the project Documentation and present the report using appropriate method.

Course Outcomes - Program Outcomes mapping

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**PSO1**	**PSO2**	**PSO3**
C425.1	✓	✓		✓		✓	✓						✓		
C425.2	✓	✓		✓		✓			✓	✓	✓		✓	✓	✓
C425.3		✓			✓							✓	✓		✓
C425.4	✓	✓	✓	✓	✓		✓	✓	✓	✓			✓		✓
C425.5			✓		✓		✓		✓		✓	✓	✓	✓	
C425.6						✓				✓		✓		✓	

Course Outcomes – Program Outcome correlation

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**PSO1**	**PSO2**	**PSO3**
C425.1	3	2		2		2	2						3		
C425.2	3	2		2		2			1	2	1		2	2	3
C425.3		3			2							2	1		3
C425.4	2	2	3	2	2		1	1	2	2			3		2
C425.5			3		2		2		1		2	2	3	2	
C425.6						3				2		2		3	

3:High 2:Medium 1: Low

CO-PO Mapping with Reasons:

1. **CO425.1** is mapped with PO1, PO2 and PO4, PO6, PO7 as basic knowledge of Engineering and problem Analysis activities are highly essential to conduct examinations on existing systems which have been using in industries as a part of and to define the problem of proposed system.
2. **CO425.2** is mapped with PO1, PO2, PO4 and PO6, PO9, PO10, PO11 as for identification, gathering analysis and classification of requirements for the proposed system, basic knowledge of engineering and Analysis steps along with complex problem analysis through the efforts of team work in order to meet the specific needs of the customer.
3. **CO425.3** is mapped with PO2, PO5 and PO12 as to conduct the literature review and to examine the relevant systems to understand and identify the merits and demerits of each too enhance and develop the proposed as per the need.
4. **CO425.4** is mapped with PO1, PO2, PO3, PO4, PO5 and PO7, PO8, PO9, PO10 because modularization and design of the project is needed after requirements elicitation. For modularization and design of the project, Basic knowledge of Engineering, Analysis capabilities, Design skills and communication is needed between team members as different modules are designed individually before integration.
5. **CO425.5** is mapped with PO3, PO5, PO7, PO9, PO11 and PO12 as to construct the project latest technologies are needed. The development of project is done individually and in groups with well-defined communication by using the engineering and management principles.
6. **CO425.6** is mapped with PO6, PO10 and PO12 because during and after completion of the project, documentation is needed along with proper methods of presentation through understanding and application of engineering and management principles, which in turn needs well defined communication between the team members with all the ethical values. Even the project development team defines the future enhancements as a part of the project development after identifying the scope of the project.

CO-PSOs Mapping with Reasons:

1. **CO425.1** is mapped with **PSO1** as examining of existing systems and identification of the problem is a part of Application Development activity and identification of evolutionary changes in latest technologies.

2. **CO425.2** is mapped with **PSO1, PSO2** and **PSO3**as identifying and classifying the requirements is a part of Application development and evolutionary computing changes and also follows ethical principles.

3. **CO425.3** is mapped with **PSO1, PSO3** as review of literature is a part of application development activity by recognizing the computing technologies and their evolutionary changes.

4. **CO425.4** is mapped with **PSO1, PSO3** because modularization and logical design is also a part of Application development and follows computing changes using Deep learning technology.

5. **CO425.5** is mapped with **PSO1, PSO2** as Testing, Development and Integration of project activities are part of Application development and follows ethical principles.

6. **CO425.6** is mapped with **PSO2** as for project documentation and presentation; the project team members apply the professional and leadership qualities

Name of Course from which Principles are applied in this project	Description of the application	Attained PO	
C324.1 C325.1 C425.1	Gathering the requirements and define the problem, plan to develop the "Property Value Estimation"	PO1,PO3	
C221.1 C221.2 C221.3	Each and every requirement is critically analysed, the process model is identified and divide into several modules.	PO2,PO3	
C313.1 C313.3 C313.4	Logical design is done by using the unified modelling language which involves individual team work.	PO3,PO5,PO6,PO9	
C324.6 C221.4	Each and every module is tested, integrated, and evaluated in our project.	PO1,PO5	
C221.1 C221.2 C222.3	Documentation is done by all four members in the form of a group.	PO10	
C221.1 C221.2 C221.3	Each and every phase of the work in group is presented periodically.	P09,PO10,PO11,P012	
C221.6 C325.1 C325.3	Implementation is done and the project will be handled by the administrator and in future updates in our project can be done by the administrator.	PO3,PO5,PO10,PO11	
C414.5	The physical design is done by using Prophecy Estates App inventor.	PO5,PO10,P	

ABSTRACT:

The Real Estate Property Value Estimation ML Project represents a groundbreaking initiative aimed at revolutionizing property valuation in the real estate industry. Utilizing cutting-edge machine learning algorithms and a multidisciplinary approach, this project addresses the inherent challenges of property valuation, providing users with accurate and reliable estimations tailored to their specific needs.

Key features of the project include the leveraging of machine learning techniques to analyze diverse real estate data, including property features, location data, market trends, and historical sales records. Through sophisticated data preprocessing and analysis, the project generates precise property valuations, empowering users to make informed decisions regarding real estate transactions.

Moreover, the project offers customization options, allowing users to adjust valuation parameters to align with their individual preferences and requirements. The user-friendly interface enhances accessibility and facilitates seamless navigation, ensuring a positive user experience.

Overall, the Real Estate Property Value Estimation ML Project showcases the potential of machine learning technology to revolutionize traditional processes within the real estate sector. By providing accurate valuations and empowering users with actionable insights, this project sets a new standard for property valuation methodologies, driving innovation and efficiency in the industry.

UNIT- I

INTRODUCTION

1.1 INTRODUCTION OF THE PROJECT:

In an ever-evolving real estate landscape, the accurate valuation of properties stands as a cornerstone for informed decision-making. However, traditional valuation methods often struggle to adapt to the complexities of modern markets, leading to uncertainties and inefficiencies in property transactions. In response to these challenges, the Real Estate Property Value Estimation ML Project emerges as a pioneering initiative, leveraging the power of machine learning to revolutionize property valuation.

This project represents a collaborative effort driven by a multidisciplinary team of experts, with the aim of providing users with accurate and reliable property valuations tailored to their specific needs. By harnessing cutting-edge machine learning algorithms, the project analyzes vast amounts of real estate data, including property features, location data, market trends, and historical sales records. Through sophisticated data preprocessing and analysis techniques, the project generates precise valuations, empowering users to make well-informed decisions in the dynamic real estate market.

1.2 EXISTING SYSTEM:

the existing systems for property valuation in the real estate industry were characterized by manual processes, limited data integration, lack of customization options, inefficiency, and complexity of use. These limitations underscored the need for a more advanced and innovative approach to property valuation, leading to the development of the Real Estate Property Value Estimation ML application.

1.3 PROBLEMS OF THE EXISTING SYSTEMS:

Subjective Valuation Methods:

Existing property valuation systems in the real estate industry often rely on subjective assessments by appraisers or agents. This subjectivity can lead to inconsistencies and variations in valuation results, as different individuals may interpret property characteristics differently.

Limited Data Utilization:

Many existing systems lack the capability to effectively utilize the vast amount of real estate data available. This includes property features, market trends, historical sales data, and demographic information. Without leveraging this data effectively, valuation results may lack accuracy and relevance.

Time-Consuming Processes:

Traditional property valuation methods can be time-consuming and labor-intensive. Manual data collection, analysis, and documentation processes contribute to delays in providing valuation reports to clients. This inefficiency can hinder timely decision-making in real estate transactions.

Lack of Transparency:

The valuation methodologies used in existing systems may not always be transparent to users. This lack of transparency can lead to distrust and uncertainty among clients regarding the accuracy and reliability of the valuation results.

Inaccurate Comparables:

Existing systems may rely heavily on comparable sales data to estimate property values. However, inaccurate or outdated comparables can lead to incorrect valuation estimates. Without access to reliable and up-to-date data, valuation results may not reflect the true market value of properties.

1.4 PROPOSED SYSTEM:

The proposed Real Estate Property Value Estimation ML Project aims to address the limitations of existing property valuation systems by leveraging advanced machine learning techniques and comprehensive data analysis. The proposed system offers a more accurate, efficient, and transparent approach to property valuation, empowering users to make informed decisions in the real estate market.

The proposed system utilizes state-of-the-art machine learning algorithms, such as regression models, neural networks, and ensemble methods, to analyze diverse real estate data and generate precise property valuations. By leveraging advanced statistical techniques and algorithmic learning, the system can identify complex patterns and relationships within the data, leading to more accurate valuation estimates.

BENEFITS OF THE PROPOSED SYSTEM:

- Increased Accuracy.
- Enhanced Efficiency.
- Customized Valuation Criteria.
- Transparency and Documentation.
- Real-Time Valuations.
- Cost Savings.

UNIT -II

ANALYSIS

The analysis is defined as a detailed examination of the elements or structure of something.

2.1 REQUIREMENT ANALYSIS:

The process to gather the software requirements from clients, analyze and document them is known as requirements engineering or requirements analysis. The goal of requirement engineering is to develop and maintain sophisticated and descriptive 'System/Software Requirements Specification' documents. It is a four-step process generally, which includes –

- Feasibility Study four-step
- Requirements Gathering
- Software Requirements Specification
- Software Requirements Validation

The basic requirements of our project are:

- Python installed
- Research Papers
- Datasets
- Accuracy calculation

2.1.1 FUNCTIONAL REQUIREMENT ANALYSIS :

Functional requirements explain what has to be done by identifying the necessary task, action, or activity that must be accomplished. Functional requirements analysis will be used as the top-level functions for functional analysis.

