BIKE SHARING DEMAND PREDICTION

High Level Design (HLD)

July 14, 2024

Rishabh Barman

Abstract

The bike share prediction system project aims to develop an accurate and adaptable machine learning model to forecast bike share demand in urban areas. This document outlines the high-level design of a bike share prediction system, which encompasses data processing, machine learning models, and a web application for visualization and interaction. The system leverages historical data, weather information, and event schedules to forecast bike availability accurately. By integrating machine learning techniques, the system aims to provide reliable predictions, enhancing operational efficiency and user satisfaction. Additionally, the web application offers a user-friendly interface for stakeholders to interact with the prediction system, visualize data insights, and make informed decisions. This comprehensive approach not only optimizes resource management but also contributes to a sustainable and efficient urban transportation system.

Table of Contents

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.

2. Description

2.1 Problem Perspective

The Bike Share Demand system is a Machine Learning prediction model designed to ensure the availability and accessibility of rental bikes to the public at the right time.

2.2 Problem Statement

The purpose of this work is the development of a prediction model to optimize Bike-Sharing Systems using continuum approaches. The model will be tested through sensitivity analysis, exemplifying relevant scenarios. This will enable bike-sharing operators to make informed decisions about bike redistribution, ensuring optimal availability and accessibility for users, minimizing waiting times, and maintaining a stable supply of rental bikes.

2.3 Proposed Solution

To optimize Bike-Sharing Systems, we will develop a predictive model to forecast the hourly bike count needed to maintain a stable supply of rental bikes. By analyzing historical data and considering factors such as season, weather, temperature, and time of day, we will create scenarios covering peak hours, seasonal variations, and special events for comprehensive testing. The model's impact on system efficiency, user accessibility, and overall performance will be evaluated to ensure it achieves the desired objectives.

2.4 Solution Improvements

To optimize Bike-Sharing Systems, we will develop an advanced machine learning predictive model to maintain a stable supply of rental bikes:

- By analyzing historical usage data and integrating external factors for comprehensive testing.
- The model will be capable of real-time adaptation to ensure it remains accurate and effective under changing conditions.
- Additionally, we will establish a feedback loop to continuously refine the model based on new data and system performance.

2.5 Data Requirements

The data in this study are shared bike riding records and bike station status, as well as local weather information for that period. The data on the operation of bikes contained information on the time of rental, return time, and place of rental. For the purpose of analyzing daily usage, the data were classified on a daily basis and the daily rental volume was set as a class. The data were then compiled by adding weekend and

holiday information to region weather information (including highest and lowest temperature, humidity, wind volume, rainfall, etc.). Thus, the processed data consisted of attributes and instances containing items converted to dummy variables. However, due to the nature of the open data, experiments related to daily usage have already been revealed on the Kaggle website.

Table 1. Bike share metadata

Fields	Description
Season	1: winter, 2: spring, 3: summer, 4: fall
year	0: 2011, 1:2012
month	1 to 12
hour	0 to 23
holiday	1 = Holiday, 0 = Not a Holiday
weekday	day of the week
working day	1 = Neither a weekend nor holiday, 0 = Either a weekend or a holiday
weather	1 = Clear, Few clouds, Partly cloudy, Partly cloudy 2 = Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist 3 = Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds 4 = Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog
temp	Normalized temperature in Celsius. The values are derived via (t-t_min)/(t_max-t_min), t_min=-8, t_max=+39 (only in hourly scale)
atemp	Normalized feeling temperature in Celsius. The values are derived via (t-t_min)/(t_max-t_min), t_min=-16, t_max=+50 (only in hourly scale)
humidity	Normalized humidity. The values are divided to 100 (max)
wind speed	Normalized wind speed. The values are divided to 67 (max)
casual	Number of non-registered user rentals initiated
registered	Number of registered user rentals initiated
count	Number of total rentals (casual + registered)

2.6 Tool Used

- The primary programming language used is Python due to its simplicity and extensive range of libraries.
- Libraries such as NumPy and Pandas are used for data manipulation and analysis, Scikit-learn for building and evaluating machine learning models, and Seaborn and Matplotlib for visualizations.

