Bike Share Prediction

Revision Number: 1.0
Last date of revision: 19/07/2023

Document Version Control

Date Issued	Version	Description	Author
19/07/2023	1	Initial HLD — V1.0	Prince Vaibhav

Contents

Abstract

Bike sharing systems are a new generation of traditional bike rentals where the whole process from membership, rental and return has become automatic. Through these systems, users can easily rent a bike from a particular position and return back to another position. Currently, there are over 500 bike-sharing programs around the world which are composed of over 500 thousand bicycles. Today, there exists great interest in these systems due to their important role in traffic, environmental and health issues. Apart from interesting real-world applications of bike sharing systems, the characteristics of data being generated by these systems make them attractive for the research. The goal here is to build an end-to-end regression task. Here the user will provide the data and the result will be given by the best performing hyper tuned Machine Learning model. The user will also get privileges to choose the deployment options.

1 Introduction

1.1 Why this High-Level Design Document?

The purpose of this High-Level Design (HLD) Document is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding and can be used as a reference manual for how the modules interact at a high level.

The HLD will:

- Present all of the design aspects and define them in detail
- Describe the user interface being implemented
- Describe the hardware and software interfaces
- Describe the performance requirements
- Include design features and the architecture of the project
- List and describe the non-functional attributes like:
    - Security
    - Reliability
    - Maintainability
    - Portability
    - Reusability
    - Application compatibility
    - Resource utilization
    - Serviceability

1.2 Scope

The HLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HLD uses non-technical to mildly technical terms which should be understandable to the administrators of the system.

1.3 Definitions

Term	*Description*
Database	Collection of all the information monitored by this system
IDE	Integrated Development Environment

2 General Description

2.1 Product Perspective

- The bike sharing system is a standalone product that integrates with existing transportation infrastructure and services. It provides an automated, convenient, and flexible solution for users to rent and return bikes from various positions. The system interacts with users through intuitive interfaces, integrates with mobile applications for seamless user experience, and generates valuable data for research and analysis.

2.2 Problem statement

- Traditional bike rentals lack automation, convenience, and flexibility. Users face challenges in finding available bikes and returning them to specific locations. Additionally, manual rental processes lead to inefficiencies and hinder scalability. There is a need for a bike sharing system that automates the entire process, enables easy rental and return, and addresses traffic, environmental, and health concerns.

2.3 Proposed Solution

- The proposed solution is a bike sharing system that offers an end-to-end automated process. It includes user registration, bike availability tracking, rental initiation, payment processing, bike unlocking, geolocation tracking, rental termination, and payment calculation. The system will have user-friendly interfaces, seamless integration with mobile applications, and robust backend architecture.

2.4 Further Improvements

- Integration with public transportation networks to offer a comprehensive multimodal transportation solution.
- Enhanced user experience through personalized recommendations, user feedback mechanisms, and loyalty programs.
- Integration with smart city infrastructure for real-time data exchange, traffic optimization, and parking availability information.
- Integration with bike maintenance systems to ensure bikes are in good condition and promptly serviced.
- Expansion to more cities and regions to provide broader accessibility and connectivity.
- Integration with alternative transportation modes such as electric scooters or electric bikes.

2.5 Technical Requirements

- Secure user authentication and data protection mechanisms.
- Real-time bike availability tracking and reservation system.
- Seamless integration with mobile applications for user registration, bike rental, and payment processing.
- Robust backend architecture for efficient data processing, storage, and retrieval.
- Geolocation tracking using GPS or network-based positioning systems.
- Integration with payment gateways for secure and convenient transactions.
- Scalable infrastructure to handle a growing user base and increasing bike fleet.
- Compatibility with multiple platforms and devices.

2.6 Data Requirements

- User profiles, including registration information, rental history, and preferences.
- Bike availability data, including bike location, status, and availability timeframes.
- Rental and return data, including timestamps, duration, and payment details.
- Geolocation data for bike positioning and user tracking.
- Operational data for system performance monitoring and optimization.
- Research data for analyzing usage patterns, traffic impact, and environmental benefits.

2.7 Tools used

Python programming language and frameworks such as NumPy, Pandas and Scikit-learn are used to build the whole model.