2.1.2 USER REQUIREMENTS ANALYSIS :

User Requirements Analysis is the process of determining user expectations for a new or modified product. These features must be quantifiable, relevant, and detailed. The main user requirements of our project are as follows:

- Internet Facility/ LAN Connection

- CPU i5+
- RAM 8 or 16 GB
- Memory 1GB

2.1.3 NON-FUNCTIONAL REQUIREMENTS ANALYSIS :

Non-functional requirements describe the general characteristics of a system. They are also known as quality attributes. Some typical non-functional requirements are Performance, Response Time, Throughput, Utilization, and Scalability.

2.1.4 SYSTEM REQUIREMENTS ANALYSIS :

Hardware Requirements:

System: Intel Core i3 2.4 GHz.

Hard Disk: 500 GB.

RAM: 4GB.

2.2 MODULES DESCRIPTION:

User Module: It is the place where user can access our website to use the services regarding diabetes and parkinson’s disease it provides the description of the website.

System Module: The task which we are done in the system is shown by the system module.

2.3 FEASIBILITY STUDY:

Feasibility Study is a high-level capsule version of the entire process intended to answer a number of questions like What is the problem? Is there any feasible solution to the given problem? Is the problem even worth solving? A feasibility study is conducted once the problem is clearly understood. A feasibility study is necessary to determine that the proposed system is Feasible by considering the technical, Operational, and Economical factors. By having a detailed feasibility study the management will have a clear-cut view of the proposed system. A well-designed feasibility study should provide a historical background of the business or project, the operations and management, marketing research and policies, financial data, legal requirements, and tax obligations. The following feasibilities are considered for the project in order to ensure that the project is variable and it does not have any major obstructions. Feasibility study encompasses the following things:

- Technical Feasibility
- Operational Feasibility
- Behavioural feasibility

In this phase, we study the feasibility of all proposed systems And pick the best feasible solution for the problem. The feasibility is studied based on three main factors as follows:

2.3.1 TECHNICAL FEASIBILITY:

In this step, we verify whether the proposed systems are technically feasible or not. i.e., all the technologies required to develop the system are available readily or not. Technical Feasibility determines whether the organization has the technology and skills necessary to carry out the project and how this should be obtained. The system can be feasible because of the following grounds.

- All necessary technology exists to develop the system
- This system is flexible and can be expanded further
- This system can give a guarantee of accuracy, ease of use, and reliability
- Our project is technically feasible because all the technology needed for our project is readily available.

2.3.2 OPERATIONAL FEASIBILITY:

In this step, we verify different operational factors of the proposed systems like manpower, time, etc., whichever solution uses less operational resources, is the best operationally feasible solution. The solution should also be operationally possible to implement. Operational Feasibility determines if the proposed system satisfied user objectives could be fitted into the current system operation. The present system of predicting can be justified as operationally feasible based on the following grounds.

- The methods of processing and presentation are completely accepted by the clients since they can meet all user requirements.
- The clients have been involved in the planning and development of the system.
- The proposed system will not cause any problem under any circumstances.

Our project is operationally feasible because the time requirements and personnel requirements are satisfied. We are a team of four members and we worked on this project for three working months.

2.3.3 BEHAVIORAL FEASIBILITY:

This device will help people to save time. As there will be no wastage of time, the user will be satisfied.

2.4 PROCESS MODEL USED:

The iterative model is a software development approach that breaks down the project into smaller, manageable iterations or cycles. Each iteration involves a subset of the project's overall features and functionalities, allowing for incremental development and continuous improvement. Here's how the iterative model can be applied to the Real Estate Property Value Estimation ML Project:

1. **Iterative Development**: The development of the Real Estate Property Value Estimation ML Project follows an iterative approach, where each iteration focuses on implementing specific features or components of the system. For example, the first iteration may focus on data collection and preprocessing, while subsequent iterations may address model development, user interface design, and integration with external data sources.
2. **Incremental Delivery**: With each iteration, the project team delivers incremental value to users by adding new features or improving existing ones. This incremental delivery allows users to provide feedback early in the development process, ensuring that the final product meets their needs and expectations.
3. **Feedback Loop**: The iterative model emphasizes the importance of feedback from users and stakeholders throughout the development lifecycle. As each iteration is completed and delivered, users have the opportunity to provide feedback on the functionality, usability, and accuracy of the system. This feedback informs subsequent iterations, allowing the team to make adjustments and improvements based on user input.
4. **Continuous Improvement**: The iterative model promotes continuous improvement by allowing the project team to refine and enhance the system with each iteration. As feedback is incorporated and new features are implemented, the system evolves over time to better meet the needs of users and adapt to changing requirements and market conditions.
5. **Risk Management**: By breaking down the project into smaller iterations, the iterative model helps mitigate risks associated with complex software development projects. Risks are addressed incrementally, allowing the team to identify and mitigate issues early in the development process before they escalate into larger problems.

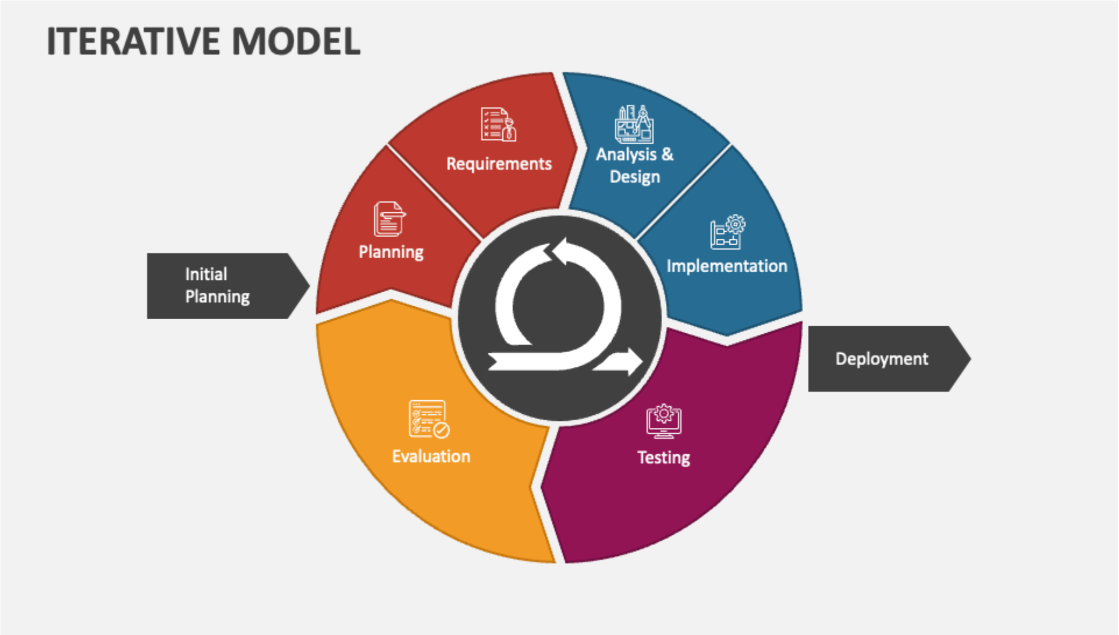


Fig. Iterative model

2.5 SOFTWARE AND HARDWARE REQUIREMENTS:

SOFTWARE REQUIREMENTS:

Operating system: Windows 11.

Coding language: Python.

IDE: VS code.

HARDWARE REQUIREMENTS:

System: Intel Core i3 2.4 GHz.

Hard Disk: 500 GB.

RAM: 4GB.

2.6 SRS SPECIFICATION:

Software Requirements Specification (SRS) – a requirements specification for a software system- is a complete description of the behaviour of a system to be developed. It includes a set of cases that describe all the interactions users will have with the software. In addition to use cases, the SRS also contains non-functional requirements. Non-functional requirements are requirements that impose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints).

for analyzing the business needs of their clients and stakeholders to help identify business problems and propose solutions. Projects are subject to three sorts of required elements. Business requirements describe in business terms what must be delivered or accomplished to provide value.

- Product requirements describe properties of a system or product (which could be one of several ways to accomplish a set of business requirements.)
- Process requirements describe activities performed by the developing organization. For instance, process requirements could specify methodologies that must be followed, and constraints that the organization must obey.

Product and process requirements are closely linked. Process requirements often specify the activities that will be performed to satisfy a product requirement. For example, a maximum development cost requirement (a process requirement) may be imposed to help achieve a maximum sales price requirement (a product requirement) a requirement that the product is maintainable (a product requirement) often is addressed by imposing requirements to follow particular development styles. A system engineering requirement can be a description of what a system must do, referred to as a Functional Requirement. This type of requirement specifies something that the delivered system must be able to do. Another type of requirement specifies something about the system itself, and how well it performs its functions. Such requirements are often called Non functional requirements, or ‘Performance requirements’ or ‘Quality of service requirements’. Examples of such requirements include usability, availability, reliability, supportability, testability, and maintainability.

A collection of requirements define the characteristics or features of the desired system. A ‘good’ list of requirements as far as possible avoids saying how the system should implement the requirements, leaving such decisions to the system designer. Specifying how the system should be implemented is called “implementation bias” or “solution engineering”. However, implementation constraints on the solution may validly be expressed by the future owner, for example for required interfaces to external systems; for interoperability with other systems; and for commonality with other owned products.

Functional requirements:

The Functional Requirements Specification gives the operations and activities that a system must be able to perform. Functional requirements should include functions performed by specific screens, outlines of workflows performed by the system, and other business or compliance requirements the system must meet. It also depends upon the type of software, expected users, and the type of system where the software is used. Some constraints were given as input. This was given as input to code which predicts the disease with accuracy. The output was in the form of a predicted result.

Non-functional requirements:

In systems engineering, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. They are contrasted with functional requirements that define specific behavior or functions. The nonfunctional requirements can be considered as quality attributes of a system.

Performance: The time required to predict the disease.