2.7 Data Constraints

- There is no information on special events that significantly impact bike-sharing demand. This lack of data limits the model's ability to predict fluctuations accurately.
- Developing more sophisticated features that capture complex patterns and interactions in the data can significantly boost the model's performance. For example, creating features that reflect the influence of weather, time of day, and local events can improve predictions.
- Integrating external data sources such as traffic conditions, public transportation schedules, and social media trends can provide additional context and improve the model’s ability to predict demand fluctuations.
- Implementing a mechanism for collecting user feedback on predictions and incorporating this feedback into the model can help refine and improve its accuracy.
- Ensuring that the solution can scale to handle increasing data volumes and more complex models without significant performance degradation is crucial for long-term success.

2.8 Assumptions

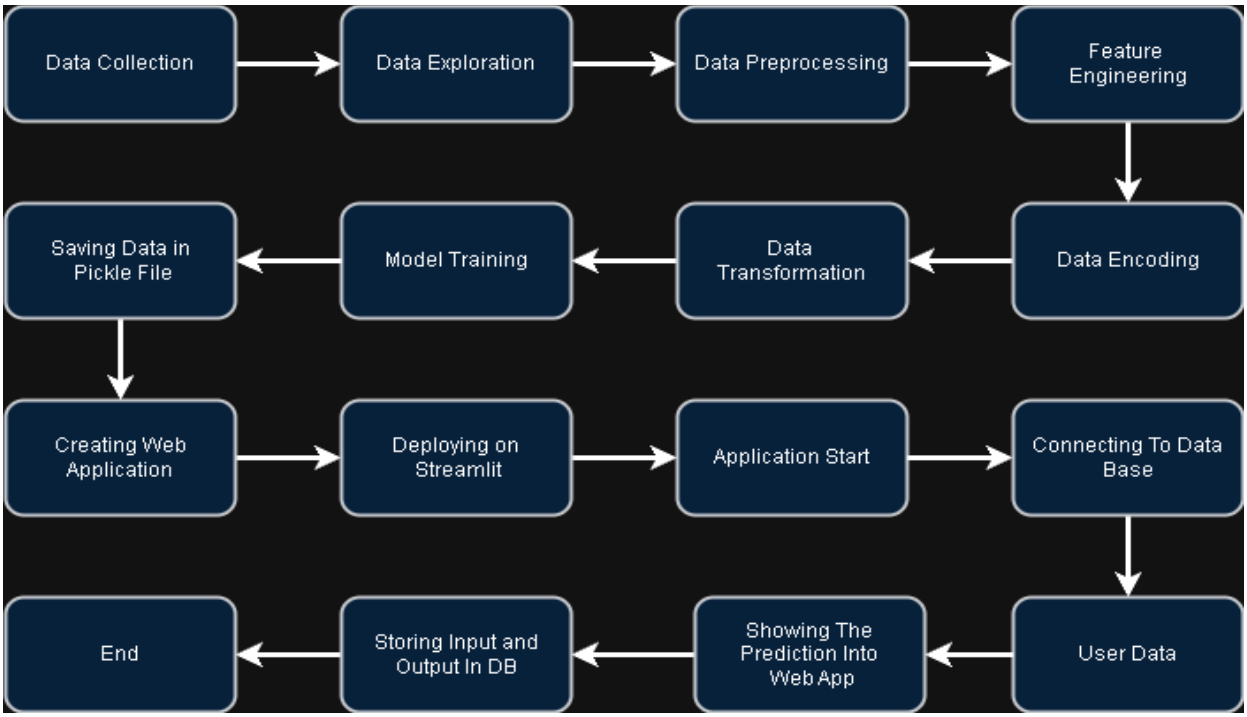
- There is sufficient historical data on bike share usage, including timestamps, locations, weather conditions, and other relevant features.
- The data is clean, well-organized, and accurately reflects real-world usage patterns without significant gaps or anomalies.
- Bike share usage exhibits clear seasonal, weekly, and daily patterns that can be captured and modeled.
- Weather conditions significantly influence bike share usage, and accurate weather data is available for prediction purposes.
- User behavior is relatively consistent and predictable based on historical trends, although there may be some variance due to special events or changes in infrastructure.
- External factors like public holidays, major events, or transportation disruptions are either accounted for in the model or assumed to have minimal impact.