- PyCharm is used as IDE.
- For visualization of the plots, Matplotlib, Seaborn and Plotly are used.
- Front end development is done using HTML/CSS
- Machine learning model is trained by using scikit learn library.
- GitHub is used as version control system.

2.7.1 Hardware Requirements

- Bike Docking Stations:

Robust and weather-resistant design.
Sufficient docking capacity.
Reliable locking mechanisms.

- Bikes:

Durable and low-maintenance frames.
Integrated electronic components for identification, tracking, and locking.
GPS or similar positioning systems.
Secure and tamper-proof locking mechanisms.

- Communication Infrastructure:

Wireless connectivity for real-time data transmission.
Adequate coverage and signal strength.
Redundancy measures for system availability.

- Payment Terminals:

Reliable and secure payment processing.
Compatibility with various payment methods.

- Backend Infrastructure:

Servers or cloud-based infrastructure.
Adequate storage capacity.
Backup and disaster recovery mechanisms.

- Power Supply:

Reliable power sources.
Backup power solutions.
Energy-efficient components.

- Maintenance and Monitoring:

Tools and equipment for maintenance.
Monitoring systems for performance tracking.

- Compliance and Regulations:

Adherence to local safety and regulatory standards.
Consideration of accessibility requirements.

2.8 Constraints

- Budget constraints: Adherence to a specific budget for hardware components and system implementation.
- Regulatory compliance: Ensuring compliance with local regulations and standards for data privacy, safety, and accessibility.
- Technical compatibility: Hardware components should be compatible and seamlessly integrate with each other and the overall system infrastructure.

2.9 Assumptions

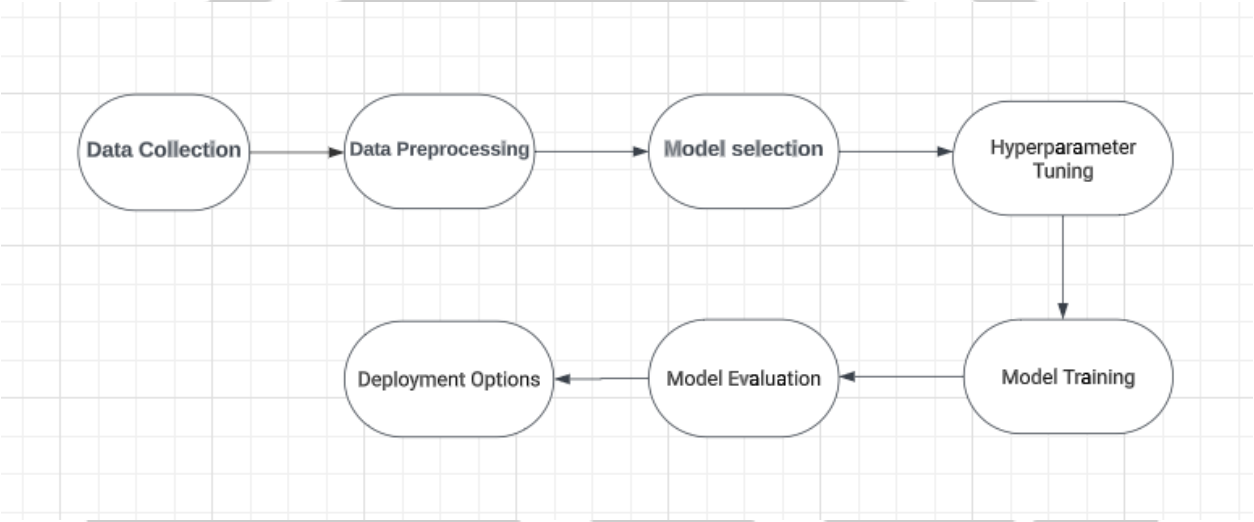
- Users will readily adopt the bike sharing system and actively engage in renting and returning bikes.
- Reliable internet connectivity will be available to users for seamless registration, transactions, and system updates.
- GPS or similar positioning systems will provide accurate bike tracking and user geolocation information.
- Regular maintenance and repairs will be conducted to ensure the proper functioning of bikes and docking stations.
- Appropriate measures will be implemented to protect user data and ensure compliance with privacy regulations.