Reliability: The system should be 90% reliable. Since it may need some maintenance or preparation for some particular day, the system does not need to be reliable every time. so, 80% reliability is enough.

Efficiency: Based upon the density of given values or input to calculate.

Availability: It is available in super-specialty hospitals in foreign.

Maintainability: The system should be optimized for supportability, or ease of maintenance as far as possible.

UNIT- III

DESIGN PHASE

3.1 DESIGN CONCEPTS:

The set of fundamental software design concepts are as follows:

1. Abstraction:
The lower level of abstraction provides a more detailed description of the solution. A sequence of instructions that contains a specific and limited function refers to a procedural abstraction. A collection of data that describes a data object is data abstraction.

2. Architecture:
The complete structure of the software is known as software architecture. The structure provides conceptual integrity for a system in a number of ways. The architecture is the structure of program modules where they interact with each other in a specialized way. The aim of the software design is to obtain an architectural framework of a system.

3. Patterns:
A design pattern describes a design structure and that structure solves a particular design problem in a specified content.

4. Modularity:
Modularity is the single attribute of software that permits a program to be managed easily.

5. Information hiding:
Modules must be specified and designed so that the information like algorithm and data presented in a module is not accessible for other modules not requiring that information.
6. Functional independence Functional independence is the concept of separation and is related to the concept of modularity, abstraction, and information hiding. The functional independence is accessed using two criteria i.e. Cohesion and coupling. Cohesion is an extension of the information hiding concept. A cohesive module performs a single task and it requires a small

interaction with the other components in other parts of the program. Coupling is an indication of interconnection between modules in a structure of software.

7. Refinement:
Refinement is a top-down design approach. It is a process of elaboration. A program is established for refining levels of procedural details.

8. Refactoring:
Refactoring is the process of changing the software system in a way that it does not change the external behaviour of the code and still improves its internal structure.

9. Design classes:
The model of software is defined as a set of design classes. Every class describes the elements of the problem domain and that focus on features of the problem which are user-visible.

3.2DesignConstraints:

Design Constraints are generally the limitations on a design. They include imposed limitations that you don't control and limitations that are self-imposed as a way to improve a design. The following are common types of design constraints. 9 Types of Design Constraints:

Commercial Constraints:
Basic commercial constraints such as time and budget come under commercial constraints

Requirements:
Requirements specify the basic needs of a project. Ex: Functional requirements.

Non-Functional Requirements:
Non-Functional Requirements are the requirements that specify intangible elements of a design.

Compliance:
Compliance refers to applicable laws, regulations, and standards.

Style:

A style guide or multiple style guides related to an organization, brand, product, service, environment or project. For example, a product development team may follow a style guide for a brand family that constrains the colors and layout of package designs.

Sensory Design:

Beyond visual design, constraints may apply to taste, touch, sound and smell. For example, a brand identity that calls for products to smell fruity.

Usability:

Usability principles imply frameworks and standards. Ex: The principle of least astonishment.

Principles:

Principles include the design principles of an organization, team, or individual. For example, a designer who uses form follows function to constrain designs.

Integration:

A design that needs to work with other things such as products, services, systems, processes, controls, partners, and information.

3.3 CONCEPTUAL DESIGN:

The iterative model is a software development approach that breaks down the project into smaller, manageable iterations or cycles. Each iteration involves a subset of the project's overall features and functionalities, allowing for incremental development and continuous improvement. Here's how the iterative model can be applied to the Real Estate Property Value Estimation ML Project:

Iterative Development: The development of the Real Estate Property Value Estimation ML Project follows an iterative approach, where each iteration focuses on implementing specific features or components of the system. For example, the first iteration may focus on data collection and preprocessing, while subsequent iterations may address model development, user interface design, and integration with external data sources.

Incremental Delivery: With each iteration, the project team delivers incremental value to users by adding new features or improving existing ones. This incremental delivery allows users to provide feedback early in the development process, ensuring that the final product meets their needs and expectations.

Feedback Loop: The iterative model emphasizes the importance of feedback from users and stakeholders throughout the development lifecycle. As each iteration is completed and delivered, users have the opportunity to provide feedback on the functionality, usability, and accuracy of the system. This feedback informs subsequent iterations, allowing the team to make adjustments and improvements based on user input.

Continuous Improvement: The iterative model promotes continuous improvement by allowing the project team to refine and enhance the system with each iteration. As feedback is incorporated and new features are implemented, the system evolves over time to better meet the needs of users and adapt to changing requirements and market conditions.

Risk Management: By breaking down the project into smaller iterations, the iterative model helps mitigate risks associated with complex software development projects. Risks are addressed incrementally, allowing the team to identify and mitigate issues early in the development process before they escalate into larger problems.

Use case diagram:

Use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. Actors are the external entities that interact with the system. The use cases are represented by either circles or ellipses.

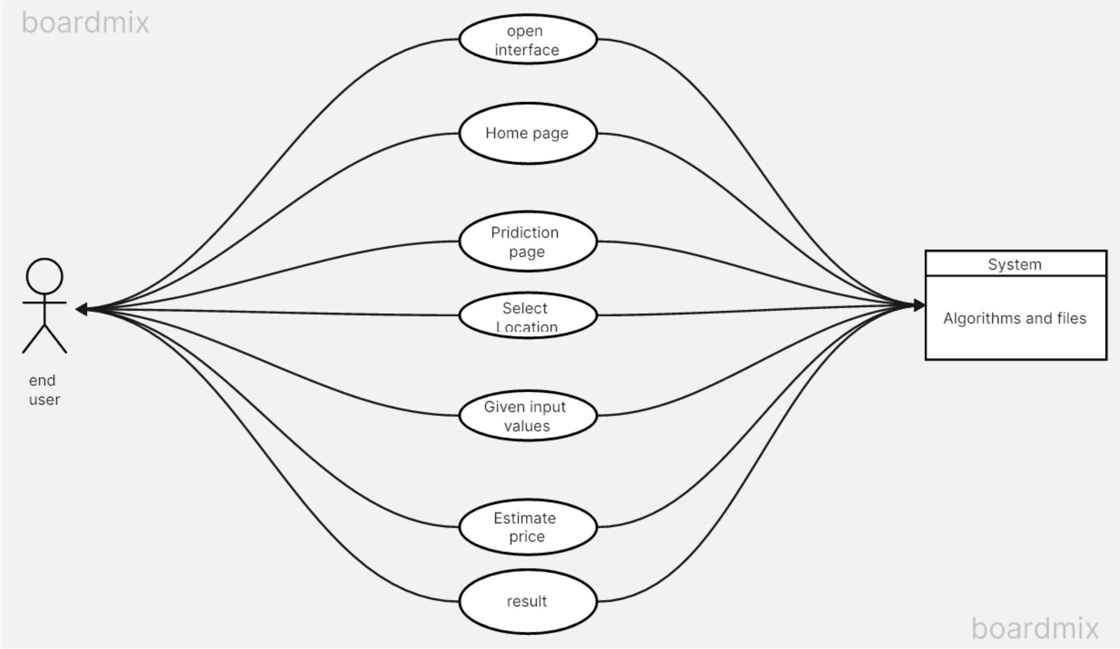


Fig. Use case diagram

3.4 LOGICAL DESIGN:

The logical design of a system pertains to an abstract representation of the data flows, inputs and outputs of the system. This is often conducted via modeling, using an over-abstract and sometimes graphical model of the actual system.

Sequence diagram:

A Sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development.

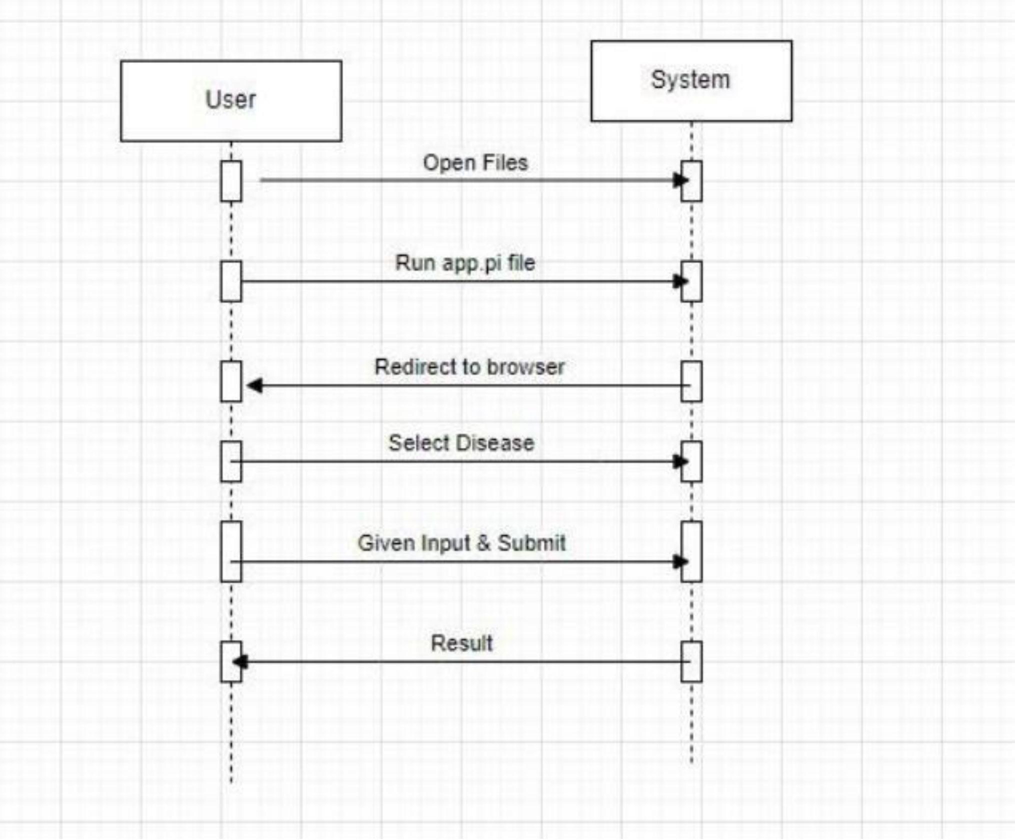


Fig. Sequence diagram

Activity diagram:

Activity diagram is essentially a fancy flowchart: Activity and state diagrams are related. Statechart diagram focuses on objects undergoing a process. An activity diagram focuses on the flow of activities involved in a single process. The activity diagram shows the activities depend on one another.

