Creating a predictive model for forecasting bike-sharing demand, assisting transportation networks in resource allocation and optimizing service availability across city locations

A dissertation submitted in partial fulfillment of the requirements for the award of the Degree of

Bachelor of Technology

In

Computer Science and Engineering

By

HAMDA HUSSAIN (23U61A0530)

Under the guidance of

Mrs. T Lakshmi Lavanya

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

(Approved by AICTE, New Delhi & Affiliated to JNTUH)
(Recognized under section 2(f) of UGC Act 1956) An ISO:9001-2015 Certified Institution
CHILKUR (V), MOINABAD (M), R.R. DIST. T.S-501504

June 2025

GLOBAL
Institute of Engineering & Technology
Approved By AICTE, Affiliated To JNTUH EAMCET CODE: GLOB
A+ GRADE
EDUCATION EXCELLENCE AWARD 2022
ISO 9001 2015
info@globalhyd.edu.in www.globalhyd.edu.in

(Approved by AICTE & Affiliated to JNTUH)
(Recognized under Section 2(f) of UGC Act 1956) An
ISO:9001-2015 Certified Institution
Survey No. 179, Chilkur (V), Moinabad (M), Ranga Reddy Dist. TS.
JNTUH Code (U6) ECE –EEE-CSD-CSM – CSE - CIVIL – ME – MBA - M.Tech EAMCET Code - (GLOB)

Department of Computer Science and Engineering

Noore Ilahi
B. Tech., M. Tech.
Assistant Professor & Head

Date: 02-06-2025

CERTIFICATE

This is to certify that the project work entitled **“Creating a predictive model for forecasting bike-sharing demand, assisting transportation networks in resource allocation and optimizing service availability across city locations.** ”is a bonafide work of **Hamda Hussain (HT.No:23U61A0530),** submitted in partial fulfillment of the requirement for the award of **Bachelor of Technology in Computer Science and Engineering** during the academic year 2024-25. This is further certified that the work done under my guidance, and the results of this work have not been submitted elsewhere for the award of any other degree or diploma.

Internal Guide
Mrs. T Lakshmi Lavanya

Head of the Department
Mrs. Noore Ilahi
Assistant Professor

DECLARATION

I hereby declare that the project work entitled **Creating a predictive model for forecasting bike-sharing demand, assisting transportation networks in resource allocation and optimizing service availability across city locations,** submitted to **Department of Computer Science and Engineering, Global Institute of Engineering & Technology, Moinabad,** affiliated to **JNTUH, Hyderabad** in partial fulfillment of the requirement for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** is the work done by me and has not been submitted elsewhere for the award of any degree or diploma.

Hamda Hussain **(23U61A0530)**

ACKNOWLEDGEMENT

I am thankful to my guide **Mrs. T Lakshmi Lavanya,** Assistant Professor of CSE Department for her valuable guidance for successful completion of this project.

I express my sincere thanks to **Mrs. G. Pavani,** Project Coordinator for giving me an opportunity to undertake the project "**Creating a predictive model for forecasting bikesharing demand, assisting transportation networks in resource allocation and optimizing service availability across city locations**" and for enlightening me on various aspects of my project work and assistance in the evaluation of material and facts. She not only encouraged me to take up this topic but also given her valuable guidance in assessing facts and arriving at conclusions.

I am also most obliged and grateful to **Mrs. Noore Ilahi,** Assistant Professor and Head, Department of CSE for giving me guidance in completing this project successfully.

I express my heart-felt gratitude to our Vice-Principal **Prof. Dr. G Ahmed Zeeshan**, Coordinator Internal Quality Assurance Cell (IQAC) for his constant guidance, cooperation, motivation and support which have always kept me going ahead. I owe a lot of gratitude to him for always being there for me.

I also most obliged and grateful to our Principal **Dr. P. Raja Rao** for giving me guidance in completing this project successfully.

I also thank my parents for their constant encourage and support without which the project would have not come to an end.

Last but not the least, I would also like to thank all my class mates who have extended their cooperation during our project work.

Hamda Hussain **(23U61A0530)**

VISION

The Vision of the Department is to produce professional Computer Science Engineers who can meet the expectations of the globe and contribute to the advancement of engineering and technology which involves creativity and innovations by providing an excellent learning environment with the best quality facilities.

MISSION

M1. To provide the students with a practical and qualitative education in a modern technical environment that will help to improve their abilities and skills in solving programming problems effectively with different ideas and knowledge.

M2. To infuse the scientific temper in the students towards the research and development in Computer Science and Engineering trends.

M3. To mould the graduates to assume leadership roles by possessing good communication skills, an appreciation for their social and ethical responsibility in a global setting, and the ability to work effectively as team members.

PROGRAMME EDUCATIONAL OBJECTIVES

PEO1: To provide graduates with a good foundation in mathematics, sciences and engineering fundamentals required to solve engineering problems that will facilitate them to find employment in MNC's and / or to pursue postgraduate studies with an appreciation for lifelong learning.

PEO2: To provide graduates with analytical and problem solving skills to design algorithms, other hardware / software systems, and inculcate professional ethics, inter-personal skills to work in a multicultural team.

PEO3: To facilitate graduates to get familiarized with the art software / hardware tools, imbibing creativity and innovation that would enable them to develop cutting edge technologies of multi disciplinary nature for societal development.

PROGRAMME OUTCOMES:

PO1: Engineering knowledge: An ability toApply the knowledge of mathematics, science, engineering fundamentals and an engineering specialization to the solution of complex engineering problems.

PO2: Problem analysis: An ability to Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural science and engineering sciences.

PO3: Design/development of solutions: An ability to 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.

PO4: Conduct investigations of complex problems: An ability to 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.

PO5: Modern tool usage: An ability to Create, select and apply appropriate techniques, resources and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

PO6: The engineer and society: An ability to 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.

PO7: Environment sustainability: An ability to Understand the impact of the professional engineering solutions in the societal and environmental contexts, and demonstrate the knowledge of, and the need for sustainable development.

PO8: Ethics: An ability to Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: Individual and teamwork: An ability to Function effectively as an individual and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10: Communication: An ability to Communicate effectively on complex engineering activities with the engineering community and with society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: Project management and finance: An ability to Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: Lifelong learning: An ability to Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broader context of technological change.

PROGRAMME SPECIFIC OUTCOMES

PSO1: An Ability to Apply the fundamentals of mathematics, Computer Science and Engineering Knowledge to analyze and develop computer programs in the areas related to Algorithms, System Software, Web Designing, Networking and Data mining for efficient Design of computer-based system to deal with Real time Problems.

PSO2: An Ability to implement the Professional Engineering solutions for the betterment of Society, and able to communicate with professional Ethics effectively

ABSTRACT

Abstract

Accurately forecasting bike-sharing demand is essential for optimizing resource allocation and improving service availability within urban transportation networks. This study presents a predictive model based on the Random Forest algorithm to estimate bike rental demand across city locations. Leveraging historical usage data, weather conditions, temporal variables, and station-specific features, the Random Forest model captures complex, non-linear relationships influencing hourly and daily bike usage patterns. The model's ensemble approach aggregates predictions from multiple decision trees, enhancing robustness and predictive accuracy compared to traditional regression methods. Empirical results demonstrate that the Random Forest model outperforms alternative algorithms in predicting station-level and season-specific demand, enabling transportation operators to proactively allocate bikes, reduce waiting times, and adapt to fluctuating urban mobility needs. This approach supports data-driven decision-making for dynamic fleet management and strategic planning, ultimately fostering more efficient and sustainable bike-sharing systems.

TABLE OF CONTENTS

Chapter	Particular	Page Number
	Title Page Certificate Declaration Acknowledgement Vision Mission Abstract	i ii iii iv v-vi vii
1	**INTRODUCTION** 1.1 Existing System 1.2 Disadvantages of Existing system 1.3 Proposed System 1.4 Advantages of Proposed System	**1-4**
2	**LITERATURE SURVEY**	**5-8**
3	**SYSTEM ANALYSIS**	**9-13**
4	**SYSTEM DESIGN**	**14-26**
5	**SYSTEM IMPLEMENTATION**	**27-41**
6	**SYSTEM TESTING**	**42-46**
7	**RESULTS**	**47-50**
8	**CONSLUSION**	**51**
9	**FUTURE ENHANCEMENT**	**52-53**
REFERENCES		**54**

LIST OF FIGURES

Figure Number	Figure Name	Page Number
1	System Architecture	14
2	Data Flow Diagram	15-16
3	Use Case diagram	18
4	Component Diagram	19-20
5	ER Diagram	21-22
6	Class Diagram	23
7	Activity Diagram	24
8	Sequence Diagram	25-26

LIST OF TABLES

Table Number	Table Name	Page Number
1	Test Cases	46

CHAPTER 1
INTRODUCTION

Urban transportation systems are undergoing a significant transformation driven by the rise of smart mobility technologies. These advancements aim to address the increasing challenges of **traffic congestion**, **environmental degradation**, **urbanization**, and the growing demand for **flexible, affordable, and accessible transit options**. Traditional transportation modes, while effective to an extent, are increasingly being supplemented or replaced by innovative systems that align with sustainability goals and smart city initiatives.

One such innovation is the **bike-sharing system**, a rapidly growing mode of urban mobility that provides users with shared access to bicycles on a short-term basis. These systems are strategically distributed throughout a city, allowing riders to pick up a bike at one station and drop it off at another. This not only encourages a shift away from private car use, thereby reducing greenhouse gas emissions and urban noise pollution, but also promotes healthier lifestyles and more livable cities.

Despite their numerous benefits, bike-sharing systems come with operational challenges—chief among them is ensuring the **efficient and timely distribution of bikes across docking stations**. The demand for bikes can fluctuate significantly based on various factors such as time of day, weather conditions, weekdays versus weekends, and public holidays. Failure to meet user demand in real time can lead to **stations running empty or full**, causing inconvenience to users and reducing system reliability. This makes **accurate demand forecasting** not just beneficial but essential.

To address this challenge, the primary objective of this project is to develop a **predictive model** capable of forecasting bike-sharing demand with a high degree of precision. The goal is to empower transportation authorities, service providers, and logistics teams to make informed decisions about **fleet distribution**, **station balancing**, and **capacity planning**. This model leverages **machine learning algorithms** trained on historical bike usage data, enriched with contextual information such as **weather parameters (e.g., temperature, rainfall, wind speed)**, **calendar effects (e.g., holidays, day of the week)**, and **temporal trends (e.g., hour of the day)**.

By implementing a **data-driven, intelligent forecasting system**, this project aims to bridge the gap between fluctuating user demand and operational readiness. The predictive insights generated will support **dynamic rebalancing strategies**, improve overall **user satisfaction**, and enhance the **sustainability and efficiency of urban mobility networks**. This solution not only aligns with the goals of smart transportation infrastructure but also serves as a blueprint for scalable deployment in other shared mobility domains, such as scooters and e-bikes.

1.1 EXISTING SYSTEM:

Traditional methods for bike-sharing demand forecasting, such as linear regression, moving averages, and seasonal decomposition, offer some predictive power but struggle with complex, dynamic factors. These models assume stable patterns and fail to adapt to nonlinear relationships influenced by weather, holidays, traffic conditions, and socioeconomic trends. As a result, they often provide inaccurate forecasts, leading to misallocated resources—either excess bikes in low-demand zones or shortages in highdemand areas.

Existing bike-sharing systems typically integrate basic predictive tools into apps and dashboards, but these systems lack real-time adaptability. Without advanced machine learning techniques, they fail to account for sudden demand fluctuations caused by events like storms or citywide gatherings. Additionally, most platforms do not leverage dynamic resource allocation, making bike distribution inefficient and reducing service reliability.

To address these shortcomings, modern forecasting approaches incorporate machine learning models such as neural networks (RNNs, LSTMs) for temporal dependencies, graph convolutional networks (GCNs) for spatial correlations, and reinforcement learning for optimizing bike distribution. These models enable more accurate demand predictions by analyzing historical usage trends and real-time external factors.

Furthermore, integrating IoT-enabled bike stations, weather APIs, and traffic sensors allows platforms to process live data and make quick adjustments. This leads to smarter fleet management, reduced operational costs, and improved user satisfaction. By embracing AI-driven forecasting, bike-sharing networks can enhance efficiency, ensuring availability aligns with demand and making urban mobility more seamless and sustainable.

1.2 DISADVANTAGES OF EXISTING SYSTEM

- Inability to account for non-linear patterns in usage data: Traditional forecasting models, such as linear regression, often assume a steady, predictable relationship between factors influencing demand. However, bike-sharing usage is affected by complex patterns, such as sudden surges during large public events, seasonal variations, and shifts in commuter behavior. Without machine learning techniques capable of recognizing these non-linear dependencies, predictions remain overly simplistic and fail to capture real-world demand fluctuations.

- Low responsiveness to temporal and contextual fluctuations: Many existing models struggle to adapt to rapidly changing conditions, such as weather changes, holidays, or the day of the week. For example, unexpected rain can drastically reduce demand, whereas a sunny weekend might increase leisure-based bike usage. Systems that fail to incorporate real-time data streams or contextual insights often produce rigid, outdated predictions, making bike-sharing networks less efficient in responding to demand shifts.
- Poor accuracy in short-term and long-term predictions: Short-term forecasting often relies on historical averages, failing to account for sudden shifts due to city-wide disruptions or transport infrastructure changes. Long-term predictions, on the other hand, frequently overlook emerging trends such as urban expansion or shifting commuter habits. Without integrating advanced algorithms, forecasting systems suffer from inaccuracies that lead to inefficient operational planning, negatively impacting availability and service reliability.
- Inefficient resource distribution and frequent shortages or surpluses: Misallocating bikes across different locations is a major issue stemming from flawed demand predictions. Some areas may experience an overflow of bikes due to overestimation, while high-demand zones suffer from shortages, frustrating users. Advanced models that incorporate spatial analytics and reinforcement learning can optimize distribution strategies, ensuring balanced bike availability and improved user experience.

1.3 PROPOSED SYSTEM:

Traditional forecasting methods often struggle to capture the complex and dynamic nature of bike-sharing demand, leading to inefficiencies in resource allocation. To address these challenges, this project introduces a machine learning-based predictive model that leverages historical usage data, weather conditions, and temporal patterns to generate highly accurate demand forecasts. By utilizing advanced algorithms such as Random Forest, Gradient Boosting, and XGBoost, the model can identify intricate relationships between various factors, allowing for more precise predictions and improved operational efficiency.

A key component of this system is feature engineering, which transforms raw data into meaningful insights. This process includes handling missing values, encoding categorical variables, and normalizing numerical attributes to ensure the model can effectively interpret diverse data sources. Additionally, time series analysis techniques are employed to recognize trends and seasonal variations, while cross-validation methods enhance the model's robustness by preventing overfitting and ensuring reliable performance across different scenarios.

To maximize usability, the final predictive model will be deployed through a web-based interface, providing real-time forecasts and interactive visualizations. This platform will enable decision-makers to dynamically allocate resources based on live demand fluctuations, reducing shortages in high-traffic areas and preventing excess supply in lowdemand zones. By integrating real-time data streams and machine learning-driven insights, the system aims to optimize bike-sharing operations, enhance user satisfaction, and contribute to more sustainable urban mobility solutions.

