Sri Lanka Institute of Information Technology

Fundamentals of Data Mining - IT3051

Mini Project – Statement of Work

2024

Group Details

Group No: 41

Group Name: The Analytica

No	Student Number	Name
1.	IT 22 3197 60	Vinushan V
2.	IT 22 3235 52	Anupama S K D N
3.	IT 22 8888 84	Mendis HKPD
4.	IT 22 3448 92	Maxwell L Y
5.	IT 22 2828 04	Gunaratne N V D P

Table of Content

Introduction

Project Background

The rapid urbanization and development of Bengaluru, often dubbed the "**Silicon Valley of India**," have led to an ever-growing demand for housing. As one of the fastest-growing cities in the country, Bengaluru faces significant challenges in its real estate market, such as fluctuating property prices, a wide disparity in pricing between different localities, and the lack of transparency for potential buyers and investors. These factors make it increasingly difficult for individuals and businesses to make informed decisions when purchasing or investing in property.

The problem of inconsistent and unpredictable housing prices is further compounded by the complexity of factors influencing property values, including proximity to tech hubs, availability of amenities, infrastructure development, and economic shifts. Without a data-driven approach, buyers and investors often rely on subjective opinions, leading to inefficient market decisions and sometimes overpaying or underestimating property values.

To address this problem, a housing price prediction application can offer an efficient and accurate solution by leveraging historical data, current market trends, and relevant factors affecting real estate pricing. By implementing such an application, users can receive precise price estimates based on location, property type, and other influential parameters. This will not only assist buyers and investors in making more informed decisions but also enhance transparency in the real estate market, contributing to a more stable and predictable housing ecosystem in Bengaluru.

This application aims to **reduce the unpredictability of the real estate market by offering a tool that can accurately predict housing prices**, thus solving the challenge of price inconsistency and helping users navigate the dynamic landscape of Bengaluru's housing market.

Objectives

The primary objective of this project is to develop a housing price prediction application for Bengaluru that accurately estimates property prices based on various features present in the dataset. By analyzing factors such as area type, availability, location, size, total square footage, number of bathrooms, balconies, and other relevant attributes, the application aims to provide precise price estimates to potential buyers and investors.

To achieve this, we will:

- Build a predictive model using Python and scikit-learn's linear regression on the Bengaluru home prices dataset from Kaggle. The model will learn the relationships between the input features and the housing prices.
- Perform data cleaning and preprocessing using Numpy and Pandas to handle missing values, inconsistencies, and outliers. Feature engineering and dimensionality reduction techniques will be applied to improve the model's performance.
- Evaluate and fine-tune the model using methods like grid search with cross-validation (GridSearchCV) and k-fold cross-validation to optimize hyperparameters and enhance accuracy.
- Develop a user-friendly web application where users can input property details such as square footage and number of bedrooms to receive predicted prices.

By completing these tasks, the project aims to assist users in making informed decisions in the dynamic Bengaluru real estate market, reducing price unpredictability and enhancing market transparency.

Scope of Work

The project focuses on developing a housing price prediction application for Bengaluru to assist potential buyers and investors in making informed decisions. The scope includes the following key tasks and deliverables:

Inclusions

1. **Data Collection and Preparation**
    - Acquire the Bengaluru home prices dataset from Kaggle.
    - Load the dataset using Pandas for initial data exploration.
    - Perform data cleaning to handle missing values and inconsistencies.
    - Detect and remove outliers to improve data quality.
    - Conduct feature engineering to create relevant variables.
    - Apply dimensionality reduction techniques to simplify the dataset.
2. **Model Building**
    - Utilize scikit-learn to develop a linear regression model.
    - Train the model using the cleaned dataset.
    - Implement GridSearchCV for hyperparameter tuning.
    - Apply k-fold cross-validation to assess model performance.
    - Use Matplotlib for data visualization to understand data patterns.
3. **Web Application Development**
    - Create a user-friendly web application using **Streamlit.**
    - Design input forms for users to enter property details like square footage and number of bedrooms.
4. **Testing and Validation**
    - Test the predictive model for accuracy and reliability.

- Validate the web application's functionality and user experience.
- Perform cross-validation to ensure the model generalizes well to unseen data.

5. **Documentation**
    - Document all steps of data cleaning, model building, and application development.
    - Prepare a comprehensive report detailing methodologies and findings.
    - Include code comments and explanations for clarity.