3. Design Flow

3.1 Modeling and Deployment Process

The diagram outlines the workflow for a bike share prediction system, illustrating each step from data collection to deployment and user interaction

Figure 1. Architecture Diagram



3.2 Event Log

Implementing an event log is essential for monitoring the activity and performance of the bike share prediction system. The event log will capture important events, such as:

- Logging when the application starts and stops, along with timestamps.
- Recording user actions, such as data inputs, predictions made, and results displayed.
- Logging database connections, queries executed, data retrieved, and data stored.
- Tracking each prediction made by the model, including input data and predicted output

3.3 Error Handling

Robust error handling ensures the system can gracefully handle unexpected issues and provide meaningful feedback. Key aspects include:

- Capturing and logging details of exceptions, including stack traces and error messages.
- Providing clear and informative error messages to users when an error occurs, ensuring they understand what went wrong and potential next steps.

4. Performance

In the model performance validation step, each prediction model's performance is validated using the following criteria to calculate the error between predicted and actual values and determine prediction accuracy: MAE measures the average magnitude of errors. MSE squares the errors before averaging, emphasizing larger errors more than MAE. R2 measures the proportion of variance in the dependent variable predictable from the independent variables. RMSE is the square root of MSE, providing an error metric on the same scale as the original data.

4.1 Reusability

The developed solution for the bike share prediction system is designed for high reusability. Its modular architecture allows individual components like data processing, feature engineering, and model training to be reused or replaced independently. Configurable parameters and cross-platform compatibility ensure easy adaptation to different datasets and environments. Comprehensive documentation and open-source availability further support reuse and customization, making the system versatile for various bike share programs.

4.2 Compatibility

The bike share prediction system is built using widely supported technologies like Python, Flask, and Stream lit, ensuring broad application compatibility. This design choice allows for seamless deployment across different platforms and integration with existing infrastructure. Additionally, the system's components are designed to work well with various data sources and APIs, enhancing its adaptability and ease of use in diverse environments.

4.3 Utilization

The bike share prediction system can be utilized in various ways to enhance the efficiency and effectiveness of bike share programs:

- **Demand Forecasting:** Predict the number of bike rentals at different times and locations to ensure optimal distribution of bikes across the network.
- **Operational Planning:** Aid in planning bike redistribution and maintenance schedules based on predicted usage patterns.
- **User Insights:** Provide valuable insights into user behaviour and preferences, helping to improve service offerings and user satisfaction.
- **Resource Allocation:** Optimize resource allocation, including bike and docking station availability, to meet demand and reduce operational costs.
- **Decision Support:** Support city planners and policymakers with data-driven insights for infrastructure development and urban mobility planning.

4.4 Deployment

The bike share prediction system is deployed using Stream lit, a powerful and easy-to-use web application framework. This enables real-time interaction and visualization of predictive analytics through a user-friendly interface

5. Conclusion

We implemented a prediction model using 70% of the total dataset and evaluated its performance on the remaining 30% to assess actual prediction accuracy. Our machine learning model incorporates variables influencing demand for shared bikes, notably leveraging urban traffic accident data to enhance predictability. The Random Forest model demonstrates superior predictability compared to existing methods. This system employs metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), R-squared (R2), and Root Mean Squared Error (RMSE) to ensure precise bike rental demand forecasting. This capability facilitates informed decision-making in resource allocation, operational planning, and enhancing user satisfaction. Deployed on stream lit, our system provides a user-friendly interface for interactive visualization and real-time insights, proving invaluable for optimizing bike share programs in urban environments.