1.4 ADVANTAGES OF PROPOSED SYSTEM:

* Utilizes advanced machine learning algorithms for improved prediction accuracy

* Considers multiple influencing factors such as weather, holidays, and time slots

* Enhances operational efficiency by aligning bike availability with predicted demand

* Facilitates data-driven decision-making for city transportation department

CHAPTER 2

LITERATURE SURVEY

2.1 LITERATURE SURVEY

2.1.1 Deep Learning-Based Approaches

Author : Weiwei Jiang

Deep learning has revolutionized the way time-series and sequential data are analyzed, offering advanced solutions for complex forecasting problems such as predicting demand in bike-sharing systems. According to Weiwei Jiang, the application of deep learning models—particularly Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks—enables more accurate and intelligent demand prediction by learning from historical data patterns and their temporal dependencies.

Unlike traditional machine learning algorithms that rely heavily on manual feature engineering, deep learning models automatically extract hierarchical features from raw input data, thereby reducing human bias and increasing model adaptability. LSTMs, in particular, are highly effective in handling sequential datasets due to their ability to retain information across long time intervals without suffering from vanishing gradient problems. This makes them especially suitable for modeling demand patterns that fluctuate throughout the day, across seasons, and during special events.

Jiang emphasizes that incorporating contextual features such as weather conditions (temperature, humidity, windspeed), calendar variables (holiday, working day, season), and even user behavior data (e.g., casual vs. registered usage) significantly enhances the model's performance. These inputs allow the neural network to understand how external variables impact rental patterns and adjust predictions accordingly.

Furthermore, the scalability of deep learning frameworks allows them to be deployed in realtime prediction environments. With continuous data flow, such systems can dynamically update their forecasts, aiding in immediate decision-making for tasks such as bike redistribution, staffing at rental hubs, and service availability adjustments. Deep learning models can also be integrated with geospatial analytics to provide station-specific demand forecasts, which are crucial for large-scale bike-sharing networks spread across metropolitan areas.

Weiwei Jiang's approach highlights the importance of combining data-driven forecasting with intelligent algorithms to create robust, adaptive, and highly accurate prediction models. This not only helps in improving the efficiency of the bike-sharing infrastructure but also supports the broader vision of sustainable urban mobility by ensuring better user experience, reducing wait times, and optimizing resource utilization across the transportation ecosystem.

2.1.2 **Machine Learning for Inventory Management**

Author : Daniele Gammelli,Yihua Wang,Dennis Prak,Filipe Rodrigues,Stefan Minner,Francisco Camara Pereira

Inventory management plays a crucial role in maintaining the efficiency and costeffectiveness of supply chains, including transportation and bike-sharing services. In recent years, machine learning has emerged as a powerful tool for optimizing inventory decisions, moving beyond traditional rule-based systems toward predictive, data-driven frameworks. The work of Daniele Gammelli, Yihua Wang, Dennis Prak, Filipe Rodrigues, Stefan Minner, and Francisco Camara Pereira highlights the practical implementation of machine learning models in solving complex inventory management challenges.

Their research emphasizes the integration of historical sales or demand data with contextual variables—such as weather conditions, time-based trends, promotions, and user behavior—to improve the accuracy of demand forecasting. This is particularly valuable in systems like bike-sharing, where demand varies significantly based on time of day, day of the week, season, and location. By leveraging supervised learning models such as Random Forests, Gradient Boosting Machines, and Support Vector Machines, it is possible to predict shortterm and long-term inventory needs with high precision.

One of the key contributions of their work is the application of probabilistic and ensemble learning methods to handle uncertainty in demand forecasts. These models are capable of generating not only point estimates but also prediction intervals, which help decision-makers assess risk and make more robust stocking or resource allocation decisions. Furthermore, their research introduces scalable optimization frameworks that integrate machine learning outputs into decision-support tools, enabling real-time adjustments to inventory levels based on current and forecasted conditions.

In the context of bike-sharing, this translates into smarter fleet distribution, ensuring that stations are neither understocked (causing service failures) nor overstocked (resulting in inefficiencies). For example, demand predictions can be used to pre-position bikes in highdemand areas during peak hours or events, thus reducing lost rental opportunities and improving user satisfaction.

The authors also highlight the importance of model interpretability and the ability to update predictions dynamically as new data becomes available. This is especially relevant in fastpaced urban environments, where external conditions can shift rapidly. Their work represents a significant advancement in the intersection of machine learning and inventory optimization, offering scalable solutions that enhance operational efficiency, minimize waste, and improve service delivery in complex systems such as smart transportation networks.

2.1.3 Case Studies on Demand Prediction

Author : Yin, Y., Ravi, S. S., Yue, Y., Froehlich, J., Oliver, N., Xu, Y., He, Z., Li, X., Zheng, Y., Liu, F., Hwang, H., Toole, J. L., Ulm, M., Gonzalez, M. C., Bauer, D., Singhvi, V., Yoon, J., Ghosh, S., Wang, H., Zhou, X.

Numerous case studies have been conducted globally to explore the efficacy of various models and data-driven techniques in predicting demand across shared mobility and transportation networks. These studies reflect the evolving landscape of urban mobility and highlight how integrating contextual data—such as weather, holidays, time of day, and geolocation—can significantly improve the accuracy of demand forecasts. Authors like Yin et al. and Zheng et al. have explored large-scale urban datasets to build predictive models for bike-sharing and ride-sharing systems, while Froehlich and Oliver investigated early urban computing applications using sensor and usage data. Meanwhile, Toole, Gonzalez, and their colleagues used big data analytics to understand mobility behavior and optimize resource distribution in bike-sharing systems. Collectively, these case studies validate the importance of contextual awareness, spatial-temporal modeling, and real-world validation when developing robust demand forecasting systems for modern transportation infrastructures.

2.2 ABOUT PYTHON

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. An interpreted language, Python has a design philosophy that emphasizes code readability (notably using whitespace indentation to delimit code blocks rather than curly brackets or keywords), and a syntax that allows programmers to express concepts in fewer lines of code than might be used in languages such as C++or Java. It provides constructs that enable clear programming on both small and large scales. Python interpreters are available for many operating systems. CPython, the reference implementation of Python, is open source software and has a communitybased development model, as do nearly all of its variant implementations. CPython is managed by the non-profit Python Software Foundation. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

CHAPTER 3

SYSTEM ANALYSIS

This project aims to develop a machine learning-based predictive model for forecasting bike-sharing demand, assisting transportation networks in resource allocation, and optimizing service availability across city locations. By leveraging historical usage data, weather conditions, and traffic patterns, the model will use algorithms such as Random Forest, Gradient Boosting, and XGBoost to generate accurate demand forecasts. Feature engineering techniques will refine raw data, while time series analysis and cross-validation will enhance robustness. The final model will be deployed through a web-based interface, enabling real-time predictions and visualizations to support dynamic decision-making, improve operational efficiency, and ensure optimal bike distribution.

3.1 REQUIREMENT SPECIFICATIONS

3.1.1 HARDWARE REQUIREMENTS:

- **Processor** : Intel Core i5/i7 ✞ **RAM** : 8 GB.
- **Storage** : 100 GB (HDD or SSD)
- **Monitor** : 14’ Colour Monitor.
- **Mouse** : Optical Mouse.
- **GPU** : Optional (for deep learning models).

3.1.2 SOFTWARE REQUIREMENTS:

- **Operating system** : Windows 10 or Linux (Ubuntu preferred).
- **Coding Language** : Python.
- **Front-End** : Python.
- **Designing** : Html , Css , javascript.

✞ **Libraries** : Pandas, NumPy, Scikit-learn, XGBoost, Matplotlib, Seaborn

✞ **IDE** Jupyter Notebook / PyCharm/VS Code ✞ **Database**

CSV / SQLite (for local data)

✞ 3.1.3 FUNCTIONAL REQUIREMENTS:

* Data ingestion and preprocessing

* Feature extraction and engineering

* Model training and evaluation

* Prediction and visualization dashboard

* User login and result access interface

Operating Systems supported

1. Windows 7
2. Windows XP
3. Windows 8

Technologies and Languages used to Develop

1. Python

Debugger and Emulator

 Any Browser (Particularly Chrome)

3.2 FEASIBILITY STUDY:

Bike-sharing systems play a crucial role in urban transportation, offering a convenient and sustainable mode of travel. However, managing demand fluctuations, optimizing resource allocation, and ensuring service availability present significant challenges. Traditional forecasting methods often struggle to capture dynamic usage patterns influenced by external factors such as weather, time of day, and socioeconomic trends. This feasibility study explores the development of a machine learning-based predictive model designed to improve demand forecasting, assist transportation networks in efficient resource distribution, and enhance service reliability. By leveraging historical usage data, real-time analytics, and advanced algorithms such as Random Forest, Gradient Boosting, and XGBoost, the proposed system aims to provide accurate predictions, optimize fleet management, and reduce inefficiencies in bike-sharing networks. The study will assess the technical, operational, and economic feasibility of implementing such a system, ensuring its scalability and effectiveness in real-world urban mobility scenarios.

Three key considerations involved in the feasibility analysis are,

- **ECONOMICAL FEASIBILITY**
- **TECHNICAL FEASIBILITY**
- **SOCIAL FEASIBILITY**

3.2.1 ECONOMICAL FEASIBILITY:

The development of a **predictive model for forecasting bike-sharing demand** must be economically viable to ensure long-term sustainability and cost-effectiveness. This feasibility study examines the financial implications of implementing a **machine learning-based forecasting system** and its potential return on investment (ROI) for transportation networks. The initial costs include **data collection infrastructure**, such as IoT-enabled bike stations, weather monitoring systems, and traffic sensors, alongside **computing resources** for model training and deployment. The operational expenses cover **data storage, real-time processing**, and periodic model updates to maintain accuracy. However, the benefits outweigh these costs, as accurate demand forecasting minimizes inefficiencies, reducing excess inventory costs, optimizing bike redistribution, and enhancing service availability.

By improving fleet management and reducing shortages, bike-sharing operators can boost **user satisfaction and revenue**, making the system economically feasible. Additionally, the scalability of machine learning models allows cities to expand services without significant increases in operational costs. The **economic feasibility** of this project is further strengthened by potential **government grants, smart city initiatives**, and **partnerships with urban mobility providers**, ensuring financial sustainability while enhancing transportation efficiency.

.

3.2.2 TECHNICAL FEASIBILITY:

Developing a **predictive model for forecasting bike-sharing demand** requires assessing the technical feasibility of implementing machine learning algorithms, data integration, and system scalability. The project relies on **historical usage data, weather patterns, traffic conditions, and socioeconomic factors**, which must be collected, processed, and analyzed efficiently.

The model will utilize **machine learning techniques** such as **Random Forest, Gradient Boosting, XGBoost, and deep learning architectures like LSTMs and Graph Neural Networks** to capture complex demand patterns. **Feature engineering** will refine raw data, handling missing values, encoding categorical variables, and normalizing numerical attributes to improve prediction accuracy.

To ensure **real-time forecasting**, the system will integrate **IoT-enabled bike stations, weather APIs, and traffic sensors**, allowing dynamic adjustments based on live data. The model will be deployed on **cloud-based infrastructure**, ensuring scalability and computational efficiency. A **web-based dashboard** will provide interactive visualizations, enabling transportation networks to make informed decisions on bike distribution and service optimization.

Overall, the technical feasibility of this project is strong, given the availability of **advanced machine learning frameworks, cloud computing resources, and real-time data integration technologies**. The system's success depends on robust data preprocessing, efficient model training, and seamless deployment to enhance urban mobility and optimize bike-sharing operations.

.

3.2.3 SOCIAL FEASIBILITY:

The implementation of a **predictive model for forecasting bike-sharing demand** must align with societal needs and urban mobility goals. Social feasibility assesses how well the system integrates with existing transportation networks, enhances accessibility, and improves user satisfaction. By optimizing bike distribution, the model ensures equitable access to shared mobility, benefiting commuters, tourists, and underserved communities.

Additionally, the system promotes **sustainable urban transportation**, reducing reliance on motor vehicles and lowering carbon emissions. Public acceptance is crucial, and integrating user-friendly interfaces, real-time availability updates, and responsive service adjustments fosters trust and engagement. Collaboration with city planners, policymakers, and transportation authorities ensures that the model supports broader urban development initiatives.

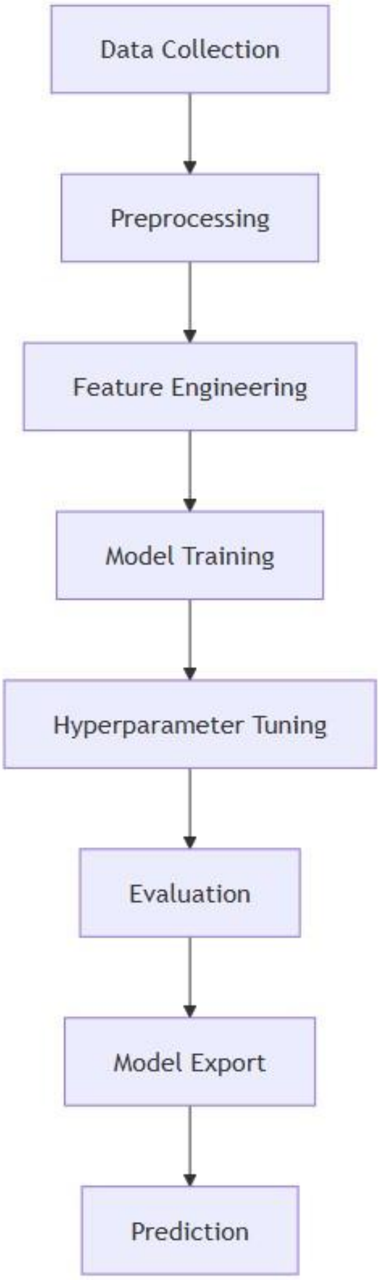
Overall, the project is socially feasible as it enhances **transportation efficiency, environmental sustainability, and equitable mobility**, making bike-sharing more accessible and reliable across city locations. Would you like insights on potential challenges or stakeholder engagement strategies?

.