Exclusions

- The project will not cover deployment on cloud platforms or handling live data updates.
- Advanced front-end frameworks or libraries beyond Streamlit python library excluded.

Deliverables

- Cleaned and preprocessed dataset ready for modeling.
- Trained linear regression model saved for deployment.
- Python scripts for data processing, model training, and evaluation.
- Source code for **Streamlit** and other necessary python files.
- Final project report documenting the entire process and results.
- Presentation slides summarizing the project's objectives, methodologies, and outcomes.

Tasks and Deliverables

Detailed Task Breakdown

Tasks

1. **Data Collection and Preparation**
   - **Collect Dataset**: Obtain the Bengaluru housing prices dataset from Kaggle.
   - **Load Data**: Use Pandas to load the dataset into a DataFrame for analysis.
   - **Data Cleaning**: Handle missing values and fix any inconsistencies in the data.
   - **Outlier Removal**: Identify and remove outliers to improve data quality.
   - **Feature Engineering**: Create new features that might help the model perform better.
   - **Dimensionality Reduction**: Apply techniques to reduce the number of features while retaining important information.
2. **Model Building**
   - **Develop Model**: Use scikit-learn to build a linear regression model.
   - **Train Model**: Train the model using the cleaned dataset.
   - **Hyperparameter Tuning**: Use GridSearchCV to find the best hyperparameters.
   - **Cross-Validation**: Apply k-fold cross-validation to assess the model's performance.
   - **Data Visualization**: Use Matplotlib to visualize data distributions and model predictions.
3. **Web Application Development**
   - **Design Website**: Use **Streamlit** python library for whole development

4. **Testing and Validation**
    - **Model Testing**: Evaluate the model's accuracy and make adjustments if necessary.
    - **Web App Testing**: Test the website for usability and fix any bugs or issues.
    - **Ensure Reliability**: Perform additional validations to make sure the model works well with new data.
5. **Documentation and Reporting**
    - **Document Processes**: Keep detailed notes on data cleaning, model building, and any challenges faced.
    - **Prepare Report**: Write a comprehensive report explaining the methodologies and results.
    - **Code Comments**: Add comments in the code to explain how different parts work.

Deliverables

- **Cleaned Dataset**: The final dataset after cleaning and preprocessing.
- **Trained Model**: The saved linear regression model ready for deployment.
- **Python Scripts**: All scripts used for data processing, model training, and evaluation.
- **Website Files**: **Streamlit** and **Pickle** files for the web application.
- **Project Report**: A detailed report documenting the project's objectives, methods, and findings.
- **Presentation Slides**: Slides summarizing the project for presentation purposes.

Project Schedule

The project is planned to be completed over a **7 - period**. Below is the timeline outlining the start and end dates for major tasks, along with key milestones where significant progress will be reviewed.

Week 1: Project Initiation and Planning

- **Tasks:**
    - Define project objectives and scope.
    - Set up the development environment and install necessary software and libraries.
    - Obtain the Bengaluru housing prices dataset from Kaggle.
- **Milestone:**
    - Completion of project plan and initial setup.

Week 2: Data Exploration and Cleaning

- **Tasks:**
    - Load the dataset using Pandas and perform initial data exploration.
    - Identify and handle missing values and inconsistencies.
    - Remove outliers to improve data quality.
- **Milestone:**
    - Cleaned and preprocessed dataset ready for analysis.

Week 3: Feature Engineering and Selection

- **Tasks:**
    - Perform feature engineering to create new relevant features.
    - Apply dimensionality reduction techniques to simplify the dataset.
    - Select the most significant features for model building.
- **Milestone:**
    - Finalized dataset with selected features for modeling.

Week 4: Model Development

- **Tasks:**

    - Develop a linear regression model using scikit-learn.
    - Train the model using the prepared dataset.
    - Visualize data distributions and relationships using Matplotlib.
- **Milestone:**
    - Initial version of the predictive model completed.

Week 5: Model Evaluation and Tuning

- **Tasks:**
    - Evaluate model performance using metrics like R-squared and MAE.
    - Implement k-fold cross-validation to assess generalization.
    - Use GridSearchCV for hyperparameter tuning to improve model accuracy.
- **Milestone:**
    - Optimized and validated predictive model.

