LOW LEVEL DESCIGN

BIKE SHARING SYSTEM PREDICTION

October 28, 2021

hRISHIKESH

hrishi4musiq@gmail.com

Contents

1. Introduction......................................................................................................................................2

1.1. What is Low-Level design document? ...........................................................................................2

1.2. Scope..............................................................................................................................................2

2. Architecture.......................................................................................................................................3

3. Architecture Description....................................................................................................................4

3.1 Technology stack…………………………………………………………………………………………………………………………4

3.2. Data for the problem......................................................................................................................4

3.3. Data Transformation.......................................................................................................................4

3.4. Data Insertion intoDatabase...........................................................................................................4

3.5. Export Data from Database.............................................................................................................4

3.6. Data Pre-processing .......................................................................................................................4

3.7. Model Building................................................................................................................................5

3.8. Data from User................................................................................................................................5

3.9. Data preparing.................................................................................................................................5

3.10. Model called for the data..............................................................................................................5

3.11. Predicted data...............................................................................................................................5

3.12. Saving output at Database............................................................................................................5

4. Unit Test Cases ..................................................................................................................................6

1. Introduction

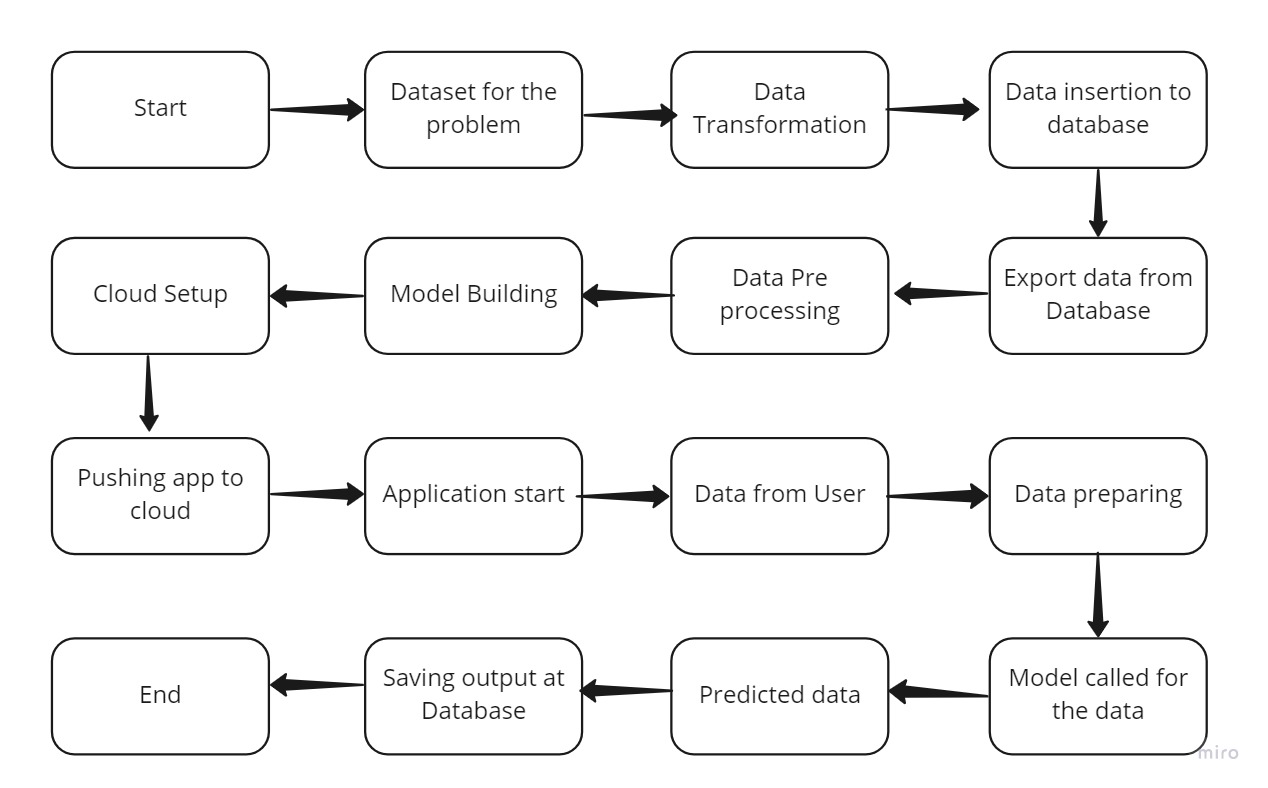
1.1. What is Low-Level design document?