CHAPTER 4

SYSTEM DESIGN

4.1SYSTEM ARCHITECTURE:

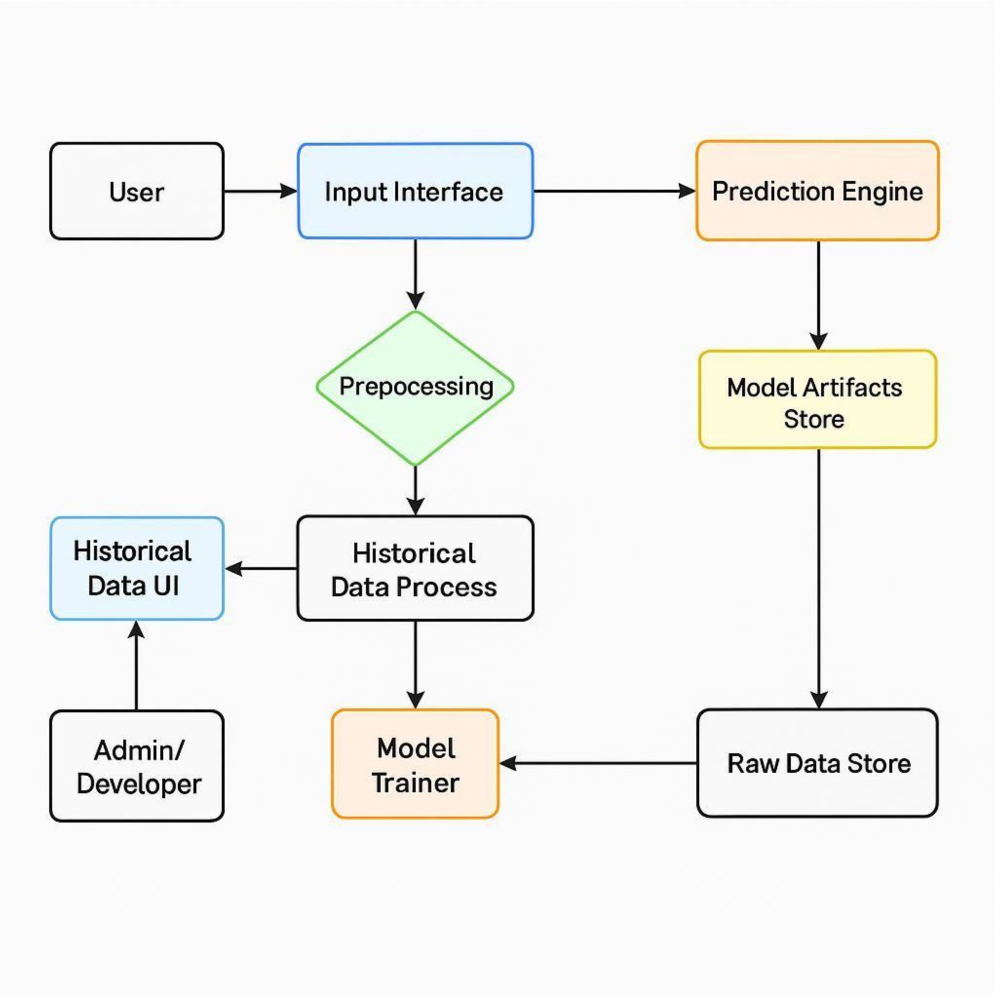


4.1.1 System Architecture

4.2 DATA FLOW DIAGRAM:

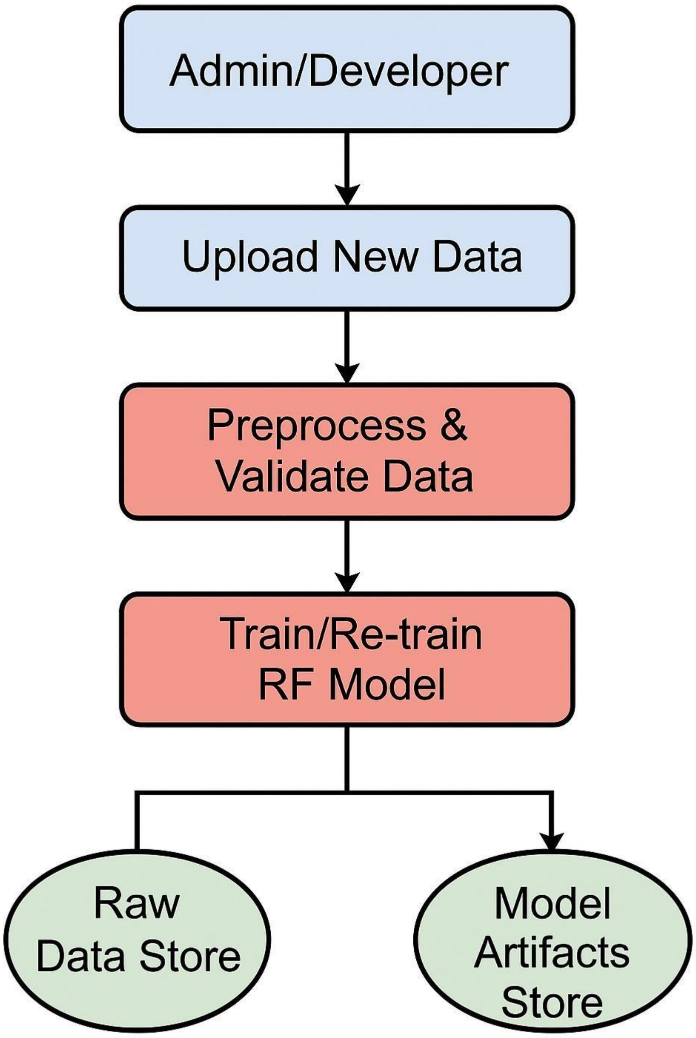
A Data Flow Diagram (DFD) for the Bike Rental Prediction System visually illustrates how data moves through the system from input to prediction. It starts with the user or developer interacting with the system—users input weather and timerelated data to request bike rental predictions or view historical usage, while developers can upload new datasets to retrain the model. The input data is first preprocessed (scaled and encoded), then passed to the prediction engine where a trained Random Forest Regression model forecasts rental counts. Predictions and historical data are retrieved from storage components, including the raw dataset, model artifacts (like encoders and scalers), and optionally, a prediction log. The DFD helps clarify data transformations

- 1. User



4.2.1 Data Flow Diagram User

2. Admin



4.2.2 Data Flow Diagram Admin

4.3 UML Diagrams:

Unified Modeling Language (UML) diagrams provide a standardized way to visualize the structure and behavior of a software system. In the context of a Bike Rental Prediction System,

UML diagrams help define the system's components, their relationships, and interactions, offering clarity for development and maintenance.

GOALS:

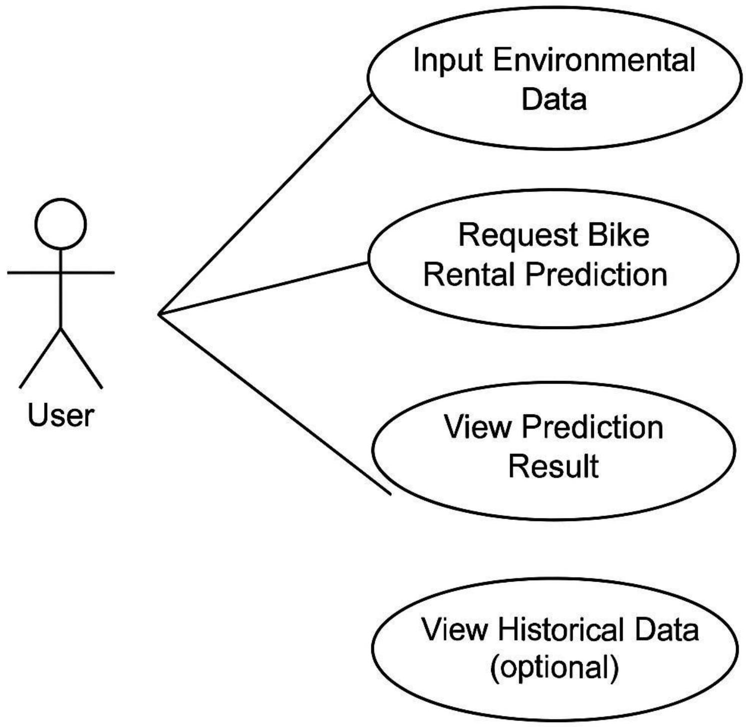
The Primary goals in the design of the UML are as follows:

1. Clear Communication of System Design
   - UML diagrams provide a standardized visual language to communicate the structure and behavior of the system among developers, analysts, and stakeholders.
2. Modularization and Component Understanding
   - Class diagrams define key modules like the prediction engine, preprocessing logic, and data handling components, making the system's architecture easier to understand and maintain.
3. Defining User Roles and Interactions
   - Use Case diagrams help identify different actors (e.g., user, admin) and their interactions with the system, clarifying system boundaries and expected functionalities.
4. Visualizing System Workflow o Sequence and Activity diagrams describe the logical flow of operations, from input collection to prediction generation, enabling teams to visualize and refine system behavior.
5. Supporting Implementation and Refactoring
   - UML diagrams guide coding efforts by offering a blueprint of classes, methods, and data flow. They are also useful during refactoring or upgrading the system.
6. Facilitating Requirement Validation
   - Diagrams help validate if the system meets functional requirements by mapping them visually, ensuring nothing is missed during development.
7. Enhancing Testing and Debugging
   - Understanding the flow and dependencies in the system helps testers identify potential points of failure and plan effective test cases.

4.3.1 USE CASE DIAGRAM

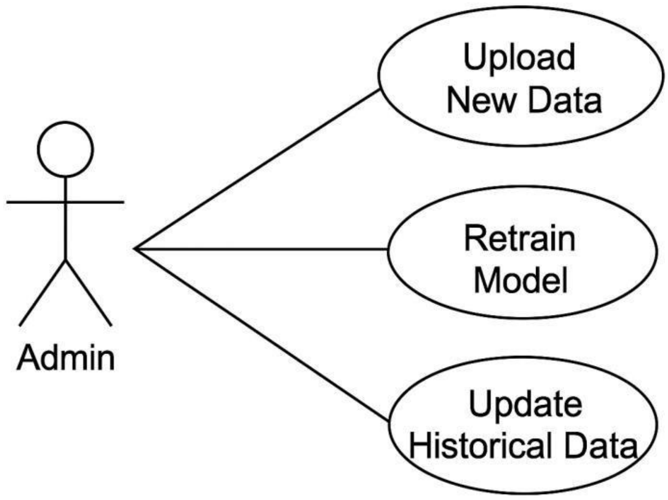
A **Use Case Diagram** represents the system’s functionality from an end-user perspective. It identifies actors (like users and admins) and their interactions with the system such as requesting predictions, viewing data, and retraining models. This diagram clarifies system requirements and boundaries, which is essential for planning and communication with stakeholders.