An activity represents the performance of the task or duty in a workflow. It may also represent the execution of a statement in a procedure. You can share activities between state machines. However, transitions cannot be shared.

Activity diagrams provide a way to model the workflow of a business process, code specific information such as a class operation. The transitions are implicitly triggered by the completion of the actions in the source activities.

The main difference between activity and state chart diagrams is activities are activity-centric, while state chart diagrams are state-centric.

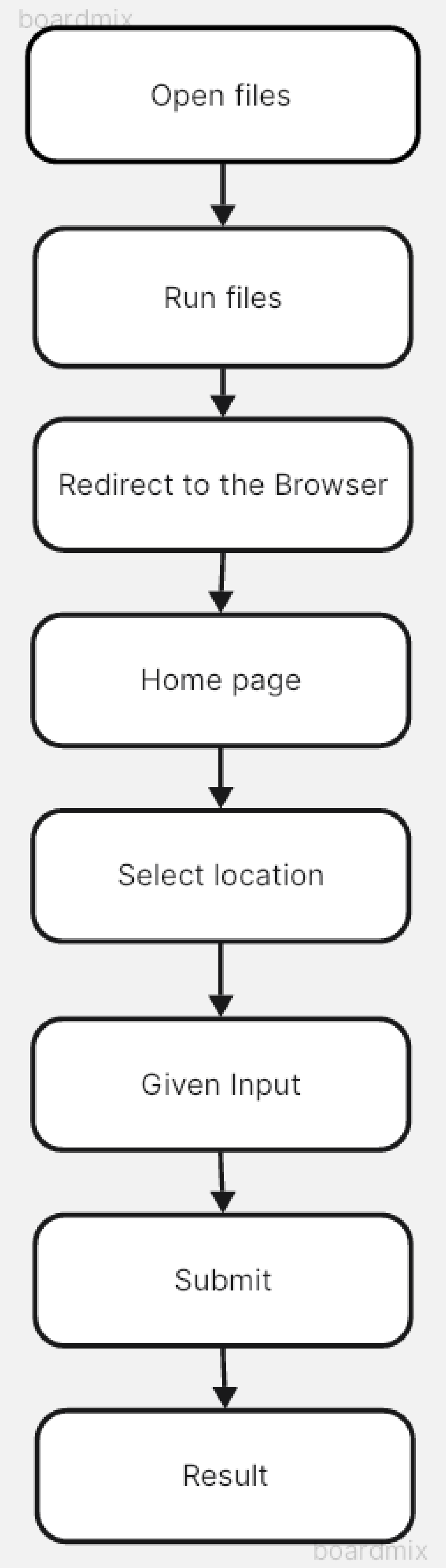
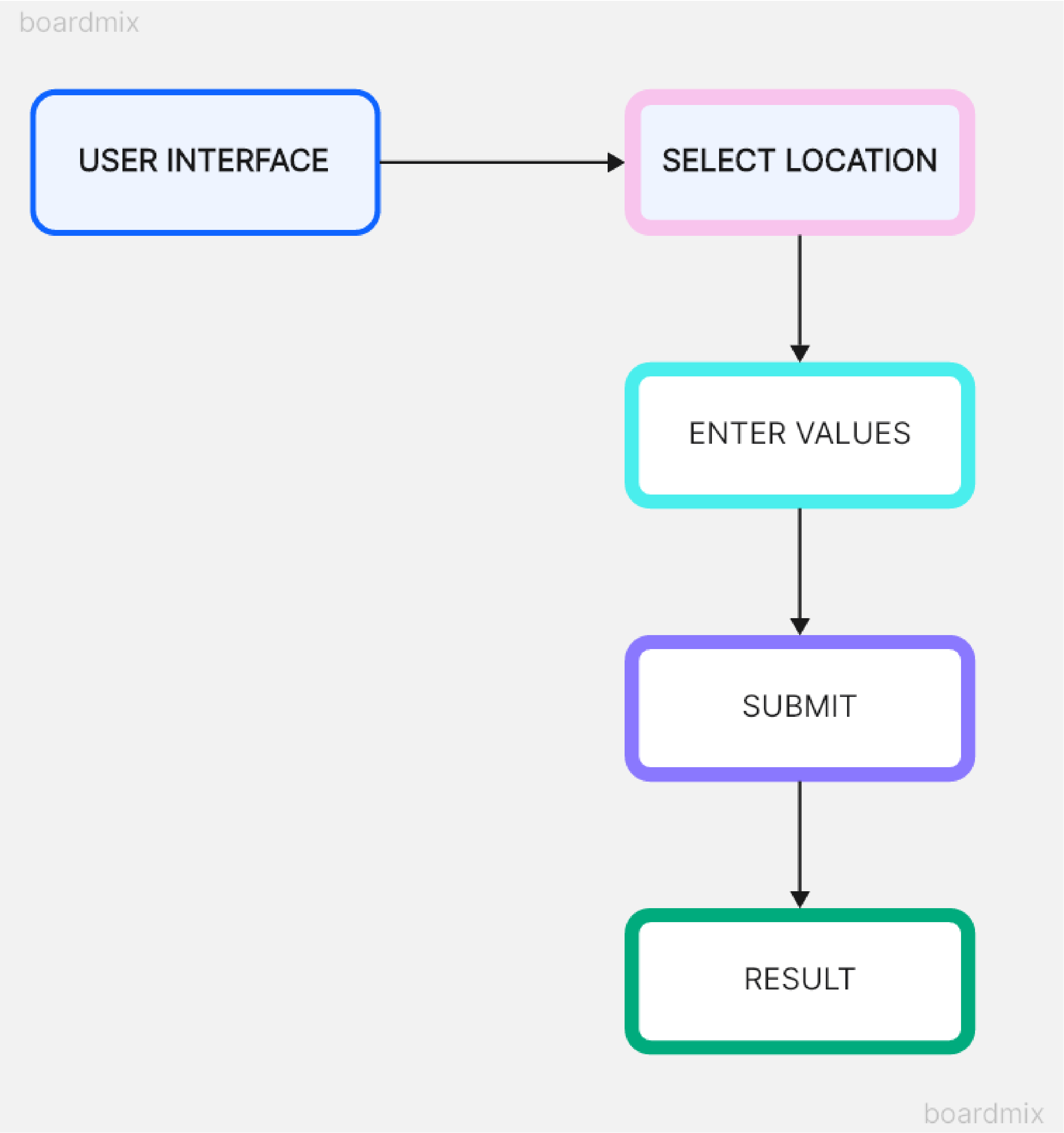


Fig. Activity diagram

3.5 ARCHITECTURAL DESIGN:



User Interface:

- The user interface component serves as the primary interaction point between users and the system. It provides a visually appealing and intuitive interface for users to input property details, customize valuation parameters, and view valuation results.

Select Locations:

The select locations feature allows users to specify the geographical location of the property for valuation. Users can input location details such as address, city, state, or zip code to identify the property's location accurately.

Enter Values:

- The enter values feature enables users to input various property details and characteristics that are relevant for valuation purposes. This includes features such as property size, number of bedrooms and bathrooms, amenities, condition, and other relevant factors.

Submit:

- The submit functionality allows users to initiate the property valuation process after entering all required information. Upon submission, the system collects the user input, preprocesses the data, and generates a valuation result based on the specified parameters.

Result:

The result component presents the property valuation result to the user in a clear and understandable format. This includes displaying the estimated property value, confidence intervals, and any additional insights or recommendations based on the valuation analysis.

3.6 ALGORITHMIC DESIGN:

Step 1: Start

Step 2: Open interface

Step 3: Upload data sets

Step 4: Pre-processing

Step 5: Training and testing

Step 6: Enter the constraints

Step 7: Submit

Step 8: Get the result

Step 9: Stop

3.7 MODULE DESIGN SPECIFICATIONS :

1. **Data Collection Module:**
   - Responsible for gathering real estate data from various sources, including property

listings, market trends, and historical sales records.

- Implements data collection mechanisms such as web scraping, API integration, or database querying to retrieve relevant data.
- Handles data preprocessing tasks, including cleaning, normalization, and feature extraction, to prepare the data for analysis.

2. **Machine Learning Model Module:**
    - Implements machine learning algorithms, such as regression models, decision trees, or neural networks, for property valuation.
    - Utilizes trained models to analyze input data and generate property valuation estimates.
    - Incorporates techniques for model evaluation, validation, and optimization to ensure accuracy and reliability of valuation results.
3. **User Interface Module:**
    - Provides a user-friendly interface for users to input property details, customize valuation parameters, and view valuation results.
    - Implements interactive input forms, dropdown menus, sliders, and other UI components for data entry and customization.
    - Utilizes data visualization tools, such as charts, graphs, and maps, to present valuation results in a visually appealing and understandable format.
4. **Integration Module:**
    - Facilitates integration with external data sources, APIs, and services for accessing additional real estate data and market information.
    - Implements data exchange protocols, authentication mechanisms, and error handling routines for seamless integration with external systems.
    - Ensures data integrity, security, and compliance with privacy regulations during data exchange processes.
5. **Reporting and Visualization Module:**
    - Generates comprehensive reports and visualizations based on valuation results, including property value estimates, confidence intervals, and trend analysis.
    - Implements interactive dashboards, charts, graphs, and tables to visualize key metrics and insights effectively.
    - Provides options for exporting reports in various formats, such as PDF, Excel, or CSV, for sharing and further analysis.
6. **Customization Module:**
    - Allows users to customize valuation parameters, such as weighting factors, model inputs, and analysis criteria, to tailor valuation results to their specific needs.
    - Implements validation checks and input validation routines to ensure that user-customized parameters are valid and consistent with system requirements.
    - Provides feedback mechanisms and guidance prompts to assist users in customizing valuation settings effectively.
7. **Feedback and Monitoring Module:**
    - Enables users to provide feedback on valuation results, user experience, and system performance.
    - Implements feedback forms, surveys, or rating systems to collect user input and

suggestions for system improvements.