Week 6: Web Application Development

- **Tasks:**
    - Design the web interface using **Streamlit library**.
    - Implement user input forms for property details.
    - Conduct usability testing and gather feedback.
- **Milestone:**
    - Fully integrated web application.

Week 7: Testing, Documentation, and Finalization

- **Tasks:**
    - Perform end-to-end testing of the entire application.
    - Fix any bugs or issues identified during testing.
    - Prepare the final project report documenting all aspects of the project.
    - Create presentation slides summarizing the project.
- **Milestone:**
    - Project deliverables completed and ready for submission.

- Final presentation prepared.

Responsibilities

To ensure the successful execution of the project, the following roles and responsibilities are assigned to the team members:

No	Student Number	Name	Responsibilities
1.	IT 22 3197 60	Vinushan V	Data Collection and Preparation Model Building Testing and Validation
2.	IT 22 3235 52	Anupama S K D N	Data Collection and Preparation Model Building Testing and Validation
3.	IT 22 8888 84	Mendis HKPD	Data Collection and Preparation Model Building Testing and Validation
4.	IT 22 3448 92	Maxwell L Y	Model Building Web Application Development Documentation
5.	IT 22 2828 04	Gunaratne N V D P	Model Building Web Application Development Documentation

Resources and Requirements

To successfully complete this housing price prediction project for Bengaluru, the following resources and requirements are essential:

Software and Tools

- **Programming Language**: Python 3.12.4
    - The primary language for all coding tasks, including data processing, model building, and server development.
- **Data Processing Libraries**:
    - **Numpy**: For numerical computations and handling arrays.
    - **Pandas**: For data loading, cleaning, and manipulation.
- **Data Visualization**:
    - **Matplotlib**: To create plots and graphs for data exploration and result presentation.
- **Machine Learning Library**:
    - **Scikit-learn (sklearn)**: For building the linear regression model, hyperparameter tuning, and cross-validation.
- **Web Development**:
    - **Streamlit:** For building responsive web application.
- **Integrated Development Environments (IDEs)**:
    - **Jupyter Notebook**: For exploratory data analysis and initial model development.
    - **Visual Studio Code**: For writing and debugging code for the web application.

Hardware Requirements

- **Personal Computer/Laptop**:

  - **Processor**: Minimum dual-core processor to handle data processing tasks efficiently.
  - **Memory (RAM)**: At least 8 GB RAM to smoothly run data manipulation and model training processes.
  - **Storage**: Sufficient disk space to store datasets, libraries, and project files (minimum 20 GB free space recommended).
  - **Operating System**: Windows 10, macOS, or a Linux distribution compatible with the required software.

Data Access

- **Dataset Source**:
  - **Bengaluru Housing Prices Dataset from Kaggle**:
    - Access requires a Kaggle account to download the dataset.
    - Ensure compliance with Kaggle's terms of service and data usage policies.
  - **Data Storage**:
    - Securely store the dataset on the local machine.
    - Backup copies to prevent data loss.

Development Environment

- **Python Environment Management**:
  - **Anaconda Distribution**: For managing Python packages and creating isolated environments.
  - **Virtual Environments**: Use **virtualenv** or **conda** environments to manage dependencies without conflicts.
- **Version Control System**:

    - **Git**: To track changes in the codebase and collaborate if working in a team.
    - **GitHub/GitLab Account**: For hosting the repository remotely and maintaining code backups.
- **Web Browser**:
    - A modern browser like Google Chrome, Mozilla Firefox, or Microsoft Edge to test and run the web application.

Documentation and Reporting Tools

- **Microsoft Office Suite or Alternatives**:
    - **Word Processor**: Microsoft Word for writing the project report and documentation.
    - **Presentation Software**: Microsoft PowerPoint for creating presentation slides.
    - **Spreadsheet Software**: Microsoft Excel for any additional data analysis needs.

Additional Resources

- **Internet Access**:
    - Reliable internet connection for:
        - Downloading datasets and software packages.
        - Accessing online documentation and resources.
        - Researching methodologies and troubleshooting issues.
- **Reference Materials**:
    - Online tutorials, documentation, and forums related to Python, machine learning, Streamlit, and web development.
- **Communication Tools**:

    - **Email**: For communication with supervisors or advisors.
    - **Messaging Apps**: Slack, Microsoft Teams, or similar platforms if collaborating with others.