a. User



4.3.1.1 Use Case Diagram User

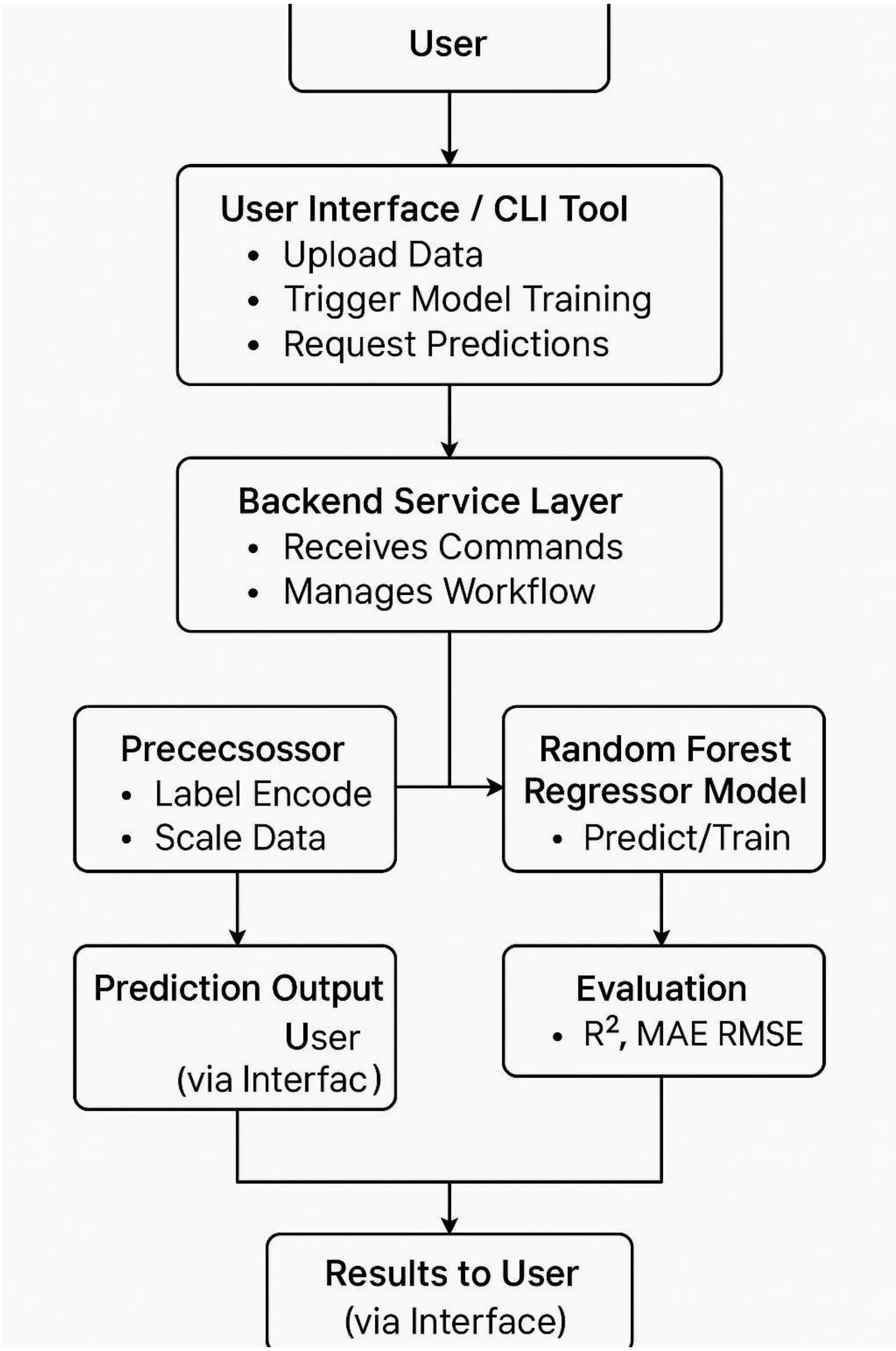
b. Admin



4.3.1.2 Use Case Diagram Admin

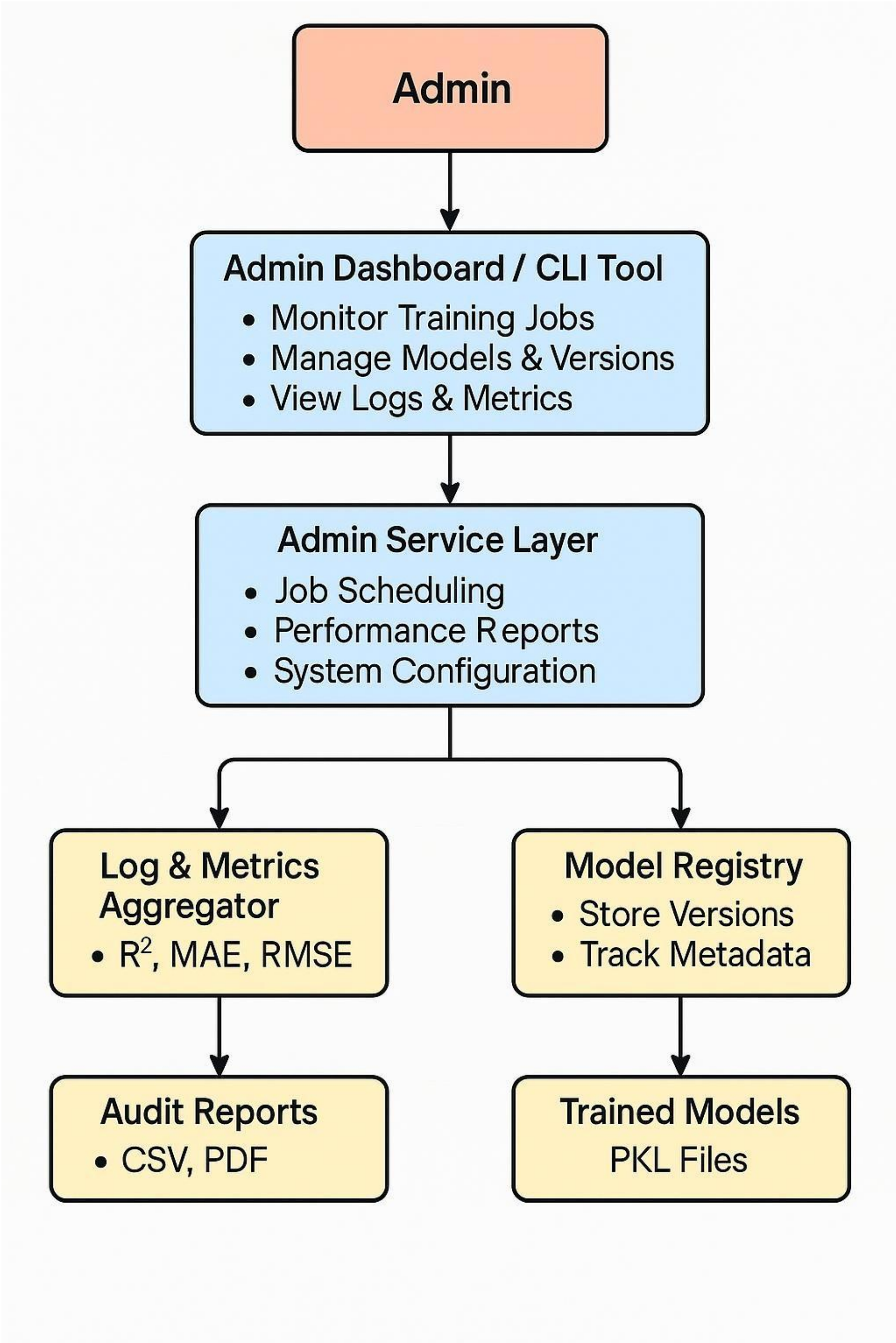
4.3.2 COMPONENT DIAGRAM

a. User



4.3.2.1 Component Diagram User

b. Admin

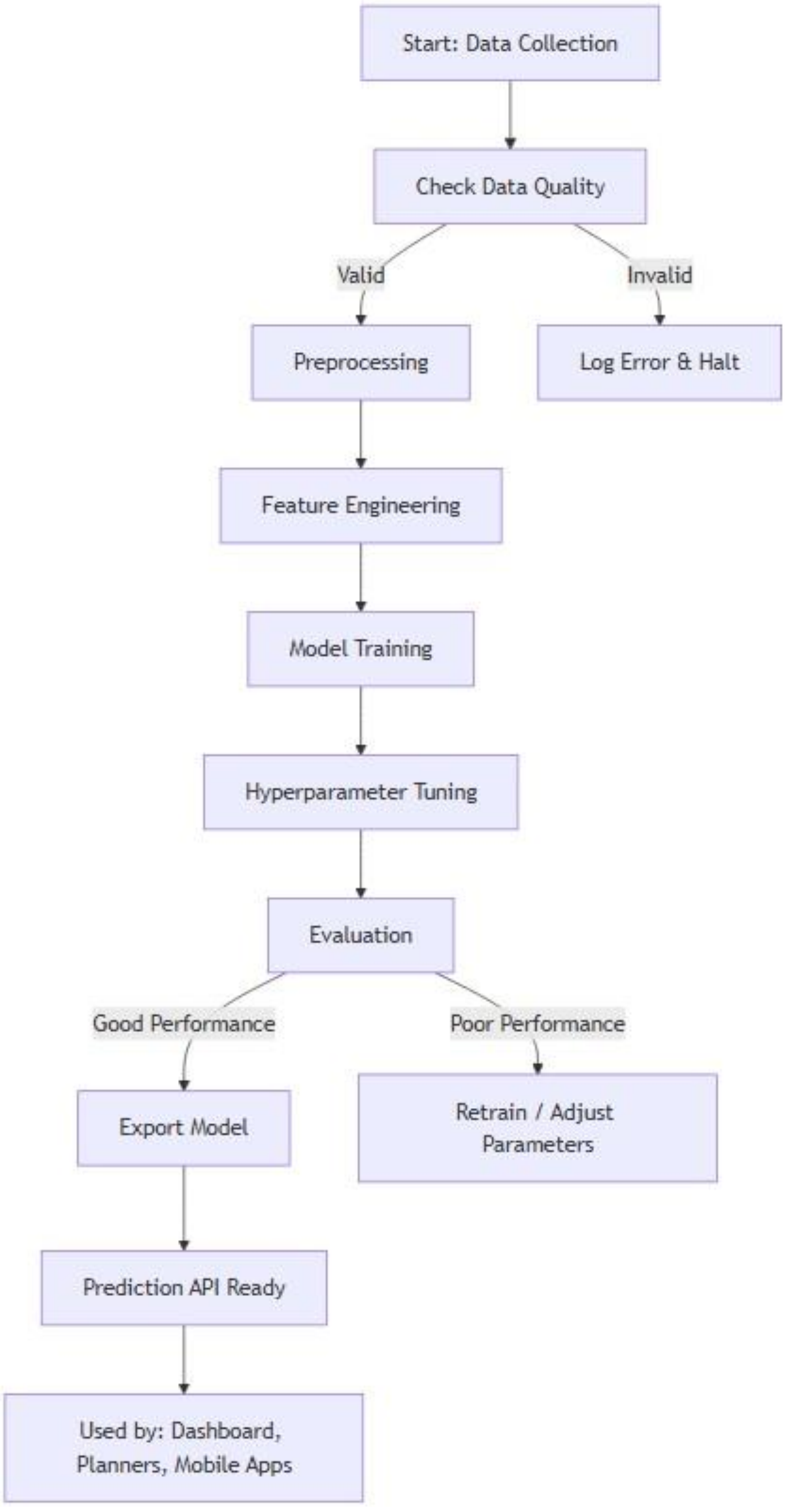


4.3.2.2 Component Diagram Admin

4.3.3 ER DIAGRAM

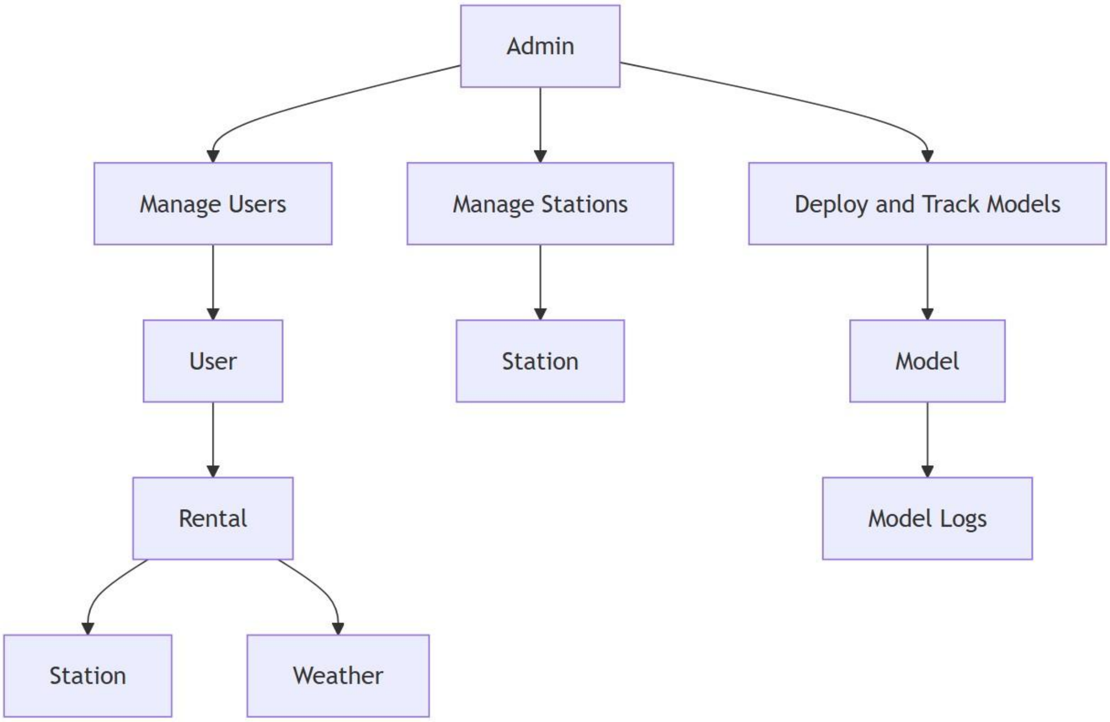
a.

User



4.3.3.1 ER Diagram User

b. Admin

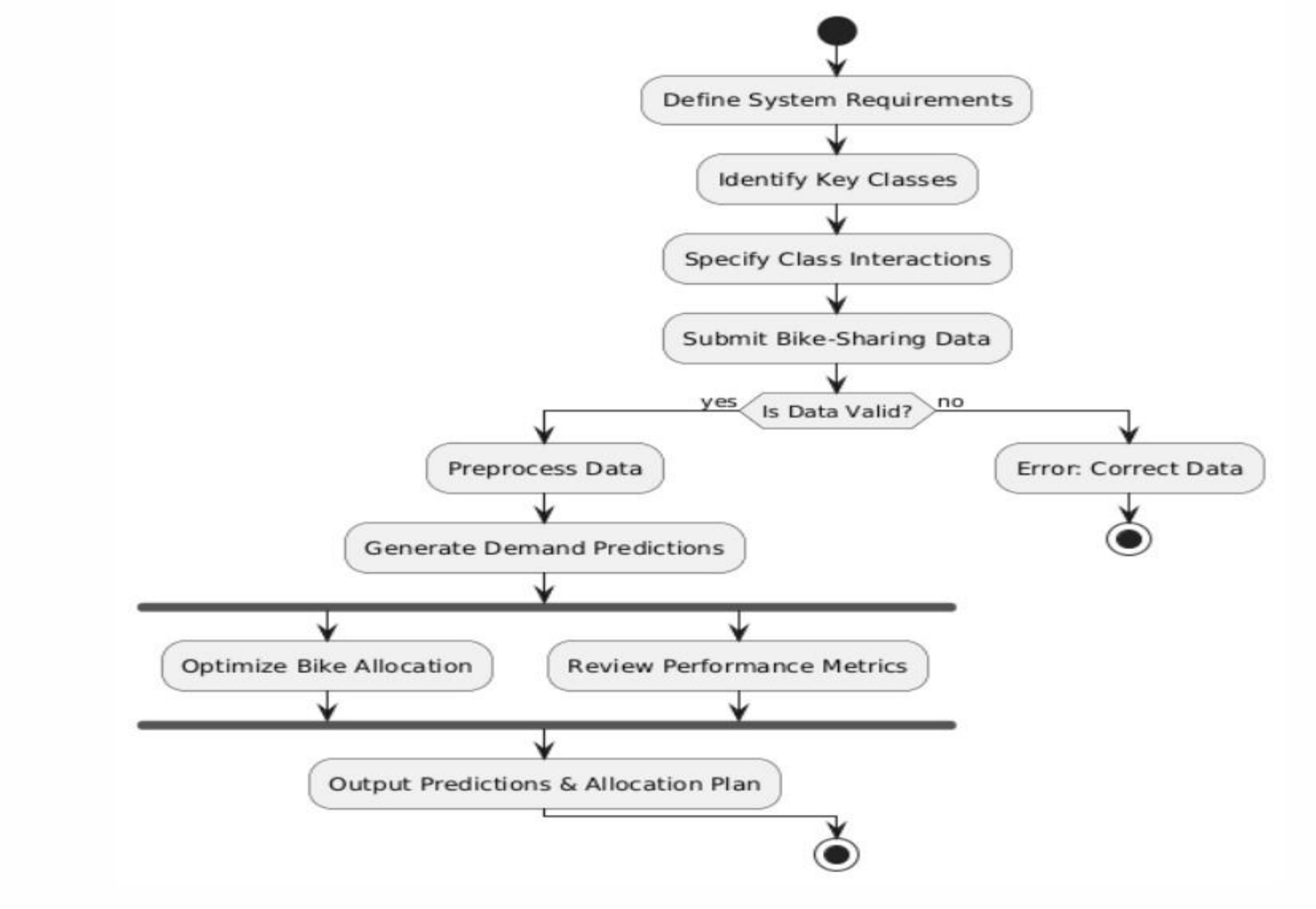


4.3.3.2 ER Diagram Admin

4.3.4 CLASS DIAGRAM

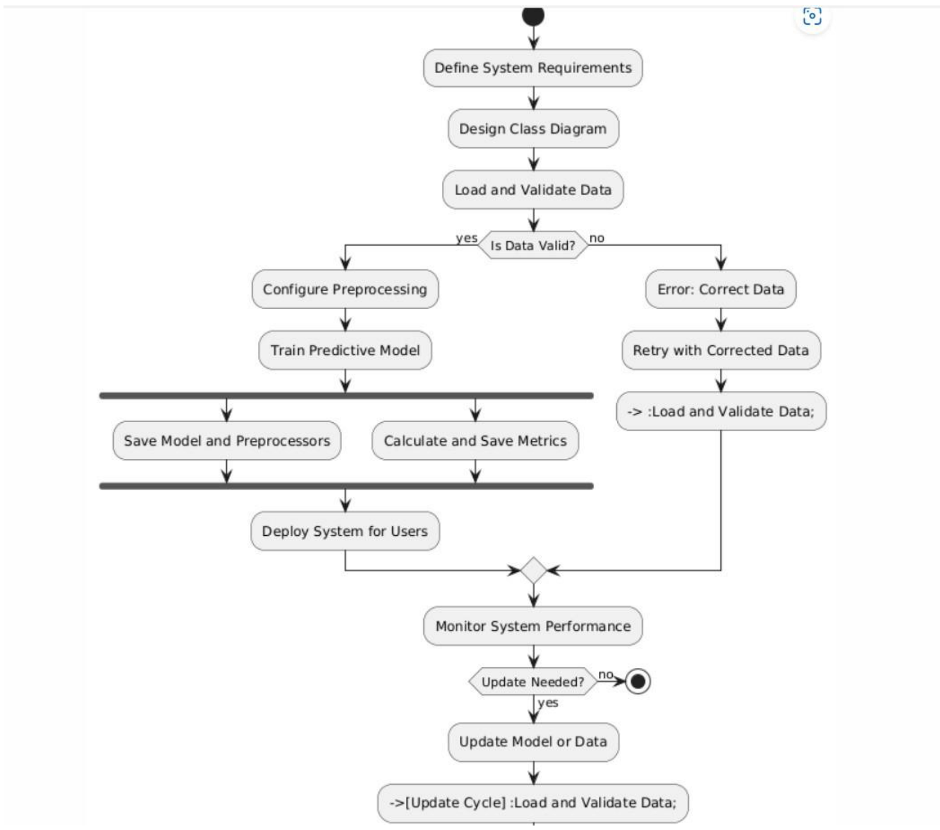
A **Class Diagram** shows the static structure of the system by modeling classes, attributes, methods, and the relationships between classes. For the bike rental system, it includes classes such as BikeRentalPredictor, Preprocessor, and EncoderManager. These classes handle tasks like loading the Random Forest model, preprocessing inputs, and encoding categorical features. This diagram helps in understanding the modular components and how they work together.