- Monitors system performance, data quality, and user engagement metrics to identify areas for optimization and enhancement.

8. **Administration and Management Module:**
    - Provides administrative tools and functionalities for system configuration, user management, and access control.
    - Implements role-based access control (RBAC), authentication, and authorization mechanisms to ensure secure access to system resources.
    - Enables system administrators to configure system settings, manage user accounts, and monitor system activity for compliance and governance purposes.

UNIT-IV

CODING AND OUTPUT SCREENS

Code link: **https://github.com/durgap4444/prophecy-_estates/blob/main/miniproject/datafinal.ipynb**

4.1 . **sample code 1**:

# Import necessary libraries

```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
%matplotlib inline
import matplotlib
matplotlib.rcParams["figure.figsize"] = (20,10)
```

# Install scikit-learn version 1.4.0

```
pip install scikit-learn==1.4.0
```

# Read the dataset

```
df1 = pd.read_csv("Bengaluru_House_Data.csv")
df1.head()
```

# Data preprocessing and cleaning

```
df1['location'].value_counts()


df1['location'].replace({'Whitefield':'Guntur'},inplace = True)
```

In [9]:

```
df1['location'].replace({'8th Phase JP Nagar':'Machilipatnam','Electronic City
Phase II':'Vijayawada','Chikka Tirupathi':'Kakinada','Kanakpura
Road':'Visakhapatnam','Thanisandra':'Vijayanagaram','Yelahanka':'Rajamandri','
Uttarahalli':'Eluru','Hebbal':'Tenali','Marathahalli':'Narasaraopet','Raja
Rajeshwari Nagar':'Chilakalurupet','Bannerghatta Road':'Ongole','Hennur
Road':'Tirupati','7th Phase JP Nagar':'Srikakulam','Haralur
Road':'Bapatla','Sarjapur':'Chirala','Kasavanhalli ':'Kurnool','Ramamurthy
Nagar':'Kadapa','Koramangala':'Anathapuram','Lingadheeranahalli':'Guntakal','S
arjapur  Road':'Nellore','Electronic City':'Nagari','Akshaya
Nagar':'Chandragiri'},inplace = True)
```

In [10]:

```
df1


df1.shape


df1.columns
```

# Drop irrelevant columns

```
df2 =
df1.drop(['area_type','society','balcony','availability'],axis='columns')
```

```
df2.shape


df2['location'].value_counts()
df2.isnull().sum()
df3 = df2.dropna()
df3.isnull().sum()

df3['bhk'] = df3['size'].apply(lambda x: int(x.split(' ')[0]))
df3.bhk.unique()
# Convert total_sqft to numeric values
def is_float(x):
    try:
        float(x)
    except:
        return False
    return True
df3[~df3['total_sqft'].apply(is_float)].head(10)

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

df4 = df3.copy()
df4.total_sqft = df4.total_sqft.apply(convert_sqft_to_num)
df4 = df4[df4.total_sqft.notnull()]
df4.head(2)

df5 = df4.copy()
df5['price_per_sqft'] = df5['price']*100000/df5['total_sqft']
df5.head()

df5_stats = df5['price_per_sqft'].describe()
df5_stats

df5.to_csv("bhp.csv",index=False)

df6 = df5[~(df5.total_sqft/df5.bhk<300)]
df6.shape

df5.location = df5.location.apply(lambda x: x.strip())
location_stats = df5['location'].value_counts(ascending=False)
location_stats

location_stats.values.sum()

len(location_stats[location_stats>50])

location_stats_less_than_50 = location_stats[location_stats<=50]
location_stats_less_than_50

len(df5.location.unique())
```

```
df5.location = df5.location.apply(lambda x: 'other' if x in
location_stats_less_than_50 else x)
len(df5.location.unique())

df5.head(5)

df5[df5.total_sqft/df5.bhk<300].head()

df6.price_per_sqft.describe()

# Data outlier removal based on location
def remove_pps_outliers(df):
    df_out = pd.DataFrame()
    for key, subdf in df.groupby('location'):
        m = np.mean(subdf.price_per_sqft)
        st = np.std(subdf.price_per_sqft)
        reduced_df = subdf[(subdf.price_per_sqft>(m-st)) &
(subdf.price_per_sqft<=(m+st))]
        df_out = pd.concat([df_out,reduced_df],ignore_index=True)
    return df_out
df7 = remove_pps_outliers(df6)
df7.shape

def plot_scatter_chart(df,location):
    bhk2 = df[(df.location==location) & (df.bhk==2)]
    bhk3 = df[(df.location==location) & (df.bhk==3)]
    matplotlib.rcParams['figure.figsize'] = (15,10)
    plt.scatter(bhk2.total_sqft,bhk2.price,color='skyblue',label='2 BHK',
s=50)
    plt.scatter(bhk3.total_sqft,bhk3.price,marker='*', color='green',label='3
BHK', s=50)
    plt.xlabel("Total Square Feet Area")
    plt.ylabel("Price (Lakh Indian Rupees)")
    plt.title(location)
    plt.legend()

plot_scatter_chart(df7,"Guntur")

plot_scatter_chart(df7,"Vijayawada")

def remove_bhk_outliers(df):
    exclude_indices = np.array([])
    for location, location_df in df.groupby('location'):
        bhk_stats = {}
        for bhk, bhk_df in location_df.groupby('bhk'):
            bhk_stats[bhk] = {
                'mean': np.mean(bhk_df.price_per_sqft),
                'std': np.std(bhk_df.price_per_sqft),
                'count': bhk_df.shape[0]
            }
        for bhk, bhk_df in location_df.groupby('bhk'):
            stats = bhk_stats.get(bhk-1)
            if stats and stats['count']>5:
                exclude_indices = np.append(exclude_indices,
bhk_df[bhk_df.price_per_sqft<(stats['mean'])].index.values)
    return df.drop(exclude_indices,axis='index')
df8 = remove_bhk_outliers(df7)
# df8 = df7.copy()
```

```
df8.shape

plot_scatter_chart(df8,"Guntur")

plot_scatter_chart(df8,"Vijayawada")

import matplotlib
matplotlib.rcParams["figure.figsize"] = (20,10)
plt.hist(df8.price_per_sqft,rwidth=0.6)
plt.xlabel("Price Per Square Feet")
plt.ylabel("Count")

df8.bath.unique()

plt.hist(df8.bath,rwidth=0.6)
plt.xlabel("Number of bathrooms")
plt.ylabel("Count")

df8[df8.bath>10]


df8[df8.bath>df8.bhk+2]
df9 = df8[df8.bath<df8.bhk+2]
df9.shape
```

Out[49]:

```
df9.head(2)

df10 = df9.drop(['size','price_per_sqft'],axis='columns')
df10.head(3)

dummies = pd.get_dummies(df10.location)
dummies.head(3)

df11 = pd.concat([df10,dummies.drop('other',axis='columns')],axis='columns')
df11.head()

df12 = df11.drop('location',axis='columns')
df12.head(2)

df12.shape

X = df12.drop(['price'],axis='columns')
X.head(3)

X.shape

y = df12.price
y.head(3)

len(y)
# Train-test split
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test =
train_test_split(X,y,test_size=0.2,random_state=10)

# Model training
from sklearn.linear_model import LinearRegression
lr_clf = LinearRegression()
```

```
lr_clf.fit(X_train,y_train)
lr_clf.score(X_test,y_test)

# Model evaluation
from sklearn.model_selection import ShuffleSplit
from sklearn.model_selection import cross_val_score

cv = ShuffleSplit(n_splits=5, test_size=0.2, random_state=0)

cross_val_score(LinearRegression(), X, y, cv=cv)

# Model selection using GridSearchCV
import pandas as pd
from sklearn.model_selection import GridSearchCV, ShuffleSplit
from sklearn.linear_model import LinearRegression, Lasso
from sklearn.tree import DecisionTreeRegressor

def find_best_model_using_gridsearchcv(X1, y1):
    algos = {
        'linear_regression': {
            'model': LinearRegression(),
            'params': {}
        },
        'lasso': {
            'model': Lasso(),
            'params': {
                'alpha': [1, 2],
                'selection': ['random', 'cyclic']
            }
        },
        'decision_tree': {
            'model': DecisionTreeRegressor(),
            'params': {
                'criterion': ['friedman_mse'],  # Update criterion parameter
                'splitter': ['best', 'random']
            }
        }
    }
    scores = []
    cv = ShuffleSplit(n_splits=5, test_size=0.2, random_state=0)
    for algo_name, config in algos.items():
        gs = GridSearchCV(config['model'], config['params'], cv=cv,
return_train_score=False)
        gs.fit(X1, y1)
        scores.append({
            'model': algo_name,
            'best_score': gs.best_score_,
            'best_params': gs.best_params_
        })
return pd.DataFrame(scores, columns=['model', 'best_score', 'best_params'])

# Example usage with X1 and y1 (your feature matrix and target variable)
best_model_results = find_best_model_using_gridsearchcv(X, y)
print(best_model_results)
```

```
# Model prediction
def predict_price(location, sqft, bath, bhk):
    loc_index = np.where(X.columns == location)[0][0]

    x = np.zeros(len(X.columns))
    x[0] = sqft
    x[1] = bath
    x[2] = bhk
    if loc_index >= 0:
        x[loc_index] = 1

    predicted_price = lr_clf.predict([x])[0]
    formatted_price = "{:.2f} lakhs".format(predicted_price)
    return formatted_price

predict_price('Guntur',1001, 2, 3)

predict_price('Vijayawada',1001, 2, 3)

# Save trained model and column information for deployment
import pickle
with open('Profecy_estates.pickle','wb') as f:
    pickle.dump(lr_clf,f)

import json
columns = {
    'data_columns' : [col.lower() for col in X.columns]
}
with open("columns.json","w") as f:
    f.write(json.dumps(columns))
```

