iNeuron

Low Level Design (HLD)
Bike Share Prediction

Revision Number : 1.2
Last Date of revision : 10/06/2023

Document Version Control

Date Issued	Version	Description	Author
02-06-2023	1.0	Abstract Introduction Architecture	SAHIL JOSAN
04-06-2023	1.1	Architectural Design	SAHIL JOSAN
10-06-2023	1.2	Deployment Unit Test Cases	SAHIL JOSAN

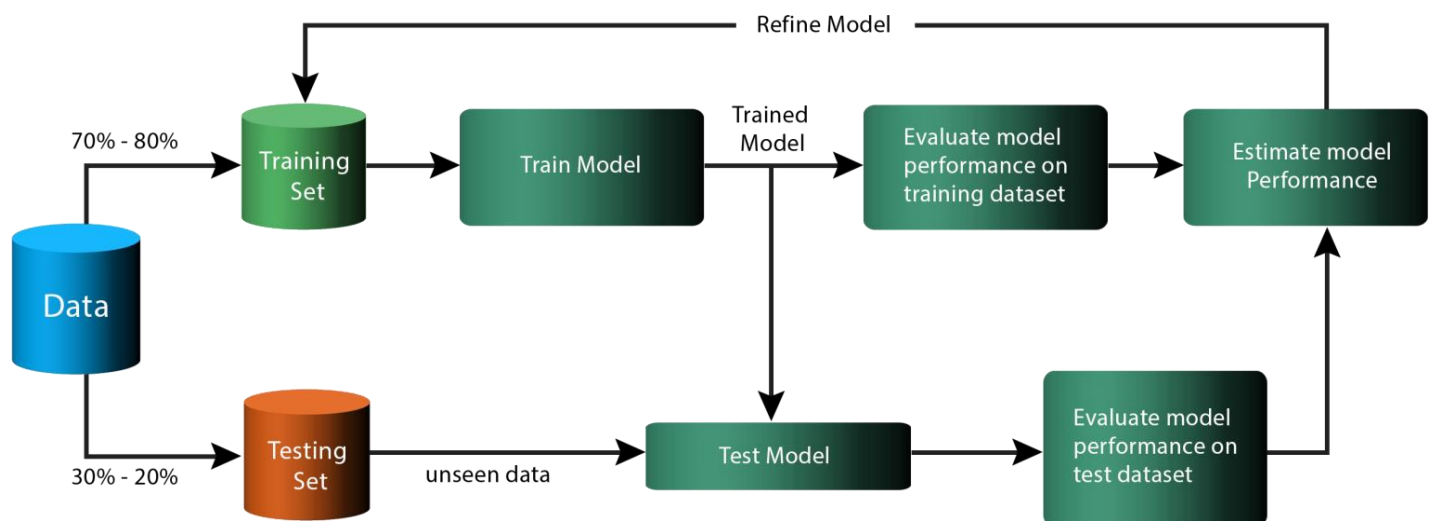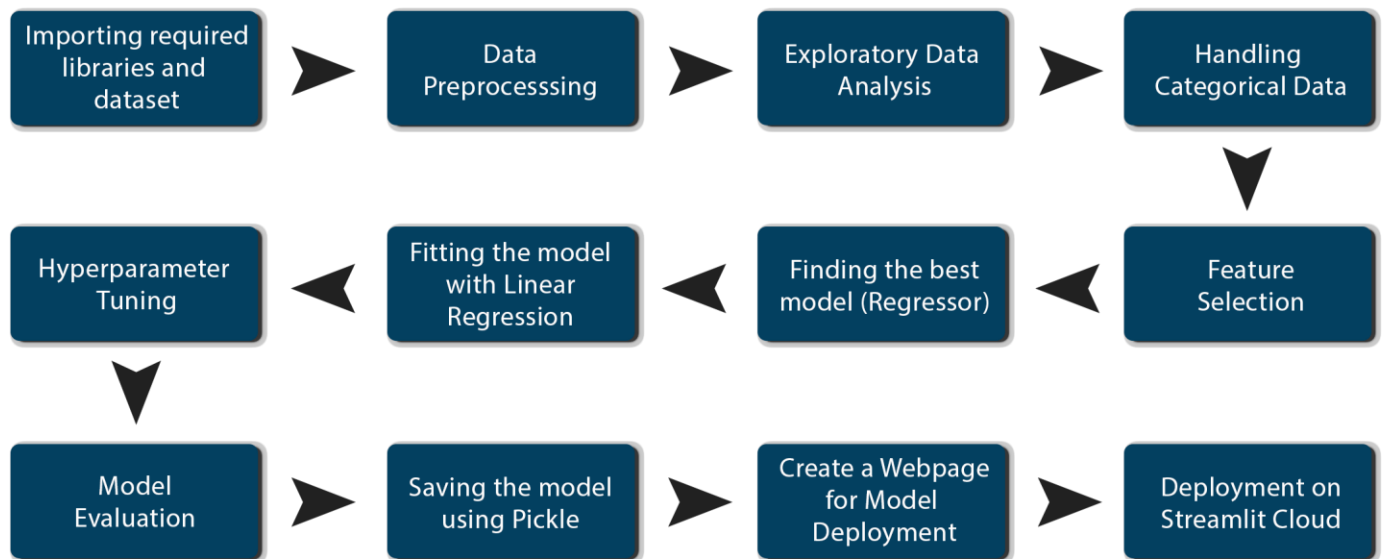# Contents