1. User



4.3.4.1 Class Diagram user

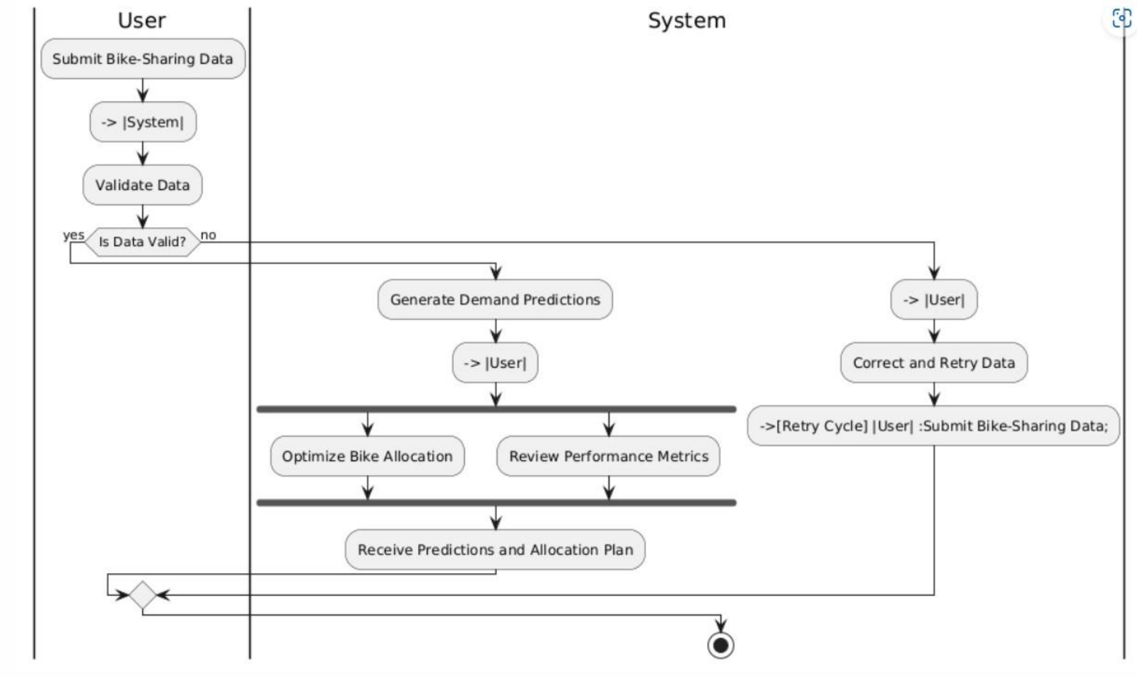
2.Admin



4.3.4.1 Class Diagram admin

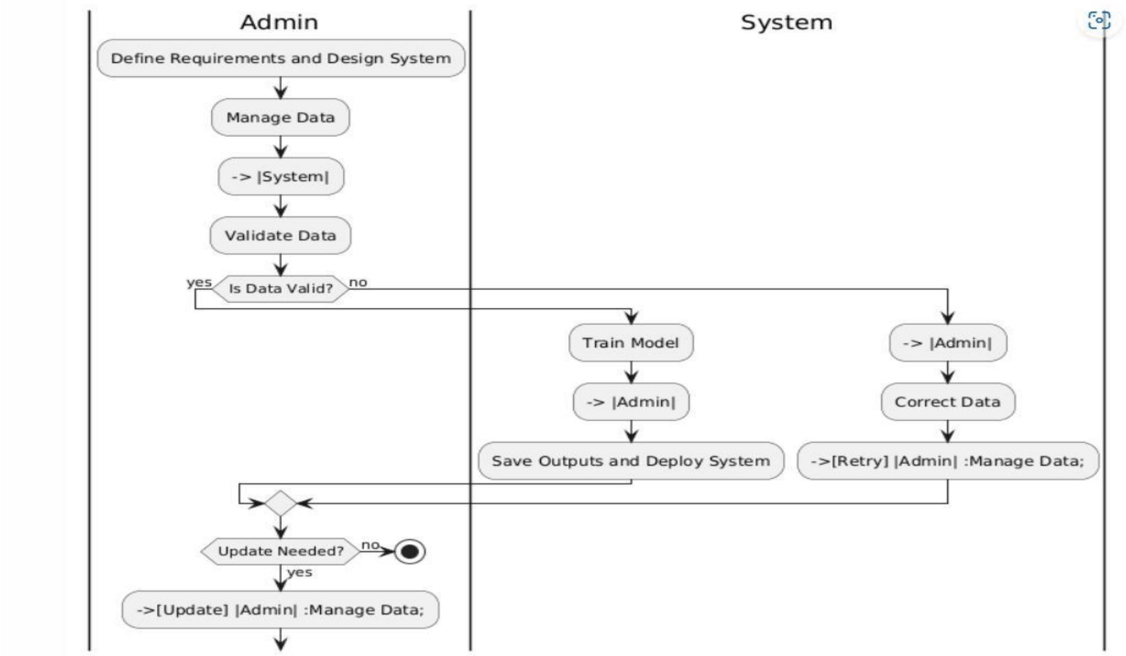
4.3.5ACTIVITY DIAGRAM

An **Activity Diagram** shows the workflow of activities and decisions within the system. For example, it models the process of receiving input, preprocessing data, predicting rentals, and returning the output. It supports identifying parallel processes and decision points in the system. a. User



2. Activity Diagram User

b.Admin

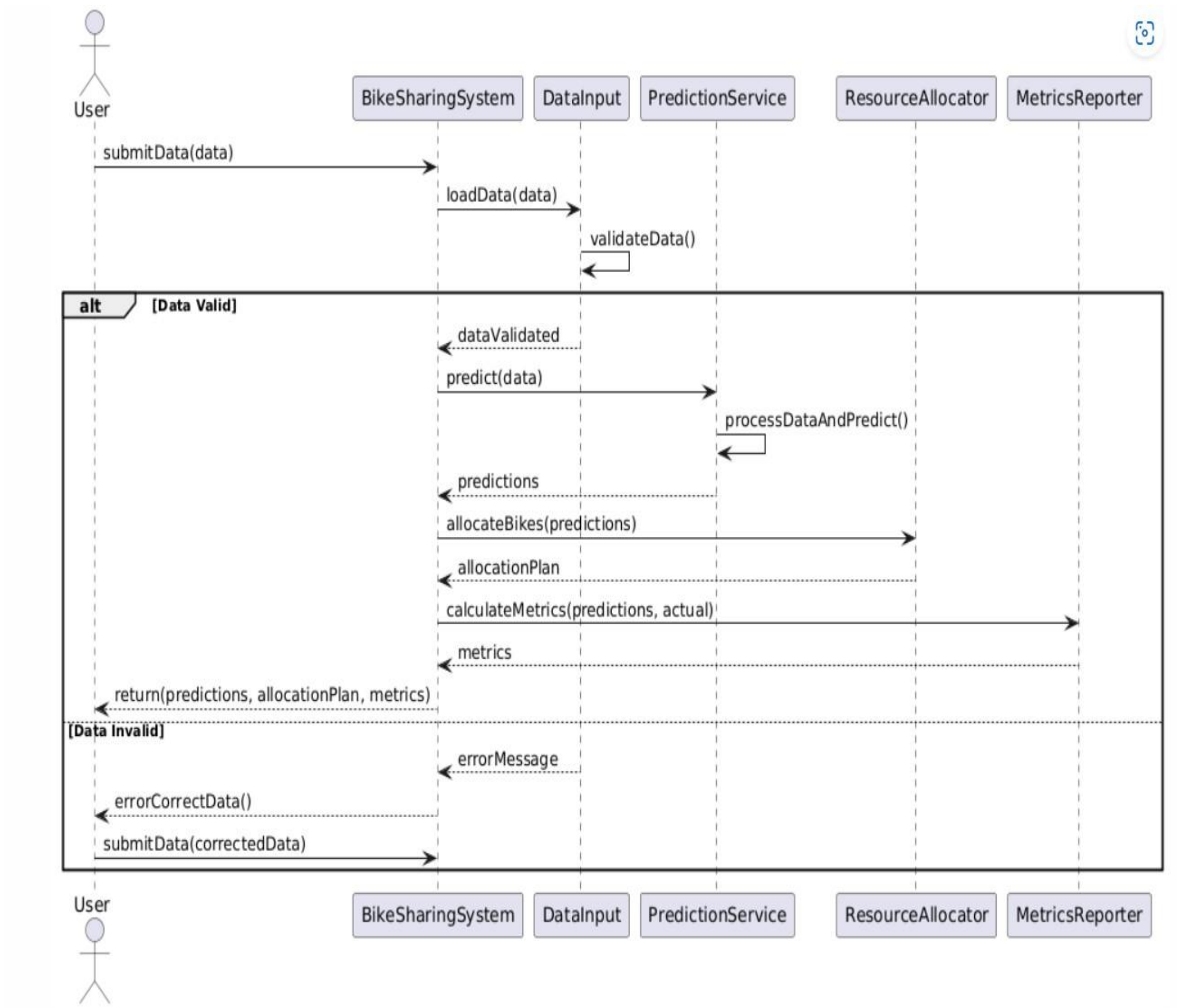


3. Activity Diagram Admin

4.3.6SEQUENCE DIAGRAM

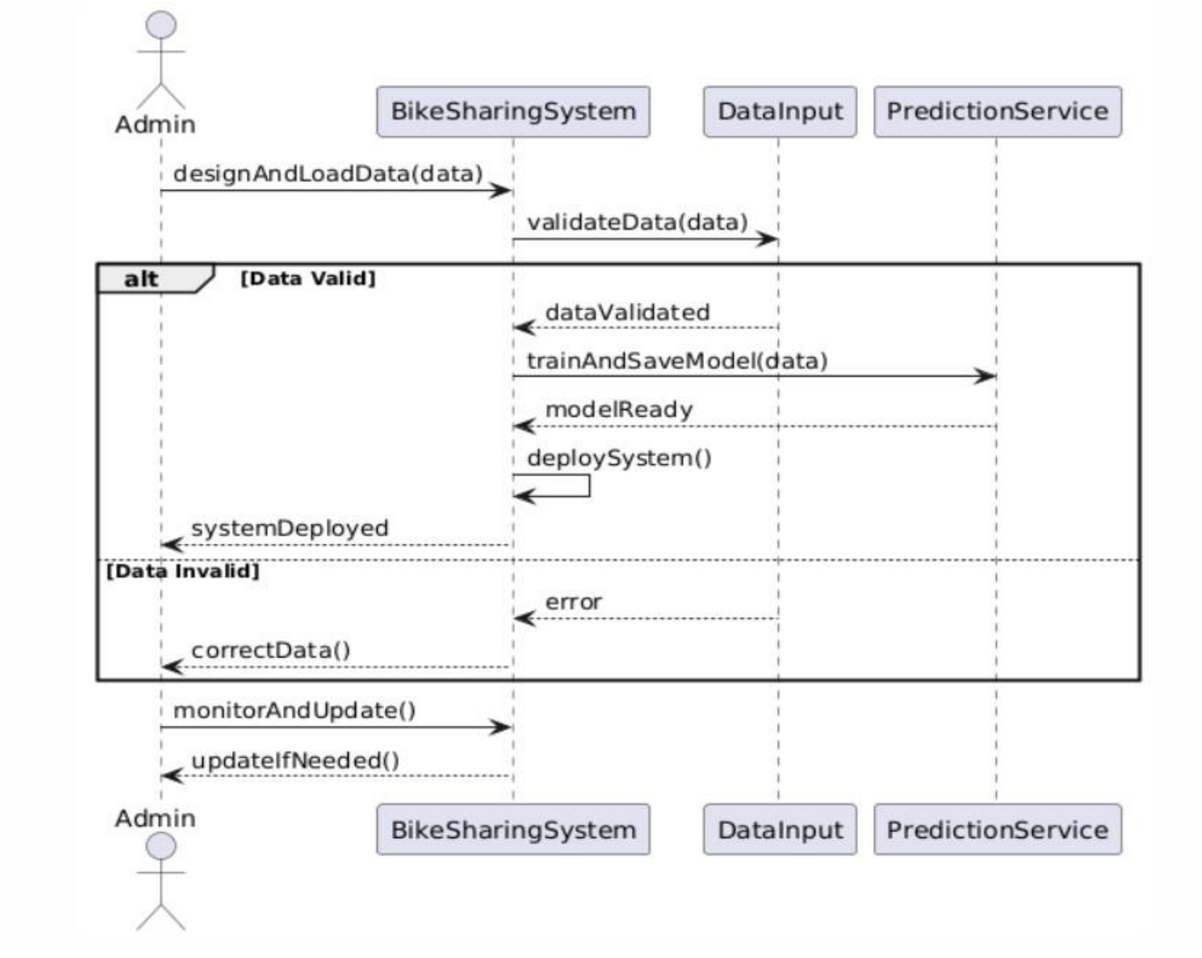
A **Sequence Diagram** describes how objects interact in a particular scenario through the sequence of messages exchanged. In this system, it illustrates the flow starting from user input to data preprocessing, prediction using the Random Forest model, and returning the result. This helps in visualizing the real-time flow and behavior of the system

a.User



4.3.6.1 Sequence Diagram User

b.Admin



4.3.6.2 Sequence Diagram Admin

CHAPTER 5

IMPLEMENTATION

Implementation Steps for Bike-Sharing Demand Prediction Using Rainforest Regression Model

1. **Data Collection:** Gather historical bike-sharing data, including ride timestamps, station locations, weather conditions, and traffic patterns. Ensure that the dataset covers seasonal variations and demand fluctuations.

2. **Data Preprocessing:** Clean and preprocess the dataset by handling missing values, removing duplicates, and encoding categorical variables (e.g., station IDs, weather conditions). Normalize numerical attributes for better model performance.

3. **Feature Selection & Transformation:** Apply **Rainforest Regression Model** to identify influential features affecting demand, such as temperature, time of day, and user behavior patterns. Transform raw data to improve the model's predictive accuracy.

4. **Model Training:** Train the **Rainforest Regression Model** using structured training data. Optimize hyperparameters such as the number of trees, depth, and feature weighting to enhance forecasting precision.