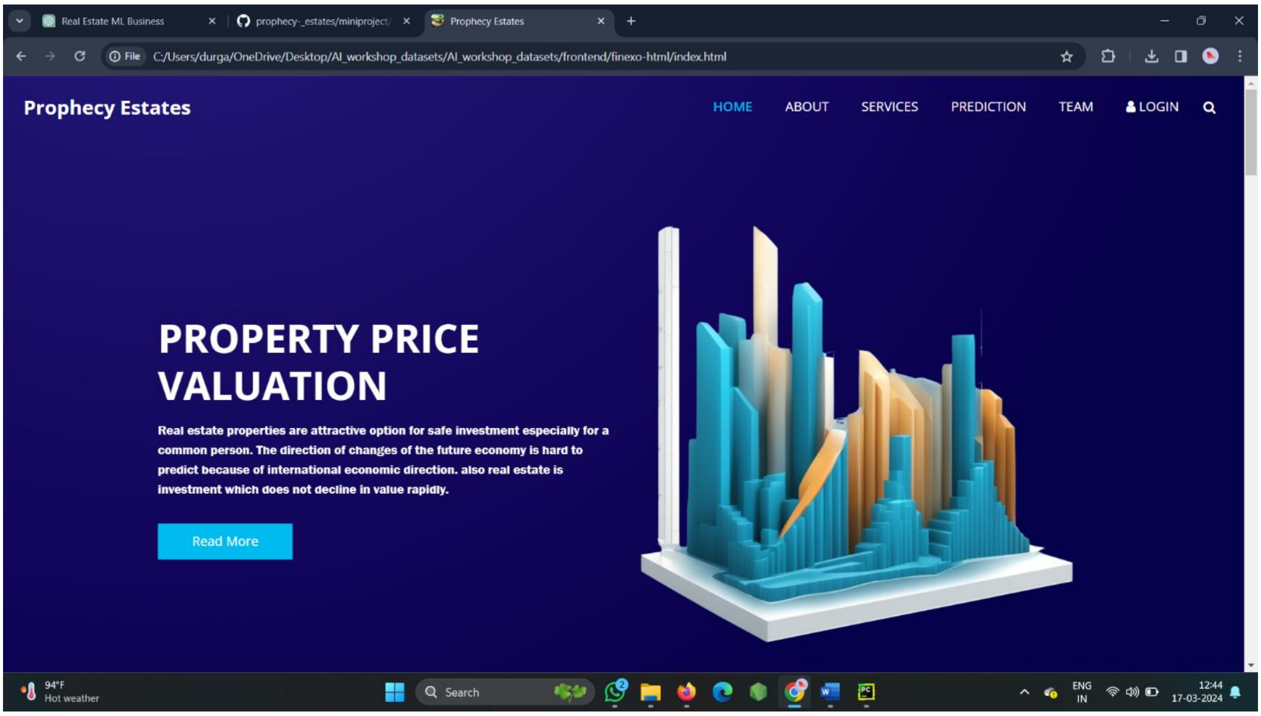
4.2 . **sample code 2**:

Code link:
https://github.com/durgap4444/prophecy_estates/blob/main/miniproject/backend/backend.py

```
from flask import Flask, request, jsonify

import util



# Initialize Flask application

application = Flask(__name__)



# Endpoint to retrieve location names

@application.route('/get_location_names', methods=['GET'])
```

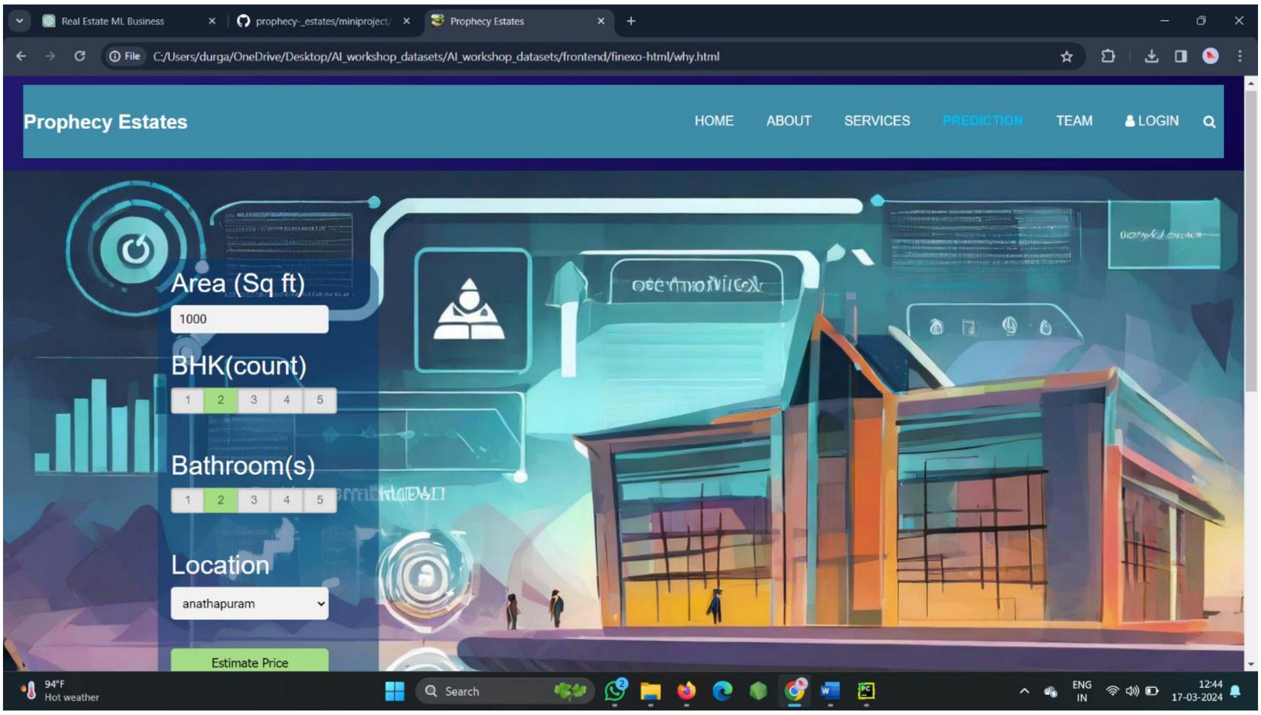
```
def get_location_names():
    # Get location names from utility function
    response = jsonify({
        'locations': util.get_location_names()
    })
    # Add CORS header to allow cross-origin requests
    response.headers.add('Access-Control-Allow-Origin', '*')

    return response

# Endpoint to predict home price
@application.route('/predict_home_price', methods=['GET', 'POST'])
def predict_home_price():
    # Extract input parameters from request
    total_sqft = float(request.form['total_sqft'])
    location = request.form['location']
    bhk = int(request.form['bhk'])
    bath = int(request.form['bath'])

    # Get estimated price from utility function
    response = jsonify({
        'estimated_price': util.get_estimated_price(location, total_sqft, bhk, bath)
    })
    # Add CORS header to allow cross-origin requests
    response.headers.add('Access-Control-Allow-Origin', '*')

    return response

# Start Flask server
if __name__ == "__main__":
    print("Starting Python Flask Server For Home Price Prediction...")
    # Load saved model artifacts
    util.load_saved_artifacts()
    application.run()
```

4.3 OUTPUT SCREENS & SCREEN REPORTS:



PROPERTY PRICE VALUATION

Real estate properties are attractive option for safe investment especially for a common person. The direction of changes of the future economy is hard to predict because of international economic direction. also real estate is investment which does not decline in value rapidly.

