Rental Bike Share Prediction.

High Level Design Documentation (HLD).

Project for:- iNeuron (Rental Bike Share Prediction Internship) By:- Piyush Aggarwal.

Content

1. 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.

2. Introduction.

2.1. Why this High level Design Documentation ?

The purpose of High Level Documentation is to (HLD) 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 the 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

2.2. Scope.

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

3. Description.

3.1. Problem Perspective and 3.2. Problem Statement

The most important problem from a business point of view for bike-sharing system like Capital Bikeshare is to predict the bike demand on any particular day. While having excess bikes results in wastage of resources (bike maintenance and land/bike stand required for parking and security), having fewer bikes leads to revenue loss (ranging from a short term loss due to missing out on immediate customers to potential longer term loss due to loss in future customer base). Thus having an estimate on the demands would enable efficient functioning of this company Capital Bikeshare.

3.3. Proposed Solution.

The solution proposed to take the required inputs from the user through the web interface created by us, pass this requirement to our machine learning model, and based on these inputs we have to print that ''XYZ'' is the total number of bike counts getting rented around them based on different conditions.

3.4. Solution Improvements.

The bike share prediction can help the company and users to determine which conditions are favorable to rent a bike.
The companies can understand how they can improve the end to end user experience.
Bike share system can help the users to see if it's favorable to rent a bike based on Temperature, Humidity level, Season, Weekend etc thus fulfilling Bike rental demands, avoid wastage of resources by only deploying required number of bikes thus saving cost for bikes and even efficiently manage space for parking.

4. Requirements

4.1 Hardware Requirements:-

A working computer to code with active internet connection.

4.2 Tools / Software Requirements:-

- Python version used for this project 3.9.0(This may get updated and some features might not be available in new version.)
- Python libraries such as NumPy, pandas, matplotlib, seaborn and scikit-learn (Used for implementation of machine learning algorithms.)
- Jupyter for Exploratory Data Analysis and testing code, Visual studio code is used as an IDE for writing the code.
- HTML, CSS & Java Scripts are used for developing the front end of our web application.
- Flask is used for backend development.
- Github is used as the version control system.
- railway.app (a platform like Heroku) is used for deployment.

5. Data Requirements.

Whenever we are working on any project the data is completely dependent on the requirement of the problem statement. For this project the problem statement was to create a Hyper tuned Regression machine learning model which can predict the total count of bikes rented at a particular day based on various parameters.
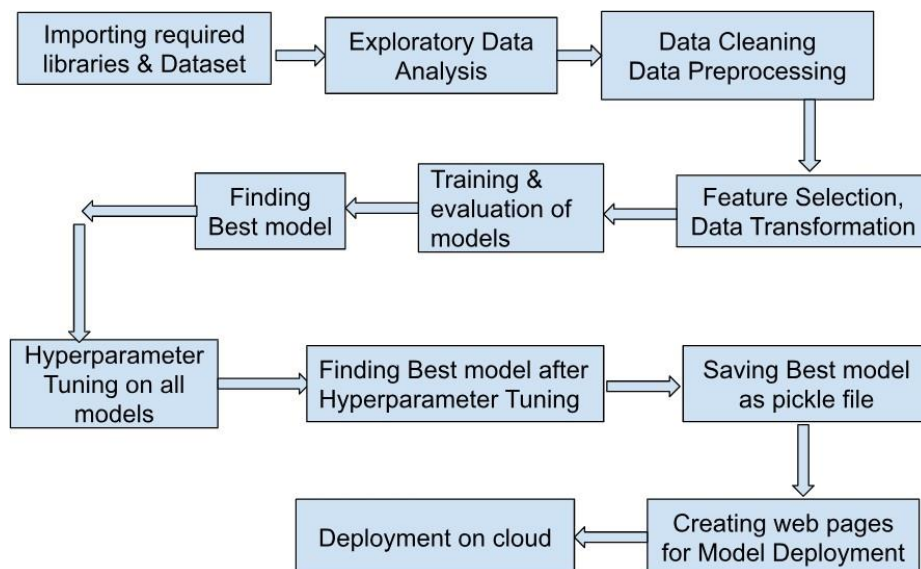
6. Constraints.

The web application should be user friendly so that without knowing any technical information user should be able to use our predictive system.

7. Assumptions.

The main objective of the project is to utilize the data which is provided by the user and to predict the count of rental bikes that would be required. The web application should be accessible from every system which is connected to the internet.

8. Design Flow.

Importing required libraries & Dataset → Exploratory Data Analysis → Data Cleaning Data Preprocessing

Finding Best model ← Training & evaluation of models ← Feature Selection, Data Transformation

Hyperparameter Tuning on all models → Finding Best model after Hyperparameter Tuning → Saving Best model as pickle file

Deployment on cloud ← Creating web pages for Model Deployment

The above flow chart represents the flow of any machine learning model which needs to be created .

9. Logging & Error Handling.

Every important step is logged within the system that runs internally, it basically shows us the data time of each process which is done with our system. It provides us with logging information for end to end web applications.

The logging which we have done in the above process helps us to handle the error because the error is being logged in log files (every time we run code) so that the developer can rectify it.

10. Performance Evaluation.

10.1. Reusability
The elements of the code is written in such a way that it can be changed and easily written again without changing or creating an entirely different web application from scratch. Just the HTML requirements and the app.py file constraints need to be adjusted and changed .

10.2. Application Compatibility.
The elements of the project are written in python, it acts as the interface between the machine learning model and the user. The application can run on any system with a network connection.

10.3. Resource Utilization.
Once the task is assigned to the model doubtlessly it will use all the resources which are allocated until the task is finished.

11. Deployment.

This model is deployed on railway.app (a platform like Heroku). The following are the steps to deploy this application:

Follow the steps to deploy on railway app using any of the instruction videos available on YouTube.

Best instruction video found:
https://www.youtube.com/watch?v=WzxEHzb2NzM&t=926s

12. Conclusion.

We have successfully built an end-to-end web application using machine learning that can help predict the total count of bikes rented by the users based on various conditions. This type of system can help users to get a better understanding of whether the day is good for riding bikes and make their decision making easier.

This system will help companies to help the users and provide a better end to end user experience, fulfill bike rental demands of users, avoid wastage of resources by only deploying required number of bikes thus saving cost and space for parking.