

High Level Design (HLD)

Share Bike Demand Prediction

WRITTEN BY
SADASHIV NANDANIKAR

Revision Number: 1.0

Last data of revision: 13/03/202

Contents

Abstract.....	3
1. Introduction.....	4
1.1 Why this HLD document?.....	4
1.2 Scope.....	4
1.3 Definitions	4
2. General Description.....	4
2.1 Problem Statement	4
2.2 Proposed Solution	5
2.3 Further Improvements	5
2.4 Technical Requirements.....	5
2.5 Data Requirements	5
2.6 Tools Used	5
2.7 Assumptions	5
3. Design Flow.....	6
3.1.1 Machine Learning Pipeline stages	6
3.1.2 Deployment Process	7
3.2 Logging.....	7
3.3 Error Handling	7
4. Performance	7
4.1 Reusability	7
4.2 Application compatibility.....	7
4.3 Deployment.....	7

Abstract

Bike sharing systems are a new generation of traditional bike rentals where the whole process from membership, rental and return back has become automatic. Through these systems, users are able to easily rent a bike from a particular position and return back at another position. Currently, there are about over 500 bike-sharing programs around the world which is 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 HLD document?

A High-Level Design (HLD) document is a technical document that describes the architecture of a software system or application at a high level. The purpose of an HLD document is to provide a blueprint for developers and stakeholders to understand the high-level technical details of the system, including the architecture, components, modules, interfaces, and other high-level design details

- The HLD document has several purposes:
- Establishing the system's architecture.
- Defining the components and modules.
- Providing a common understanding.
- Facilitating communication and collaboration.
- Guiding the low-level design.
- List and describe the non - functional attributes like:
 1. Security
 2. Reliability
 3. Maintainability
 4. Portability
 5. Reusability
 6. Resource
 7. Utilization

1.2 Scope

An HLD document is an essential document that helps ensure the successful development of a software system or application by providing a clear and high-level technical specification. The HLD document is often used as a starting point for the development process and serves as a foundation for the more detailed low-level design documents.

1.3 Definitions

TERM	DESCRIPTION
IDE	Integrated Development Environment.
Stream lit	Streamlit's open-source app framework
VS Code	VS Code stands for Visual Studio code and it is an IDE.
MongoDB	MongoDB is NoSQL database, used for data storage

2. General Description

2.1 Problem Statement

The problem statement involves building an end-to-end regression task for bike-sharing systems. With over 500 programs and 500 thousand bicycles worldwide, bike-sharing systems are becoming increasingly important for traffic, environmental and health issues. The aim is to create the best performing hyper-tuned machine learning model to predict bike-sharing data, allowing users to choose their preferred deployment options.

2.2 Proposed Solution

To build an end-to-end regression task for bike-sharing systems, we can use machine learning models to predict bike usage patterns and demand. By analysing the data generated by bike-sharing programs, we can identify factors that influence bike usage such as weather, time of day, temperature, and holidays. This data can be used to build predictive models that can accurately forecast bike demand, helping bike-sharing programs to optimize their bike distribution and management strategies.

2.3 Further Improvements

Further improvements in bike-sharing systems can include the integration of smart technology, such as IoT sensors and real-time data analytics, to improve the accuracy of demand forecasting and optimize bike distribution. Additionally, the inclusion of features such as bike availability notifications, personalized route planning, and in-app payments can enhance the user experience and encourage greater adoption of bike-sharing services.

2.4 Technical Requirements

There are no hardware requirements for this project. For the backend, we will be using Streamlit to deploy the application.

2.5 Data Requirements

The dataset is currently available as a raw file on UCI. To minimize latency when accessing the dataset, opted to store it in a MongoDB database.

2.6 Tools Used

- The environment was created using Python 3.8.10.
- The data is stored in a MongoDB database.
- VS Code is used as IDE.
- GitHub serves as the code repository.
- The Streamlit application is utilized to host the application.

Other libraries such as Pandas, NumPy, Matplotlib, Seaborn, and Scikit-Learn were utilized in building the whole model.