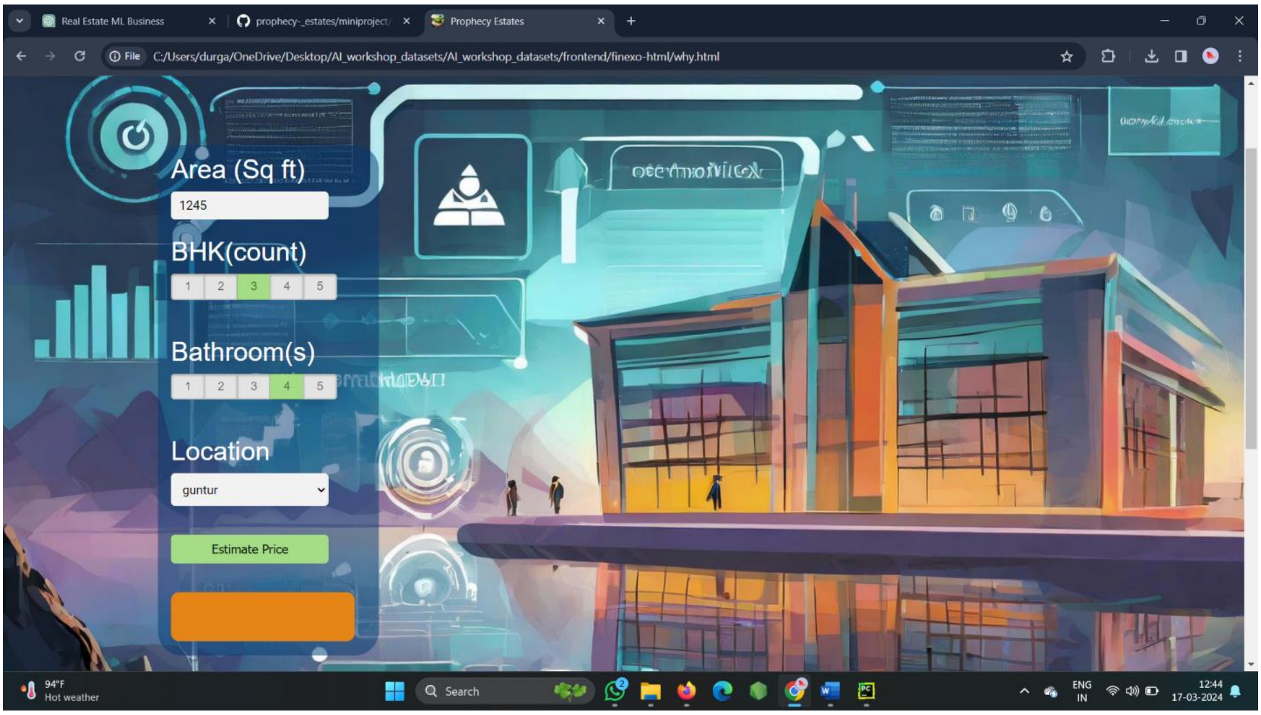


Area (Sq ft): 1000

BHK(count): 2

Bathroom(s): 2

Location: anathapuram


Real Estate ML Business
prophecy-_estates/miniproject/
Prophecy Estates
File
C:/Users/durga/OneDrive/Desktop/AI_workshop_datasets/AI_workshop_datasets/frontend/finexo-html/why.html
94°F
Hot weather
Search
ENG
IN
12:44
17-03-2024

Area (Sq ft)

1245

BHK(count)

1 2 [✓ 3] 4 5

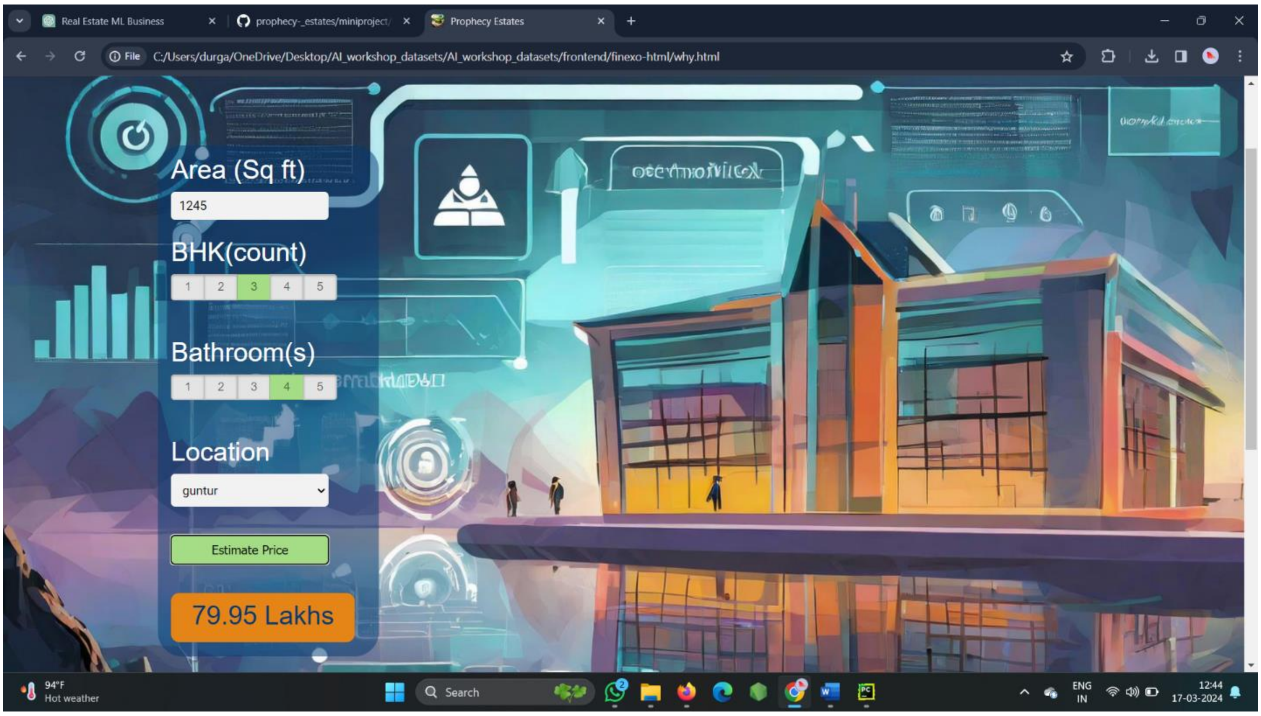
Bathroom(s)

1 2 3 [✓ 4] 5

Location

guntur

Estimate Price


Real Estate ML Business
prophecy-_estates/miniproject/
Prophecy Estates
File
C:/Users/durga/OneDrive/Desktop/AI_workshop_datasets/AI_workshop_datasets/frontend/finexo-html/why.html
94°F
Hot weather
Search
ENG
IN
12:44
17-03-2024

Area (Sq ft)

1245

BHK(count)

1 2 [✓ 3] 4 5

Bathroom(s)

1 2 3 [✓ 4] 5

Location

guntur

Estimate Price

79.95 Lakhs

UNIT- V
TESTING

5.1 INTRODUCTION TO TESTING:

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies, and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

5.2 TYPES OF TESTING:

Unit Testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at the component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration Testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event-driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfied, as shown by successful unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components

Functional Test:

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals. Functional testing is centered on the following items:
Valid Input: identified classes of valid input must be accepted.
Invalid Input: identified classes of invalid input must be rejected.
Functions: identified functions must be exercised.
Output: identified classes of application outputs must be exercised. Systems/Procedures: interfacing systems or procedures must be invoked. The organization and preparation of functional tests are focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identifying Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined

System Test:

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing:

White Box Testing is a testing in which the software tester has knowledge of the inner workings, structure, and language of the software, or at least its purpose. It is purposeful. It is used to test areas that cannot be reached from a black box level.

Black Box Testing:

Black Box Testing is testing the software without any knowledge of the inner workings, structure, or language of the module being tested. Black box tests, like most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated as a black box. You cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

5.3 TEST CASES AND TEST REPORTS:

Unit testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Integration testing:

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results:

All the test cases mentioned above passed successfully. No defects were encountered.

Acceptance testing:

User Acceptance Testing is a critical phase of any project and requires significant participation by the end-user. It also ensures that the system meets the functional requirement.

Test Results:

All the test cases mentioned above passed successfully. No defects were encountered.

UNIT- VI

IMPLEMENTATION

6.1 IMPLEMENTATION PROCESS :

Linear Regression Algorithm for Machine Learning

Linear regression is a type of supervised machine learning algorithm that computes the linear relationship between a dependent variable and one or more independent features. When the number of the independent feature, is 1 then it is known as Univariate Linear regression, and in the case of more than one feature, it is known as multivariate linear regression.

The interpretability of linear regression is a notable strength. The model's equation provides clear coefficients that elucidate the impact of each independent variable on the dependent variable, facilitating a deeper understanding of the underlying dynamics. Its simplicity is a virtue, as linear regression is transparent, easy to implement, and serves as a foundational concept for more complex algorithms.

Linear regression is not merely a predictive tool; it forms the basis for various advanced models. Techniques like regularization and support vector machines draw inspiration from linear regression, expanding its utility. Additionally, linear regression is a cornerstone in assumption testing, enabling researchers to validate key assumptions about the data.

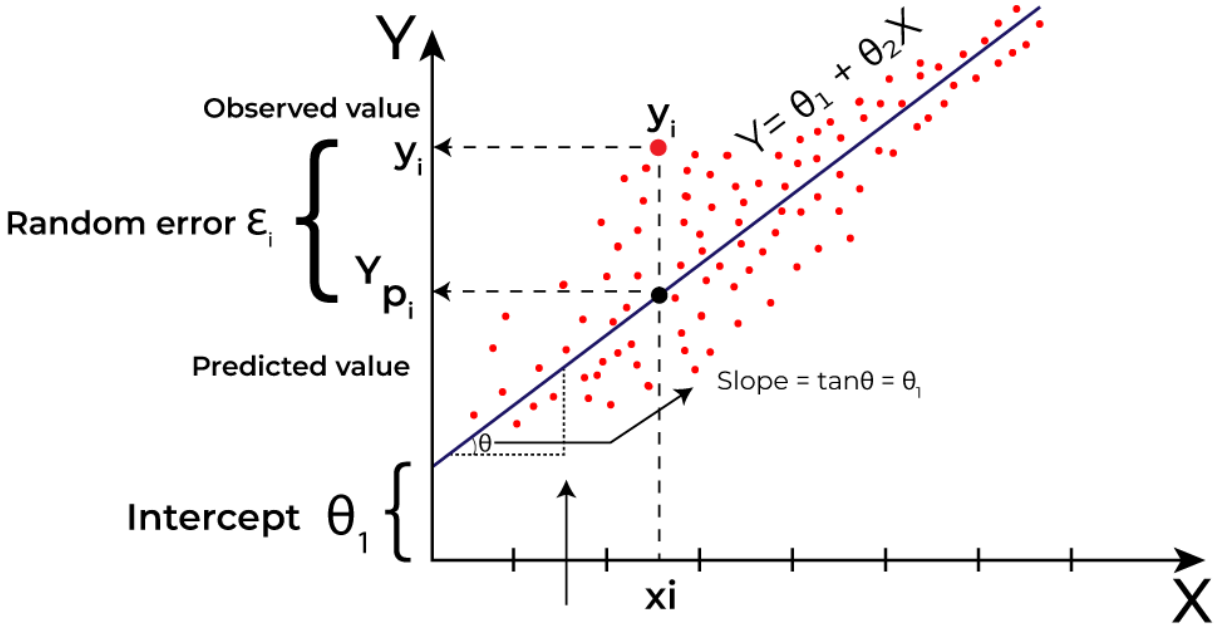

Y
Observed value
y_i
y_i
Random error ε_i
Y_{p_i}
Predicted value
$Y = \theta_1 + \theta_2 X$
Slope = tanθ = θ_1
θ
Intercept θ_1
xi
X

Fig.Linear Regression

Types of Linear Regression :

There are two main types of linear regression:

Simple Linear Regression

This is the simplest form of linear regression, and it involves only one independent variable and one dependent variable.