The goal of LLD or a low-level design document (LLDD) is to give the internal logical design of the actual program code for Bike sharing prediction System. LLD describes the class diagrams with the methods and relations between classes and program specs. It describes the modules so that the programmer can directly code the program from the document.

1.2. Scope

Low-level design (LLD) is a component-level design process that follows a step-by-step refinement process. This process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work

2. Architecture



3. Architecture Description

3.1 Technology stack

Front End HTML/CSS

Backend Python Flask

Database Cassandra

Deployment AWS

3.2. Data for the problem

UCI\_Bike\_Sharind dataset from UCI repository which has 731 datapoints. This dataset contains last two years dates, weathers situation details, importance of days(working/Holiday), number of bikes used by casual and registered users and total count of bikes used.

3.3. Data Transformation

In the Transformation Process, we will convert our original dataset which is in excel format to CSV

Format.

3.4. Data Insertion into Database

a. Database Creation and connection - Create a database with name passed. If the database is

already created, open the connection to the database.

b. Table creation in the database.

c. Insertion of files in the table

3.5. Export Data from Database

Data Export from Database - The data in a stored database is exported as a CSV file to be used for

Data Pre-processing and Model Training.

3.6. Data Pre-processing

The classical machine learning tasks like Data Exploration, Data Cleaning,

Feature Engineering, Feature Selection, Feature scaling, Data Balancing using sampling techniques etc.

3.7. Model Building

After all the data pre-processing we will find the best model data. Each try, algorithms

will be passed with the best parameters derived from Grid-Search. We will calculate the Accuracy scores for models and select the model with the best score. Then the best model will be saved for the prediction purpose.

3.8. Data from User

Here we will collect data from user such as date , temperature, feeling temperature, humidity, season, weather situation for a particular day.

3.9. Data preparing

Here given data will be undergone all the pre-processing techniques (3.5) which we done on the early available dataset.

3.10 Model called for the data

The saved model will be called for the prediction on the given data.

3.11 Predicted data

On the given data the loaded model will perform prediction.

3.12 Saving output in Database

The given data with the newly predicted data will be inserted to the pre-defined table available in the database for future usages.

4. Unit Test Cases

|  |  |  |
| --- | --- | --- |
| Test Case Description | Pre-Requisite | Expected Result |
| Verify whether the Application URL is  accessible to the user | 1. Application URL  should be defined | Application URL should be  accessible to the user |
| Verify whether the Application loads  completely for the user when the URL  is accessed | 1. Application URL  is accessible  2. Application is  deployed | The Application should load  completely for the user when the  URL is accessed |
| Verify whether user is able to see input  fields on logging in | 1. Application is  accessible  2. User is able to see input fields. | User should be able to see input  fields on logging in |
| Verify whether user is able to edit all  input fields | 1. Application is  accessible  2. User is able to see input fields.  3. User is able to edit input fields. | User should be able to edit all input  Fields |
| Verify whether user gets Submit  button to submit the inputs | 1. Application is  accessible  2. User is able to see input fields.  3. User is able to edit input fields.  4.User is able to see submit button. | User should get Submit button to  submit the inputs |
| Verify whether user is presented with  prediction results on clicking  submit | 1. Application is  accessible  2. User is able to see input fields.  3. User is able to edit input fields.  4.User is able to see submit button. | User should be presented with  Predicted with results on clicking submit |