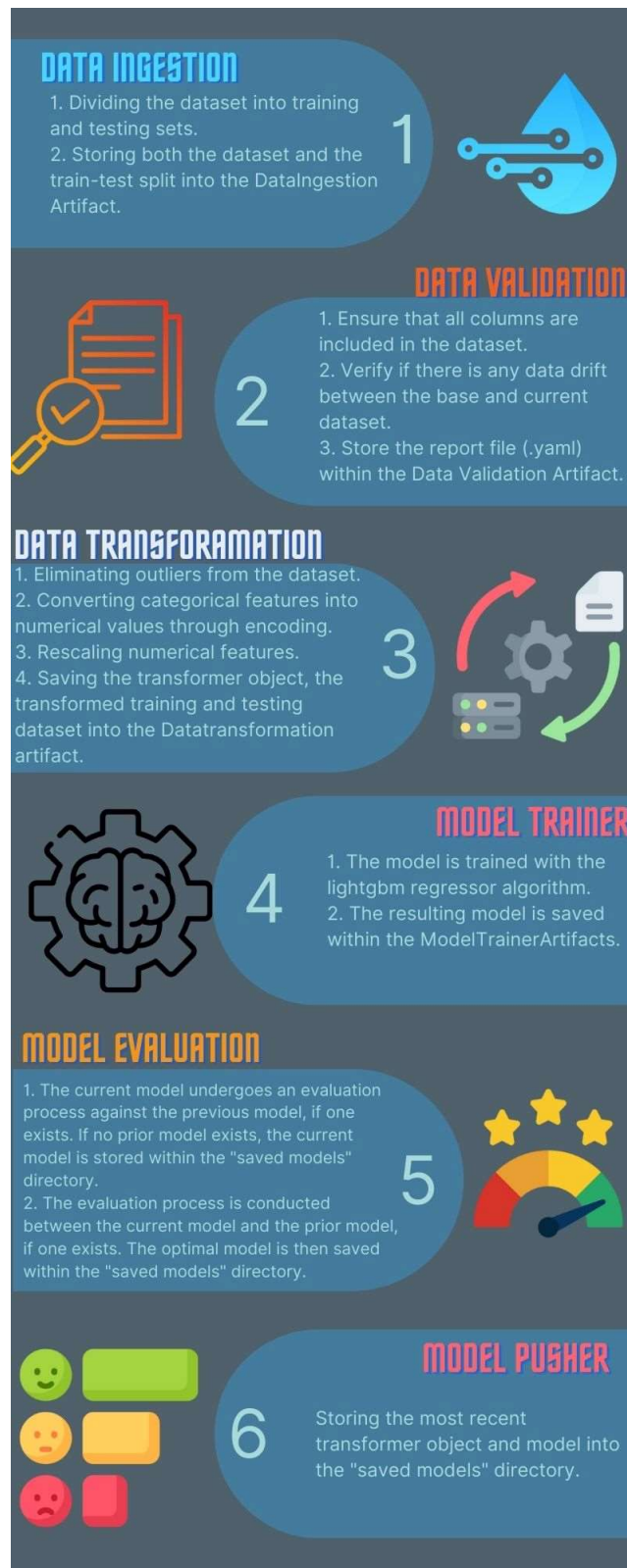


2.7 Assumptions

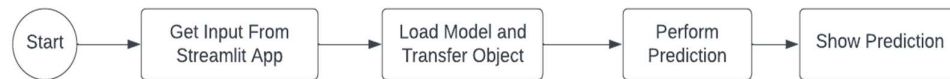
The primary aim of the project is to implement a regressor model, assuming that all functionalities are performing as expected.

3. Design Flow

3.1.1 Machine Learning Pipeline stages



3.1.2 Deployment Process



3.2 Logging

The entire execution process is logged with timestamps to facilitate the quick identification of errors.

3.3 Error Handling

The Exception handling is implemented to find the errors quickly and can easily be detected any wrong as has happened.

4. Performance

4.1 Reusability

The code has been written as per the industry standards and can be reused without any interruption.

4.2 Application compatibility

The various components of this project are built using python programming language. Each component has its own task to perform according to the pipeline

4.3 Deployment

This Model is deployed using Streamlit application.