Multiple Linear Regression

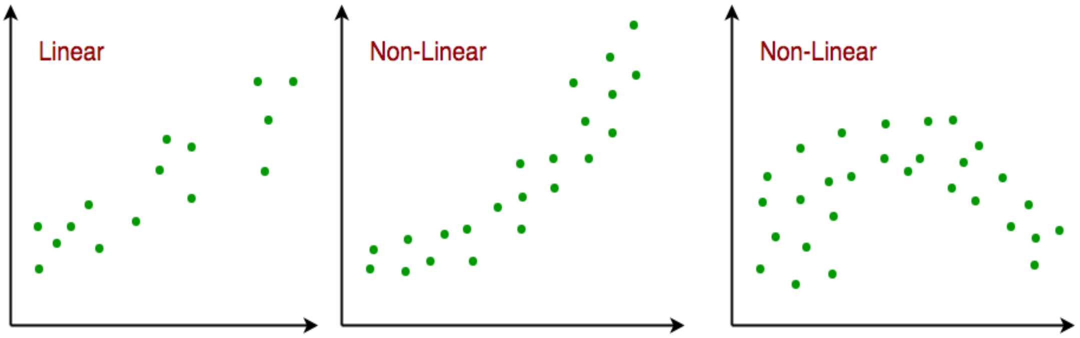
This involves more than one independent variable and one dependent variable.

The best Fit Line equation provides a straight line that represents the relationship between the dependent and independent variables. The slope of the line indicates how much the dependent variable changes for a unit change in the independent variable(s).

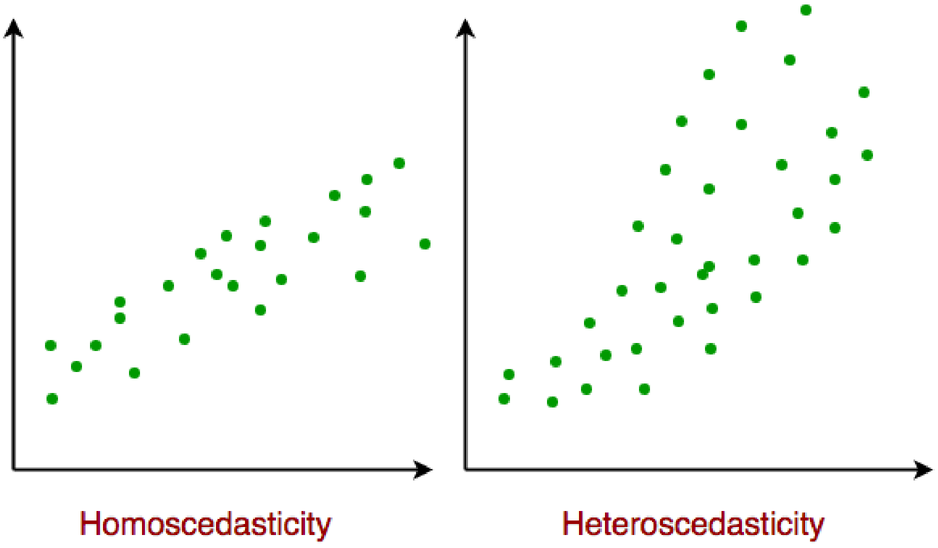
Assumptions of Simple Linear Regression

Linear regression is a powerful tool for understanding and predicting the behavior of a variable, however, it needs to meet a few conditions in order to be accurate and dependable solutions.

1. **Linearity**: The independent and dependent variables have a linear relationship with one another. This implies that changes in the dependent variable follow those in the independent variable(s) in a linear fashion. This means that there should be a straight line that can be drawn through the data points. If the relationship is not linear, then linear regression will not be an accurate model.



2. **Independence**: The observations in the dataset are independent of each other. This means that the value of the dependent variable for one observation does not depend on the value of the dependent variable for another observation. If the observations are not independent, then linear regression will not be an accurate model.
3. **Homoscedasticity**: Across all levels of the independent variable(s), the variance of the errors is constant. This indicates that the amount of the independent variable(s) has no impact on the variance of the errors. If the variance of the residuals is not constant, then linear regression will not be an accurate model.



Homoscedasticity in Linear Regression

4. **Normality**: The residuals should be normally distributed. This means that the residuals should follow a bell-shaped curve. If the residuals are not normally distributed, then linear regression will not be an accurate model.

6.4 USER MANUAL:

The user has to open the folder of the project. Then the user has to open the file i.e., ” prophecyestates.ip”.Then the link will be generated automatically the link will be redirected to the browser and the Homepage will be displayed. In the home page, the diseases list will be displayed. The user has to select the disease and enter the attributes then after giving attributes as an input. The system will preprocess and display the output.

UNIT- VII

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 CONCLUSION:

In conclusion, the Real Estate Property Value Estimation Project represents a significant advancement in leveraging machine learning algorithms to provide accurate and reliable property valuations. Through meticulous data collection, preprocessing, and model development, we have created a robust platform capable of estimating property values based on key factors such as location, size, and amenities.

Our platform offers users a user-friendly interface where they can easily input property details and receive instant estimations of property values. By harnessing the power of linear regression and other machine learning techniques, we have built a system that empowers users with valuable insights for informed decision-making in real estate transactions.

Moving forward, we remain committed to enhancing our platform, incorporating user feedback, and staying abreast of the latest advancements in machine learning technology. We are dedicated to providing users with the most accurate and reliable property valuations possible, thereby revolutionizing the real estate industry and empowering individuals and businesses alike.

Thank you for joining us on this journey to transform property valuation through technology.

.

7.2 FUTURE ENHANCEMENTS:

1. **Integration of Advanced Algorithms:** Explore the integration of more advanced machine learning algorithms such as decision trees, random forests, or gradient boosting machines. These algorithms may offer improved accuracy and robustness in predicting property values, especially in complex scenarios.
2. **Feature Engineering:** Continuously refine and expand the set of features used for property valuation. This may involve incorporating additional data sources such as neighborhood amenities, proximity to schools or transportation, crime rates, or economic indicators. Enhanced feature engineering can lead to more accurate estimations and better insights for users.
3. **Geospatial Analysis:** Implement geospatial analysis techniques to capture spatial relationships and patterns that may influence property values. This could involve utilizing geographic information systems (GIS) data to analyze factors such as neighborhood

boundaries, land use, and spatial autocorrelation.

4. **Dynamic Pricing Models:** Develop dynamic pricing models that take into account temporal variations and market trends. By analyzing historical property data and market dynamics, the platform can adapt to changing market conditions and provide up-to-date estimations that reflect current trends and demand.
5. **User Personalization:** Implement features for user personalization and customization based on individual preferences and requirements. This could include the ability for users to save favorite properties, set custom search criteria, receive personalized recommendations, and track property value trends in specific areas of interest.
6. **Interactive Data Visualization:** Enhance the platform with interactive data visualization tools that enable users to explore property data in a more intuitive and engaging manner. This could include interactive maps, charts, and graphs that provide insights into property trends, price distributions, and comparative analysis.

UNIT- VIII

BIBILOGRAPHY

8.1 BOOKS REFERRED:

- "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow" by Aurélien Géron - This book provides a comprehensive introduction to machine learning techniques, including linear regression, decision trees, and ensemble methods, with practical examples using popular Python libraries.
- "Python for Data Analysis" by Wes McKinney - This book is a valuable resource for data preprocessing and analysis using Python and pandas. It covers topics such as data manipulation, cleaning, and visualization, which are essential for preparing datasets for machine learning models.
- "Introduction to Linear Regression Analysis" by Douglas C. Montgomery, Elizabeth A. Peck, and G. Geoffrey Vining - This classic textbook offers a thorough introduction to linear regression analysis, covering fundamental concepts, model assumptions, diagnostics, and interpretation of results.
- "Real Estate Finance & Investments" by William B. Brueggeman and Jeffrey D. Fisher - This book provides insights into real estate finance principles, investment analysis, and valuation techniques. It offers a solid foundation for understanding the economic factors that influence property values.

8.2 WEBSITES VISITED :

- Kaggle (https://www.kaggle.com/) - Kaggle is a popular platform for data science competitions, datasets, and kernels. It hosts various datasets related to real estate, as well as machine learning competitions focused on predictive modeling tasks.
- Zillow (https://www.zillow.com/) - Zillow is a leading online real estate marketplace that provides property information, home values, and rental listings. It can be a valuable resource for accessing real-world data for real estate valuation projects.
- Towards Data Science (https://towardsdatascience.com/) - Towards Data Science is a Medium publication that covers a wide range of topics in data science, machine learning, and artificial intelligence. It features articles, tutorials, and case studies relevant to real estate analytics and predictive modeling.
- GitHub (https://github.com/) - GitHub hosts numerous repositories containing code, datasets, and projects related to real estate valuation and machine learning. It's a great source for accessing open-source tools, libraries, and resources for building predictive models.
- Stack Overflow (https://stackoverflow.com/) - Stack Overflow is a community-driven question and answer platform for programming and software development. It can be helpful for troubleshooting technical issues, seeking advice, and sharing knowledge related to machine learning and data analysis.