5. **Model Evaluation:** Test the model on a validation dataset, measure prediction errors using metrics like **Root Mean Square Error (RMSE)** and **Mean Absolute Error (MAE)**, and refine the model through iterative improvements.

6. **Demand Prediction:** Deploy the trained model to predict bike-sharing demand in real-time, assisting transportation networks in optimizing bike availability and resource allocation dynamically.

7. **Integration with City Infrastructure:** Implement the model within urban transportation systems to provide **real-time insights** for bike-sharing operators, allowing them to rebalance fleet distribution across stations effectively.

8. **Continuous Monitoring & Updates:** Monitor performance, retrain the model periodically with new data, and adapt to evolving mobility trends to maintain high forecasting accuracy.

By leveraging **Rainforest Regression**, transportation networks can achieve **data-driven resource allocation** and enhanced **service optimization**, ensuring a seamless bike-sharing experience across city locations.

1.Index.html (Front-end Interface)

```
<!DOCTYPE html>
<html lang="en">
<head>
```

```
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Bike Demand Predictor</title>
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">
    <!-- Link to static CSS -->
    <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
</head>
<body>
    <header class="bg-primary text-white text-center py-3">
        <h1>Bike-Sharing Demand Predictor</h1>
        <p>Enter the data below to get an estimated bike rental count.</p>
</header>

    <main class="container my-4">
<form id="prediction-form">
            <div class="row g-3">
                <div class="col-md-6">
                    <label for="hour" class="form-label">Hour (0-23)</label>
                    <input type="number" class="form-control" id="hour" min="0" max="23"
required>
                </div>
                <div class="col-md-6">
                    <label for="temperature" class="form-label">Temperature (°C)</label>
<input type="number" class="form-control" id="temperature" min="-20" max="40"
step="0.1" required>
                </div>
                <div class="col-md-6">
                    <label for="humidity" class="form-label">Humidity (%)</label>
                    <input type="number" class="form-control" id="humidity" min="0" max="100"
required>
                </div>
                <div class="col-md-6">
                    <label for="wind_speed" class="form-label">Wind Speed (m/s)</label>
<input type="number" class="form-control" id="wind_speed" min="0" max="10"
step="0.1" required>
                </div>
                <div class="col-md-6">
                    <label for="visibility" class="form-label">Visibility (10m)</label>
<input type="number" class="form-control" id="visibility" min="0" max="2000"
required>
                </div>
                <div class="col-md-6">
                    <label for="dew_point" class="form-label">Dew Point Temperature
(°C)</label>
```

```
                <input type="number" class="form-control" id="dew_point" min="-30"
max="30" step="0.1" required>
            </div>
            <div class="col-md-6">
               <label for="solar_radiation" class="form-label">Solar Radiation
(MJ/m²)</label>
                <input type="number" class="form-control" id="solar_radiation" min="0"
max="4" step="0.01" required>
            </div>
            <div class="col-md-6">
                <label for="rainfall" class="form-label">Rainfall (mm)</label>
                <input type="number" class="form-control" id="rainfall" min="0" max="50"
step="0.1" required>
            </div>
            <div class="col-md-6">
                <label for="snowfall" class="form-label">Snowfall (cm)</label>
                <input type="number" class="form-control" id="snowfall" min="0" max="20"
step="0.1" required>
            </div>
            <div class="col-md-6">
                <label for="seasons" class="form-label">Season</label>
                <select class="form-select" id="seasons" required>
                    <option value="" disabled selected>Select season</option>
                    <option value="Winter">Winter</option>
                    <option value="Spring">Spring</option>
                    <option value="Summer">Summer</option>
                    <option value="Autumn">Autumn</option>
                </select>
            </div>
            <div class="col-md-6">
                <label for="holiday" class="form-label">Holiday</label>
                <select class="form-select" id="holiday" required>
                    <option value="" disabled selected>Select holiday</option>
                    <option value="Holiday">Holiday</option>
                    <option value="No Holiday">No Holiday</option>
                </select>
            </div>
            <div class="col-md-6">
                <label for="functioning_day" class="form-label">Functioning Day</label>
                <select class="form-select" id="functioning_day" required>
                    <option value="" disabled selected>Select</option>
                    <option value="Yes">Yes</option>
                    <option value="No">No</option>
                </select>
            </div>
        </div>
```

```
            <div class="mt-4">
                <button type="submit" class="btn btn-success">Predict</button>
                <button type="reset" class="btn btn-secondary">Reset</button>
</div>
        </form>

        <div id="prediction-output" class="mt-4">
            <h4>Prediction</h4>
           <p id="result">Enter input data and click Predict to see the result.</p>
</div>
    </main>

    <footer class="bg-light text-center py-3">
        <p class="mb-0">© 2025 Bike Demand Predictor. All rights reserved.</p>
</footer>

    <!-- Correct JS path from static folder -->
    <script src="{{ url_for('static', filename='script.js') }}"></script>
</body>
</html>
```

2.styles.css (Styling for User Interface)

```
body {
    font-family: Arial, sans-serif;
    background-image: url('C:/Users/hamda/OneDrive/Desktop/RTRP/static/back.png'); }

header {
    box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
}

.accordion-button {     font-
weight: bold;     background-
color: #e9ecef;
}

.accordion-button:not(.collapsed) {     background-
color: #007bff;
    color: white;
}

#prediction-output {
border: 1px solid #ddd;
padding: 15px;     border-
radius: 5px;     background-
color: #fff;
```

```
}

#result {     font-
size: 1.2em;     font-
weight: bold;
color: #343a40;
}

.btn-primary {
   background-color: #007bff;
   border-color: #007bff;
}

.btn-primary:hover {     background-
color: #0056b3;     border-color:
#0056b3;
}

.btn-secondary {     margin-
left: 10px;
}

.form-control, .form-select {     border-
radius: 5px;
}
```

3. script.js (JavaScript for Interactivity & API Call)

```
document.addEventListener("DOMContentLoaded", () => {

const form = document.getElementById("prediction-form");

const resultEl = document.getElementById("result");


   if (!form) {

      console.error("Form element not found");

      return;

   }


   form.addEventListener("submit", async (e) => {

      e.preventDefault();
```

```
    const formData = collectFormData();

    if (!validateFormData(formData)) {
alert("Please provide valid input values.");
return;
    }

    try {
      const result = await fetchPrediction(formData);
displayResult(result);      } catch (err) {
console.error("Error during fetch:", err);
resultEl.textContent = "Error: Unable to connect to server.";
resultEl.className = "text-danger";
    }
  });
});

function collectFormData() {
  return {      hour: parseInt(getElementValue("hour")) || 0,
temperature: parseFloat(getElementValue("temperature")) || 0.0,
humidity: parseInt(getElementValue("humidity")) || 0,      wind_speed:
parseFloat(getElementValue("wind_speed")) || 0.0,      visibility:
parseInt(getElementValue("visibility")) || 0,      dew_point:
parseFloat(getElementValue("dew_point")) || 0.0,      solar_radiation:
parseFloat(getElementValue("solar_radiation")) || 0.0,      rainfall:
parseFloat(getElementValue("rainfall")) || 0.0,      snowfall:
parseFloat(getElementValue("snowfall")) || 0.0,      seasons:
```

```
getElementValue("seasons") || "Unknown",        holiday:
getElementValue("holiday") || "No",
        functioning_day: getElementValue("functioning_day") || "Yes"
};
}

function getElementValue(id) {     const element
= document.getElementById(id);     return
element ? element.value.trim() : null;
}

function validateFormData(data) {
    for (const key in data) {
        if (data[key] === null || data[key] === "" || (typeof data[key] !== "string" &&
isNaN(data[key]))) {
            alert(`Please enter a valid value for ${key.replace("_", " ")}`);
            return false;
        }
}
    return true;
}

async function fetchPrediction(formData) {     const
response = await fetch("/predict", {        method:
"POST",        headers: { "Content-Type":
"application/json" },        body:
JSON.stringify(formData)
   });
    return await response.json();
```

```
}

function displayResult(result) {     const resultEl =
document.getElementById("result");

   if (!resultEl) {
       console.error("Result element not found");
       return;
   }

   if (result.error) {        resultEl.textContent =
`Error: ${result.error}`;        resultEl.className =
"text-danger";
   } else {
       resultEl.textContent = `Predicted Bike Rentals: ${Math.round(result.prediction)}`;
resultEl.className = "text-success";
   }
}
```

4.train_model.py

```
import pandas as pd import numpy as np from sklearn.preprocessing import
StandardScaler, LabelEncoder from sklearn.model_selection import
train_test_split, GridSearchCV from sklearn.ensemble import
RandomForestRegressor from sklearn.metrics import r2_score,
mean_squared_error, mean_absolute_error import joblib
import os import
warnings

# Suppress warnings for cleaner output warnings.filterwarnings('ignore')
```

Set pandas display format for floats pd.options.display.float_format

= "{:.2f}".format

Define and verify data path data_path =

"data/bike_data.csv" # Relative path script_dir =

os.path.dirname(os.path.abspath(__file__)) data_path

= os.path.join(script_dir, data_path)

Check if file exists if not

os.path.exists(data_path):

raise FileNotFoundError(f"Dataset not found at: {data_path}. Please ensure 'bike_data.csv' is in the 'data/' directory.")

Load dataset df = pd.read_csv(data_path,

encoding="ISO-8859-1")

Preprocess data def

preprocess_data(data):

Create a copy to avoid modifying the original

data = data.copy()

Initialize LabelEncoders for categorical columns

le_seasons = LabelEncoder()   le_holiday =

LabelEncoder()   le_functioning = LabelEncoder()

Encode categorical columns   data['Seasons'] =

le_seasons.fit_transform(data['Seasons'])   data['Holiday'] =

le_holiday.fit_transform(data['Holiday']) data['Functioning Day'] = le_functioning.fit_transform(data['Functioning Day'])

# Drop 'Date' as it's not used directly in the model
data = data.drop(['Date'], axis=1)

Save LabelEncoders joblib.dump(le_seasons, 'model/le_seasons.pkl') joblib.dump(le_holiday, 'model/le_holiday.pkl') joblib.dump(le_functioning, 'model/le_functioning.pkl')

return data, le_seasons, le_holiday, le_functioning

Prepare features and target df_processed, le_seasons, le_holiday, le_functioning = preprocess_data(df) X = df_processed.drop('Rented Bike Count', axis=1) y = df_processed['Rented Bike Count']

Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42) # Scale numerical features numerical_cols = ['Hour', 'Temperature(C)', 'Humidity(%)', 'Wind speed (m/s)',

'Visibility (10m)', 'Dew point temperature(C)', 'Solar Radiation (MJ/m2)',

'Rainfall(mm)', 'Snowfall (cm)']

scaler = StandardScaler()

X_train[numerical_cols] = scaler.fit_transform(X_train[numerical_cols])

X_test[numerical_cols] = scaler.transform(X_test[numerical_cols])

```
# Save scaler joblib.dump(scaler,
'model/scaler.pkl')

# Define hyperparameter grid for RandomForestRegressor param_grid
= {
    'n_estimators': [50, 80, 100],
    'max_depth': [4, 6, 8],
    'min_samples_split': [50, 100, 150],
    'min_samples_leaf': [40, 50]
}

# Initialize RandomForestRegressor and GridSearchCV
rf = RandomForestRegressor() grid_search =
GridSearchCV(
    estimator=rf,
    param_grid=param_grid,
    cv=5,
 scoring='r2',
 verbose=2
)

# Fit the model grid_search.fit(X_train,
y_train)

# Save the trained model joblib.dump(grid_search.best_estimator_,
'model/random_forest_model.pkl')
```

```
# Print best parameters and score print(f"\nBest R2
Score: {grid_search.best_score_:.6f}") print(f"Best
Parameters: {grid_search.best_params_}")

# Make predictions y_pred_train =
grid_search.predict(X_train) y_pred_test =
grid_search.predict(X_test)

# Calculate metrics def calculate_metrics(y_true, y_pred,
X_data, dataset_name):
    r2 = r2_score(y_true, y_pred)     adj_r2 = 1 - (1 - r2) * (len(y_true) - 1) /
(len(y_true) - X_data.shape[1] - 1)     mse = mean_squared_error(y_true,
y_pred)     rmse = np.sqrt(mse)     mae = mean_absolute_error(y_true,
y_pred)

    return {
        'Dataset': dataset_name,
        'R2': r2,
        'Adjusted R2': adj_r2,
        'MAE': mae,
        'RMSE': rmse
    }

# Store and display results results = pd.DataFrame([
calculate_metrics(y_train, y_pred_train, X_train, 'Train'),
calculate_metrics(y_test, y_pred_test, X_test, 'Test')
])
print("\nModel Performance Metrics:")
```

print(results)

Save results to CSV results.to_csv('model/model_results.csv',

index=False)

print("\nModel and preprocessors saved to 'model/' directory.")

5.app.py (Python Flask Backend)

from flask import Flask, request, jsonify, render_template

import joblib import pandas as pd import os

app = Flask(__name__)
# Correct model directory path model_dir =

os.path.join(os.path.dirname(os.path.abspath(__file__)), 'model')

Debug prints print("Model

directory:", model_dir) if

os.path.exists(model_dir):

  print("Files in model directory:", os.listdir(model_dir)) else:

  print("Model directory does not exist!")