Assistance and Guidance

- **Academic Advisors or Mentors**:
    - Guidance on project direction, methodology, and best practices.
    - Feedback on deliverables and project progress.

Assumptions and Constraints

Assumptions

1. **Dataset Availability**: It is assumed that the Bengaluru housing prices dataset from Kaggle is readily accessible and contains all the necessary features required for building an accurate predictive model.
2. **Data Quality**: We assume that the dataset is of sufficient quality, with manageable levels of missing data and outliers that can be addressed through standard data cleaning procedures.
3. **Software and Tools Access**: All required software and tools, including Python, Numpy, Pandas, Matplotlib, scikit-learn, Streamlit, and IDEs like Jupyter Notebook, Visual Studio Code, and PyCharm, are assumed to be installed and functioning correctly on our development machines.
4. **Technical Proficiency**: It is assumed that the project team possesses the necessary skills in Python programming, data science methodologies, machine learning techniques, and basic web development to effectively execute the project tasks.
5. **Stable Development Environment**: We assume a stable development environment without significant technical disruptions that could impede progress.
6. **Project Timeline**: The project is assumed to be achievable within the allocated time frame set by the university for this mini project.
7. **Support and Guidance**: It is assumed that we will have access to necessary academic resources, including guidance from instructors or mentors when required.

Constraints

1. **Time Limitations**: The project's scope is constrained by the limited time available for a mini project, which may restrict the depth of data analysis and the extent of model optimization.

2. **Computational Resources**: Limited hardware capabilities may affect our ability to process large datasets or run computationally intensive algorithms, potentially impacting model training and evaluation.
3. **Dataset Limitations:** The dataset may have inherent limitations such as outdated information, missing values, or limited features, which could affect the accuracy and generalizability of the predictive model.
4. **Model Complexity**: Due to time and resource constraints, the project will focus solely on linear regression models and will not explore more complex algorithms that might improve prediction accuracy.
5. **No Deployment on Cloud Platforms**: The application will not be deployed on cloud services or made accessible over the internet, limiting its usage to local environments and reducing opportunities for external user feedback.
6. **Data Privacy and Compliance**: All activities must comply with Kaggle's terms of service and data usage policies, which may restrict certain uses of the dataset, such as sharing or redistribution.
7. **Team Availability**: Other academic responsibilities and commitments may limit the amount of time team members can dedicate to the project, potentially affecting the project's progress and deadlines.
8. **Software Dependencies**: Potential compatibility issues with software libraries and dependencies could arise, requiring additional time for troubleshooting and resolution.
9. **Evaluation Metrics**: The effectiveness of the model will be evaluated using standard metrics, but constraints may limit the ability to perform extensive validation or comparison with alternative models.

Acceptance Criteria

Success Metrics

The success of the project will be measured based on the following criteria:

1. **Model Accuracy:**
   - **R-squared Value**: Our linear regression model should achieve an R-squared value of at least 0.75 on the test dataset, indicating that the model explains 75% of the variance in housing prices.
   - **Mean Absolute Error (MAE)**: The model should have a Mean Absolute Error (MAE) less than ₹10 lakhs, ensuring that the predictions are reasonably close to the actual prices.
2. **Cross-Validation Performance**:
   - **Consistency Across Folds**: Through k-fold cross-validation, the model should demonstrate consistent performance across different subsets of the data, indicating good generalization capabilities.
   - **Low Variance**: The standard deviation of the model's performance metrics across folds should be minimal, suggesting stability.
3. **Web Application Functionality**:
   - **Accurate Predictions**: The web application must return accurate price predictions that align closely with the model's test results.
   - **User Interface**: The website should be intuitive and user-friendly, allowing users to input property details easily.
   - **Responsive Design**: The web application should function correctly on various devices and screen sizes.
4. **Documentation Quality**:

   - **Comprehensive Reporting**: The final report should thoroughly document all stages of the project, including data cleaning, model development, and application deployment.
   - **Code Clarity**: All code should be well-commented and organized, making it easy to understand and maintain.
   - **Presentation Slides**: The slides should effectively summarize the project's objectives, methodologies, results, and conclusions.

5. **Deadline Adherence**:
   - **Timely Completion**: All tasks and deliverables must be completed and submitted by the deadlines set by the university.