3 Design Details

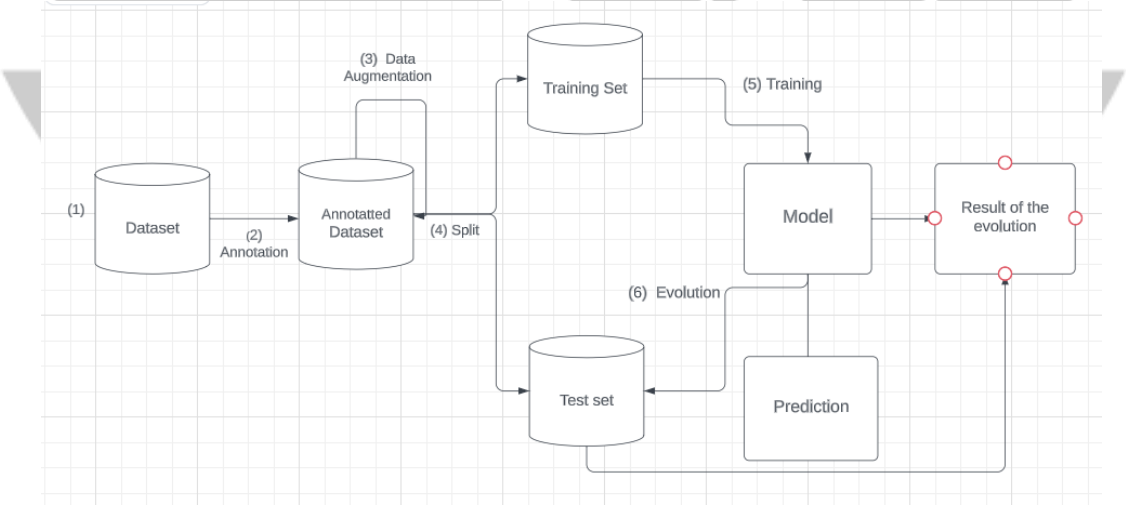
3.1 Process Flow

For identifying the different types of anomalies, we will use a deep learning base model. Below is the process flow diagram is as shown below.

Proposed methodology



3.1.1 Model Training and Evaluation



3.2 Event log

The system should log every event so that the user will know what process is running internally.

Initial Step-By-Step Description:

1. The System identifies at what step logging required
2. The System should be able to log each and every system flow.
3. Developer can choose logging method. You can choose database logging/ File logging as well.
4. System should not hang even after using so many loggings. Logging just because we can easily debug issues so logging is mandatory to do.

3.3 Error Handling

Should errors be encountered, an explanation will be displayed as to what went wrong? An error will be defined as anything that falls outside the normal and intended usage.

4 Performance

- Scalability: Design the system to handle a large number of users and bikes without performance issues.
- Response Time: Ensure fast response times for user interactions like registration, bike availability queries, and bookings.
- Throughput: Determine the desired number of transactions or requests the system can handle simultaneously.
- Availability: Aim for high uptime and reliability through redundancy and fault tolerance measures.
- Data Processing: Efficiently manage and process large volumes of real-time and historical data for accurate predictions.
- Performance Monitoring: Implement real-time monitoring to track key performance metrics.
- Performance Testing: Conduct load testing to simulate user scenarios and measure system response under stress.

4.1 Reusability

The code written and the components used should have the ability to be reused with no problems.

4.2 Application Compatibility

The different components for this project will be using Python as an interface between them. Each component will have its own task to perform, and it is the job of the Python to ensure proper transfer of information.

4.3 Resource Utilization

When any task is performed, it will likely use all the processing power available until that function is finished.

5 Conclusion

The bike share prediction system holds significant potential in enhancing the efficiency and user experience of bike sharing services. By leveraging data analysis and predictive modeling, the system can provide valuable insights into bike availability, demand patterns, and user behavior. This enables better resource allocation, optimized bike distribution, and improved overall service quality. With accurate predictions, users can make informed decisions, leading to increased customer satisfaction and enhanced operational performance. The bike share prediction system represents a valuable tool for both bike sharing service providers and users, fostering a more sustainable and convenient mode of transportation.

6 References

1. https://en.wikipedia.org/wiki/Unmanned_ground_vehicle
2. Google.com for images of UGV hardware.
3. https://www.ros.org/