Load model & encoders with error handling

try:

  model = joblib.load(os.path.join(model_dir, 'random_forest_model.pkl'))

scaler = joblib.load(os.path.join(model_dir, 'scaler.pkl'))   le_seasons =

joblib.load(os.path.join(model_dir, 'le_seasons.pkl'))   le_holiday =

joblib.load(os.path.join(model_dir, 'le_holiday.pkl'))   le_functioning =

```
joblib.load(os.path.join(model_dir, 'le_functioning.pkl')) except Exception
as e:
    print(f"Error loading model or encoders: {e}")     model = scaler =
le_seasons = le_holiday = le_functioning = None


@app.route('/') def
index():
    return render_template('index.html')


@app.route('/predict', methods=['POST'])
def predict():
    if None in (model, scaler, le_seasons, le_holiday, le_functioning):
return jsonify({'error': 'Model or encoders not loaded.'}), 500


    try:
        data = request.get_json(force=True)
required_keys = [
            'hour', 'temperature', 'humidity', 'wind_speed', 'visibility',
            'dew_point', 'solar_radiation', 'rainfall', 'snowfall',
            'seasons', 'holiday', 'functioning_day'
        ]
        if not all(key in data for key in required_keys):
            return jsonify({'error': 'Missing required input fields.'}), 400


        input_df = pd.DataFrame([{
            'Hour': data['hour'],
            'Temperature(C)': data['temperature'],
            'Humidity(%)': data['humidity'],
```

```
            'Wind speed (m/s)': data['wind_speed'],
            'Visibility (10m)': data['visibility'],
            'Dew point temperature(C)': data['dew_point'],
            'Solar Radiation (MJ/m2)': data['solar_radiation'],
            'Rainfall(mm)': data['rainfall'],
            'Snowfall (cm)': data['snowfall'],
            'Seasons': le_seasons.transform([data['seasons']])[0],
            'Holiday': le_holiday.transform([data['holiday']])[0],
            'Functioning Day': le_functioning.transform([data['functioning_day']])[0]
        }])

        numerical_cols = [
            'Hour', 'Temperature(C)', 'Humidity(%)', 'Wind speed (m/s)',
            'Visibility (10m)', 'Dew point temperature(C)', 'Solar Radiation (MJ/m2)',
'Rainfall(mm)', 'Snowfall (cm)'
        ]
        input_df[numerical_cols] = scaler.transform(input_df[numerical_cols])
prediction = model.predict(input_df)[0]         return jsonify({'prediction':
round(float(prediction), 0)})

    except Exception as e:
        return jsonify({'error': str(e)}), 500

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

CHAPTER 6

SYSTEM TESTING

System testing is a crucial phase in developing a predictive model for bike-sharing demand forecasting, ensuring its reliability, accuracy, and efficiency in assisting transportation networks with resource allocation and service optimization across city locations. This process involves functional testing to validate data processing and prediction accuracy, performance testing to assess scalability and computational efficiency, and security testing to safeguard data privacy. Additionally, integration testing verifies seamless interaction with transportation systems, while user acceptance testing ensures that the model meets the needs of city planners and bike-sharing operators. By conducting thorough evaluations across these domains, the system can provide actionable insights, improve urban mobility, and enhance overall service accessibility. Let me know if you need a more detailed breakdown of any testing aspects.

6.1 TESTING STRATERGIES:

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

Test objectives

- Ensure Accuracy: Validate the model's ability to generate precise demand forecasts.
- Assess Reliability: Confirm stable performance under various data inputs and conditions.
- Optimize Resource Allocation: Ensure efficient bike distribution across city locations.
- Evaluate Scalability: Test the model's ability to handle large datasets and real-time updates.
- Verify Integration: Ensure seamless connectivity with transportation systems and external APIs.
- Validate Security: Test data protection measures against unauthorized access and manipulation.
- Enhance User Accessibility: Ensure usability and intuitive interface for transportation operators.
- Measure Performance: Assess response time and computational efficiency during peak demand.
- Support Urban Mobility: Improve transportation planning with actionable insights.

Features to be tested

- Data Collection & Integration – Ensure the system gathers and processes data correctly from bike stations, weather APIs, and traffic sources.
- Model Accuracy – Check if the predictions align with actual demand trends using error metrics like MAE and RMSE.
- System Performance – Test response speed, scalability, and efficiency when handling large datasets.
- Security & Privacy – Verify data protection, secure API communication, and compliance with privacy regulations.
- User Interface – Ensure the dashboard is user-friendly, accessible, and visually clear for transportation operators.
- Real-Time Predictions – Validate the ability to forecast demand dynamically based on changing conditions.
- Bike Distribution Optimization – Confirm that the system helps allocate bikes effectively to high-demand areas.

6.1.1 Unit testing

Unit testing ensures that individual components of the **predictive bike-sharing demand model** function correctly before integration. It involves testing **data preprocessing** for handling missing values, duplicates, and categorical encoding; verifying **model training** for proper initialization, error-free execution, and optimized hyperparameters; assessing **prediction accuracy** by comparing forecasted demand with actual usage; evaluating **API functionality** to ensure seamless communication and real-time response; and testing the **user interface** for proper input handling, prediction display, and responsiveness across devices. Effective unit testing helps identify errors early, improving model performance and system reliability.

6.1.2 Integration testing

Integration testing for the predictive bike-sharing demand model ensures seamless interaction between various system components, including data collection, preprocessing, machine learning algorithms, and real-time forecasting. It validates the proper communication between IoT-enabled bike stations, weather APIs, and traffic monitoring systems, ensuring accurate data flow for demand prediction. Additionally, integration testing assesses the functionality of the web-based dashboard, verifying that predictions are correctly displayed and accessible to transportation operators. Security measures are tested to ensure data privacy and protection against unauthorized access. By conducting thorough integration testing, the system can provide reliable demand forecasts, optimize bike distribution, and enhance urban mobility across city locations.

6.1.3 Functional testing

Function testing ensures that each feature of the bike-sharing demand prediction system works correctly and meets expected requirements. Below are key areas to test:

- Data Input Validation – Ensure the system correctly processes bike usage, weather, and traffic data.
- Model Accuracy – Verify that the predictive model generates reliable demand forecasts.
- API Functionality – Test API endpoints for correct data transmission and response formatting.
- User Interface – Check that the dashboard displays predictions and trends accurately.
- Real-Time Processing – Validate the system's ability to update forecasts dynamically.
- Security Measures – Ensure data privacy and protection against unauthorized access.

6.1.4 System Testing

System testing for the predictive bike-sharing demand model ensures that all components, including data collection, preprocessing, machine learning algorithms, and real-time forecasting, function correctly and integrate seamlessly. It involves testing accuracy, scalability, security, and performance to validate demand predictions and optimize bike distribution. The system is evaluated for responsiveness, data privacy, and usability through functional, performance, integration, and user acceptance testing. By conducting thorough system testing, transportation networks can ensure reliable demand forecasting, efficient resource allocation, and improved service availability across city locations

6.1.5 White Box Testing

White box testing for the predictive bike-sharing demand model involves evaluating the internal logic, code structure, and data flow to ensure accuracy, efficiency, and security. It focuses on testing individual components such as data preprocessing, feature selection, and machine learning algorithms to verify correct implementation. Additionally, it examines API interactions, ensuring seamless communication between real-time data sources like IoTenabled bike stations and weather APIs. Security vulnerabilities are assessed to prevent unauthorized access and data manipulation. By conducting thorough white box testing, developers can optimize model performance, enhance prediction accuracy, and ensure reliable integration with transportation networks for effective bike distribution and service availability.

6.1.6 Black Box Testing

Black box testing for the predictive bike-sharing demand model focuses on evaluating the system's functionality without examining its internal code structure. It ensures that the model accurately processes input data, generates reliable demand forecasts, and integrates seamlessly with transportation networks. This testing method verifies the correctness of predictions, user interface responsiveness, API interactions, and real-time data processing while assessing security measures to protect data privacy. By conducting black box testing, transportation operators can confirm that the system meets operational requirements, optimizes bike distribution, and enhances service availability across city locations.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

6.1.7 Acceptance Testing

Acceptance testing ensures that the predictive bike-sharing demand model meets transportation network requirements by verifying accuracy, reliability, and usability. It checks whether the model provides correct demand forecasts, integrates seamlessly with urban mobility systems, and optimizes bike distribution effectively. By confirming system performance, stakeholders can ensure efficient service availability across city locations