Approval Process

The deliverables will be reviewed and accepted through the following process:

1. **Initial Submission**:
   - We will submit all project deliverables, including the predictive model, source code, web application, and documentation, to our project supervisor by the agreed-upon deadline.
2. **Supervisor Review**:
   - The supervisor will review the submitted materials to ensure they meet the project's objectives and the university's academic standards.
   - The review will focus on the accuracy and performance of the predictive model, the functionality of the web application, and the quality of the documentation.
3. **Feedback and Revisions**:
   - If the supervisor identifies any issues or areas for improvement, we will receive feedback outlining the necessary changes.

    - We will address all feedback promptly, revising enhance the project's quality.

4. **Final Evaluation**:
    - After incorporating the feedback, we will resubmit the revised deliverables for final evaluation.
    - The supervisor will verify that all concerns have been addressed and that the project meets all acceptance criteria.
5. **Acceptance Confirmation**:
    - Once the supervisor is satisfied with the project, we will receive formal confirmation of acceptance.
    - The project will then be considered complete and ready for any required presentations or defenses.
6. **Grading and Feedback**:
    - The project will be graded based on predefined rubrics covering technical accuracy, innovation, thoroughness, and presentation.
    - We will receive final feedback highlighting the strengths of the project and areas for future improvement

Risk Management

Potential Risks

1. **Data Quality Issues**:
    - The dataset may contain missing values, inconsistencies, or errors that could negatively impact the accuracy of the predictive model.
2. **Limited Dataset Features**:
    - Important factors influencing housing prices might be missing from the dataset, leading to an incomplete analysis.
3. **Overfitting or Underfitting**:
    - The model may not generalize well to new data if it is too complex (overfitting) or too simple (underfitting).
4. **Technical Challenges**:
    - Difficulties with software installation, library dependencies, or coding errors could cause delays.
5. **Time Constraints**:
    - The project's limited timeframe might restrict the depth of data analysis and model optimization.
6. **Resource Limitations**:
    - Limited computational power may hinder the processing of large datasets or complex computations.
7. **Data Privacy and Compliance**:
    - Misuse of the dataset could lead to violations of Kaggle's terms of service or data protection regulations.
8. **Lack of Technical Expertise**:

   - Insufficient knowledge in certain areas of data science, machine learning, or web development could impede progress.

9. **User Interface Challenges**:
   - The web application may not be user-friendly or may have bugs that affect the user experience.

Mitigation Strategies

1. **Data Quality Issues**:
   - **Mitigation**: Perform thorough data cleaning and preprocessing to handle missing values and correct inconsistencies. Use data validation techniques to ensure the integrity of the dataset.
2. **Limited Dataset Features**:
   - **Mitigation**: Engage in feature engineering to create new variables from existing data that could enhance the model's predictive power. If possible, supplement the dataset with additional relevant data sources.
3. **Overfitting or Underfitting**:
   - **Mitigation**: Use cross-validation techniques like k-fold cross-validation to monitor the model's performance on unseen data. Adjust the model's complexity and apply regularization methods to achieve a good balance.
4. **Technical Challenges**:
   - **Mitigation**: Allocate time for troubleshooting and debugging. Seek assistance from instructors, peers, or online communities when facing technical issues. Keep software and libraries updated to the latest stable versions.
5. **Time Constraints**:
   - **Mitigation**: Create a detailed project timeline with clear milestones and deadlines. Prioritize critical tasks and avoid scope creep by sticking to the defined project scope.

6. **Resource Limitations**:
    - **Mitigation**: Optimize code for efficiency and manage computational resources wisely. Utilize available resources such as university computing labs or cloud services if necessary.
7. **Integration Issues**:
    - **Mitigation**: Develop and test each component (model, server, web interface) separately before integrating. Ensure consistent data formats and communication protocols between components. Follow best practices for API development.
8. **Data Privacy and Compliance**:
    - **Mitigation**: Review Kaggle's terms of service and ensure compliance with all data usage policies. Do not share the dataset publicly and store it securely. Anonymize any sensitive information if required.
9. **Lack of Technical Expertise**:
    - **Mitigation**: Allocate time for learning and skill development through online tutorials, documentation, and courses. Collaborate with classmates or seek guidance from instructors to fill knowledge gaps.
10. **User Interface Challenges**:
    - **Mitigation**: Keep the web application's design simple and focus on essential functionalities. Conduct user testing with peers to gather feedback and make iterative improvements. Ensure the interface is intuitive and accessible.