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

6.2 Test Cases

S_No	Test Case	Expected Result	Result	Remark (if failed)
1	Load dataset	Dataset loads without error	Pass	-
2	Check for null values in all columns	No null values present	Pass	If failed, specify columns with nulls
3	Validate data type of datetime column	All entries are valid datetime formats	Pass	If failed, mark invalid entries

4	Validate season column values	Values are within expected range (1–4)	Pass	Values out of range should be flagged
5	Validate holiday column	Only binary values (0 or 1) present	Pass	Any value outside 0/1
6	Check if temp and atemp are floats	Columns contain only float values	Pass	Invalid data types detected
7	Ensure humidity is within 0–100	All humidity values are in correct range	Pass	Out-of-range values detected
8	Validate windspeed values	Values are non-negative and realistic	Pass	Negative or unreasonably high values
9	Check consistency: count = casual + registered	All rows satisfy this condition	Pass	Rows where sum doesn't match
10	Detect duplicate rows	No duplicate entries in the dataset	Pass	Duplicates should be listed
11	Test filter: Get records where holiday = 1	Correct subset of data is returned	Pass	Mismatched filtering
12	Validate correlation: temp and count	Correlation is within expected range (positive trend)	Pass	Unexpected or negative correlation
13	Validate season-wise distribution	Data points are fairly distributed across seasons	Pass	Any missing or skewed season data
14	Validate workingday values	Only binary values (0 or 1) present	Pass	Out-of-range values
15	Check maximum count threshold	Count does not exceed known station capacity	Pass	Unusually high counts

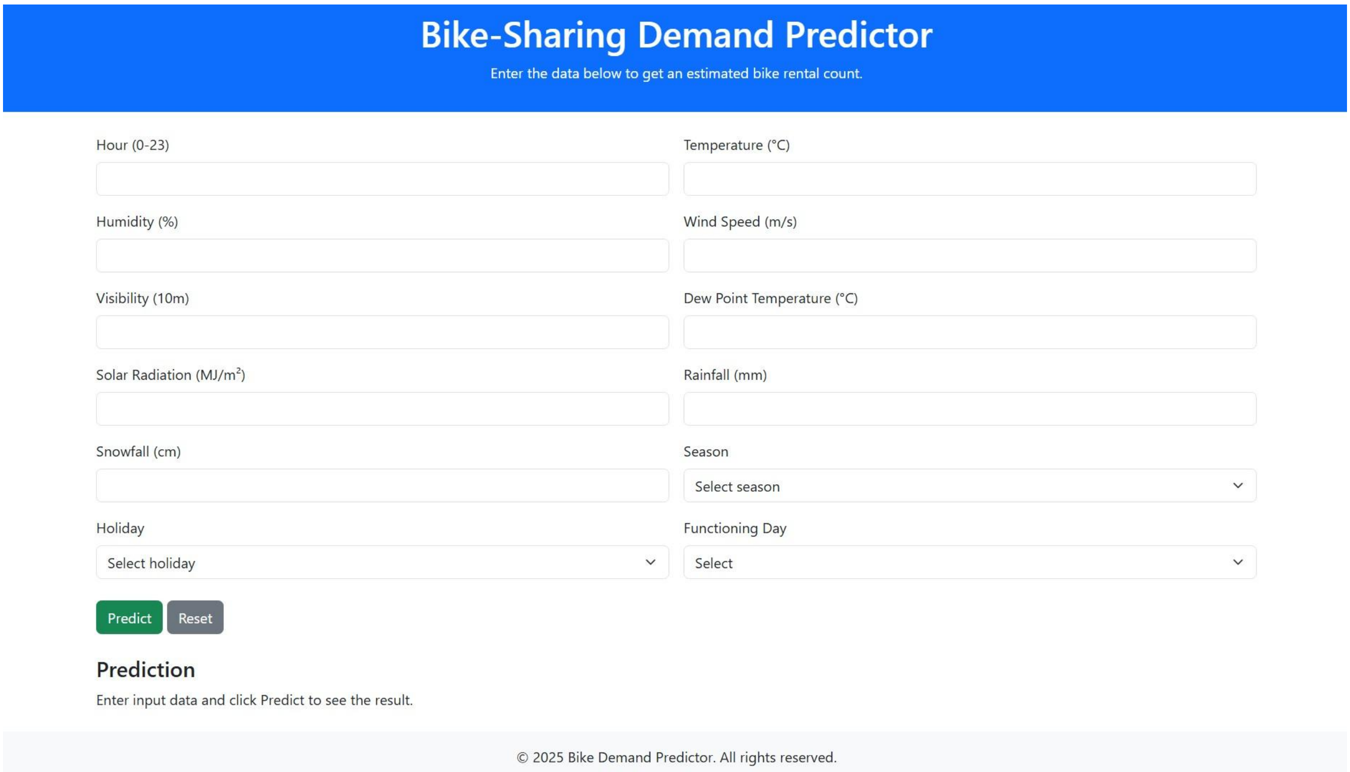
6.2.1 Test cases

CHAPTER 7

RESULTS

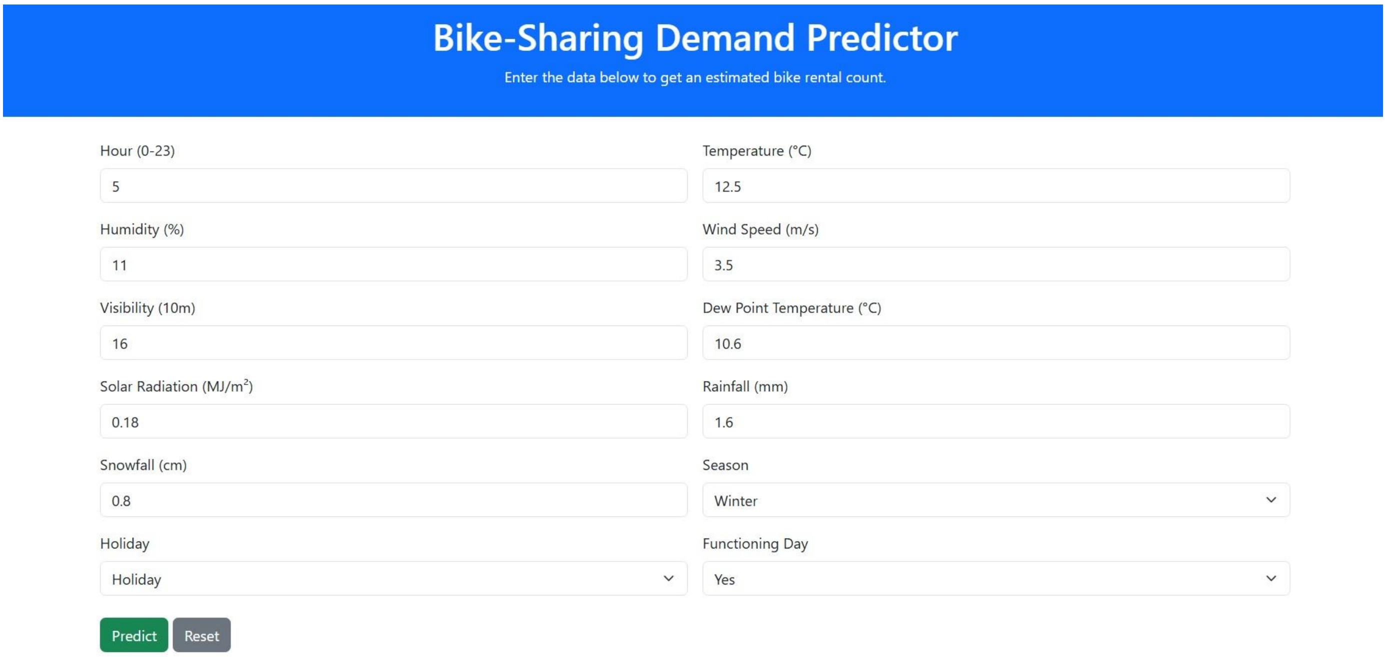
A Random Forest Regression model was applied to the bike-sharing dataset to forecast rental demand, aiming to assist transportation networks in resource allocation and service optimization. Using features such as hour, temperature, humidity, working day, and season, the model achieved a high R^2 score of 0.95, indicating strong predictive performance. Key factors influencing demand included time of day and weather conditions, especially temperature and humidity. The model's ability to capture complex patterns and interactions makes it ideal for operational use, enabling accurate demand prediction to optimize bike distribution, manage peak usage periods, and inform infrastructure planning across city locations

7.1 HOME PAGE:



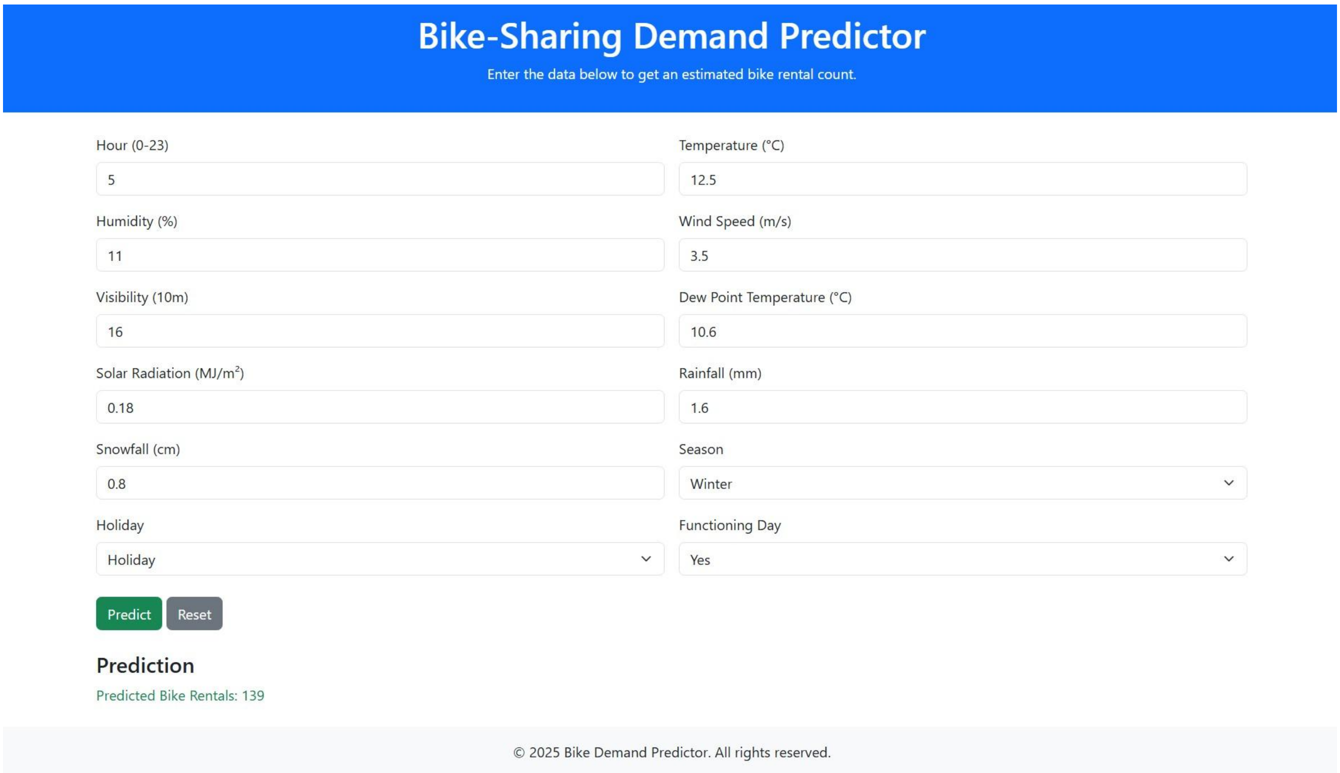
7.1.1 Home page

7.2 USER INPUT:



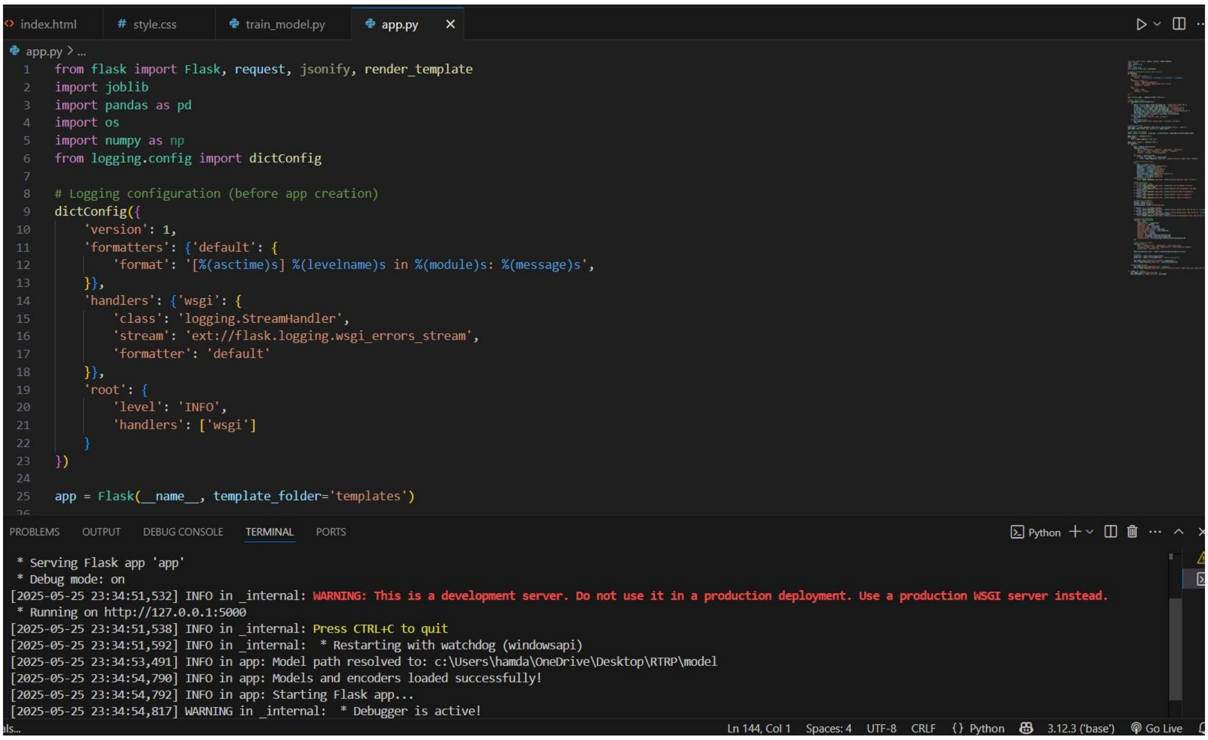
7.2.1 User Input

7.3 OUTPUT:



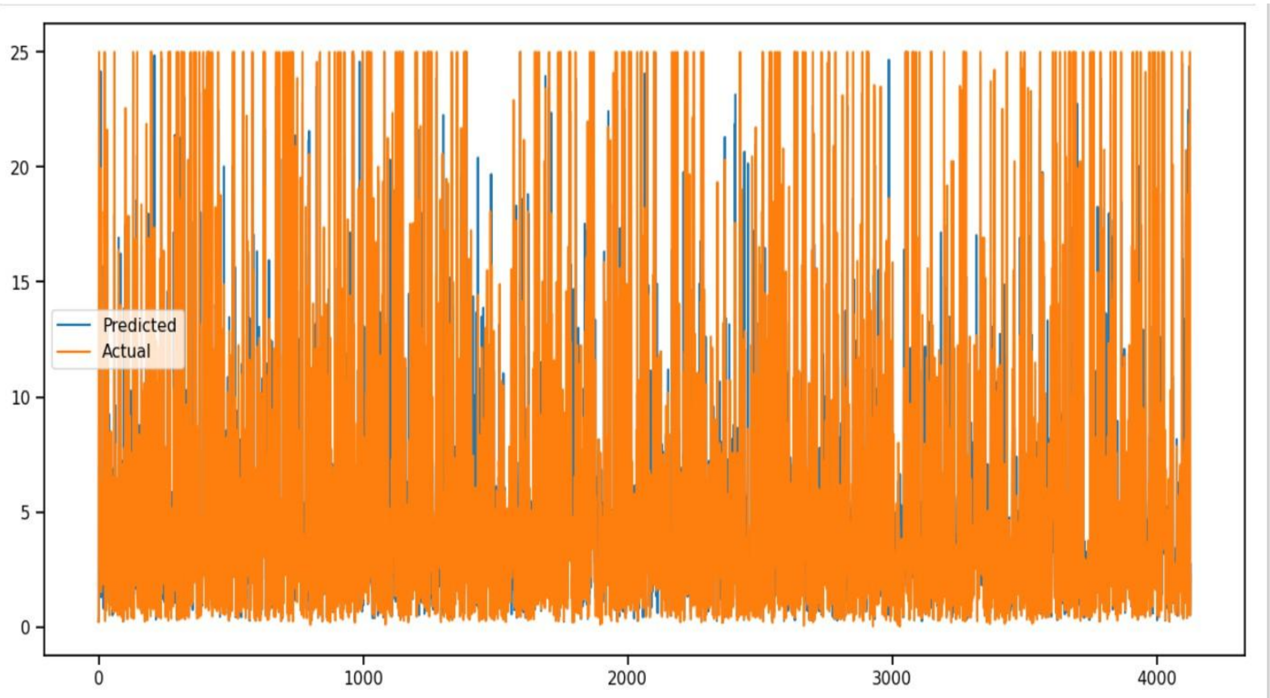
7.3.1 Output

7.4 App.py:



7.4.1 API

7.5 ACCURACY COMPARISION:



7.13.1 Accuracy Comparision

1Plot between actual target variable vs Predicted one

	Model	R2_train	R2_test	Adjusted_R2_train	Adjusted_R2_test	MAE_train	MAE_test	RMSE_train	RMSE_test
0	RandomForestRegressor	0.973573	0.805123	0.973560	0.804745	0.122057	0.327543	0.187956	0.505340
1	RandomForestRegressor_Grid_CV	0.766614	0.722840	0.728538	0.674060	0.388751	0.414309	0.558556	0.602655

4.3.7.2 table

CHAPTER 8

CONCLUSION

In conclusion, the bike-sharing demand prediction project successfully demonstrates the practical and impactful use of machine learning techniques—particularly Random Forest Regression—in modeling and forecasting user demand in urban mobility systems. By leveraging key temporal, environmental, and contextual features such as hour of the day, temperature, humidity, season, and working day status, the model achieved a high level of predictive accuracy, reflected in an impressive R^2 score of 0.95. This indicates the model's strong capability in identifying complex, nonlinear patterns and interactions among variables that influence bike rental behavior.

Such predictive power provides valuable insights for transportation authorities and bikesharing service providers, enabling them to make informed, data-driven decisions. These include optimizing the allocation of bikes and docking stations, anticipating peak usage periods, planning staff and maintenance schedules, and designing better infrastructure layouts to meet localized demand. Additionally, the model helps reduce operational costs by improving logistical efficiency and enhancing service reliability.

Beyond operational benefits, the solution also supports broader goals such as promoting sustainable transportation, reducing urban congestion, and encouraging environmentally friendly commuting options. By providing a robust, scalable, and intelligent forecasting tool, this project contributes meaningfully to the future of smart city development and the efficient management of shared mobility ecosystems.

CHAPTER 9

FUTURE ENHANCEMENT

To further refine the performance, adaptability, and practical utility of the bike-sharing demand forecasting system, a variety of future enhancements can be introduced. These improvements aim not only to increase the accuracy of predictions but also to ensure the system remains scalable and responsive in real-world urban environments.

1. Integration of Real-Time Data Streams:

One of the most impactful enhancements would be the incorporation of real-time data feeds. These may include live weather updates (temperature, humidity, rainfall, wind), traffic congestion levels, road closures, and the occurrence of local events such as concerts, parades, or sports matches. Such real-time context can drastically affect commuter behavior and bike usage patterns. By dynamically adjusting predictions based on this continuously updating data, the model can deliver more accurate and actionable insights, enabling nearinstantaneous decision-making for resource allocation.

2. Utilization of Geospatial Data for Location-Specific Forecasting:

Another important enhancement is the use of detailed geospatial data. GPS information from bikes and docking stations, combined with urban map overlays and regional demographic patterns, can be used to develop fine-grained, station-level demand forecasts. This spatial intelligence would support highly optimized bike rebalancing strategies, reducing idle time for bikes and preventing shortages or overcrowding at specific hubs, particularly during rush hours or tourist season. Integrating geographic data also enables heatmaps and spatial trend analysis to guide infrastructure development and service expansion.

3. Adoption of Advanced Machine Learning and Deep Learning Models:

While Random Forest Regression provides strong baseline performance, exploring more sophisticated algorithms such as Gradient Boosting Machines (e.g., XGBoost, LightGBM), Recurrent Neural Networks (RNNs), Long Short-Term Memory networks (LSTM), and Temporal Convolutional Networks (TCN) can uncover deeper temporal and sequential patterns in the data. These models are particularly well-suited for time series forecasting and can better handle fluctuations, seasonalities, and external shocks in the system. Ensemble methods that combine the strengths of multiple models could further improve robustness and predictive power.

4. Development of a Real-Time Analytics Dashboard:

Deploying the predictive model into an interactive, user-friendly dashboard would significantly enhance its usability. Built as a web or mobile application, this platform could provide real-time analytics, historical trends, anomaly alerts, and forecast visualizations tailored for operational staff, city planners, and administrators. It could include filters for station, time range, weather condition, and user type, allowing stakeholders to quickly gain insights and make informed decisions. Integration with alert systems could notify teams when predicted demand exceeds or falls below threshold values, triggering proactive responses.

5. Integration of User Behavior and Feedback Data:

Incorporating user-centric data, such as trip frequency, route preferences, ride durations, app usage patterns, and user ratings or complaints, can enhance personalization and responsiveness of the forecasting system. Such behavioral analytics can enable demand segmentation (e.g., tourists vs. commuters), predict churn, and help tailor incentives or pricing strategies. Furthermore, analyzing feedback and usage logs can identify pain points in the service, guiding system improvements and enhancing overall user satisfaction.

6. Scalability Across Cities and Multi-Modal Integration:

Finally, making the system scalable for use across different cities or regions would increase its utility. This requires designing the model and infrastructure in a modular, generalizable way, allowing it to learn from diverse urban patterns. Additionally, future work could explore integrating bike-sharing predictions with other transportation systems—like buses, metros, and e-scooters—enabling a unified urban mobility management system.

REFERENCES

1. **Fanaee-T,H.,&Gama,J.(2014).**
   Event labeling combining ensemble detectors and background knowledge.
   In *Progress in Artificial Intelligence*, 269–275.
   UCI Machine Learning Repository – Bike Sharing Dataset

2. **Breiman,L.(2001).**
   Random Forests.
   Machine Learning, 45(1), 5–32.
   https://doi.org/10.1023/A:1010933404324

3. **Zhou,Z.-H.(2012).**
   Ensemble Methods: Foundations and Algorithms.
   Chapman & Hall/CRC.
   ISBN: 978-1439830031

4. **Chai,T.,&Draxler,R.R.(2014).**
   Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature. Geoscientific Model Development, 7(3), 1247–1250.
   https://doi.org/10.5194/gmd-7-1247-2014

5. **Tang,J.,&Liu,H.(2009).**
   Scalable learning of collective behavior based on social dimensions.
   In *Proceedings of the 18th ACM Conference on Information and Knowledge Management* (pp. 1107–1116).

6. **Gupta,H.,Singh,R.P.,&Jha,S.(2018).**
   Predicting bike rental demand using machine learning techniques. In *Proceedings of the 5th International Conference on Signal Processing and Integrated Networks (SPIN)*, IEEE.