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. 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-bystep 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. Archiecture

```
┌──────────────────┐     ┌──────────────────┐     ┌──────────────────┐     ┌──────────────────┐
│ Importing required│ ──▶ │      Data        │ ──▶ │ Exploratory Data │ ──▶ │    Handling      │
│  libraries and   │     │  Preprocesssing  │     │     Analysis     │     │ Categorical Data │
│     dataset      │     │                  │     │                  │     │                  │
└──────────────────┘     └──────────────────┘     └──────────────────┘     └──────────────────┘
                                                                                     │
                                                                                     ▼
┌──────────────────┐     ┌──────────────────┐     ┌──────────────────┐     ┌──────────────────┐
│  Hyperparameter  │ ◀── │ Fitting the model│ ◀── │ Finding the best │ ◀── │     Feature      │
│      Tuning      │     │   with Linear    │     │ model (Regressor)│     │    Selection     │
│                  │     │    Regression    │     │                  │     │                  │
└──────────────────┘     └──────────────────┘     └──────────────────┘     └──────────────────┘
         │
         ▼
┌──────────────────┐     ┌──────────────────┐     ┌──────────────────┐     ┌──────────────────┐
│      Model       │ ──▶ │ Saving the model │ ──▶ │ Create a Webpage │ ──▶ │  Deployment on   │
│    Evaluation    │     │   using Pickle   │     │    for Model     │     │  Streamlit Cloud │
│                  │     │                  │     │   Deployment     │     │                  │
└──────────────────┘     └──────────────────┘     └──────────────────┘     └──────────────────┘
```

3. Architecture Description

3.1 Data Description
This dataset contains the hourly and daily count of rental bikes between years 2011 and 2012 in Capital bikeshare system with the corresponding weather and seasonal information. I have used daily count data set which has 731 records and 14 attributes, which shows information like season, yr, month, weekday, weather the day was working day or not elc.

3.2 Data Ingestion
In the Ingestion Process, we will convert our original dataset which is in Zip format to csv format. After that we will split them into train and test dataset.

3.3 Data Validation
In Data validation steps we could use Null value handling, outlier handling, Imbalanced data set handling, Handling columns with standard deviation zero or below a threshold, etc.

3.4 Data Transformation
In this step we will transform out data. We will use standard scaler for numeric data and we will convert categorical data into numeric data using label encoding technique so that machine can understand it.

3.5 Model Building
Here we will build the Machine Learning model using all regression algorithms.

3.6 Model Evaluation
Here model evaluation will be done on the model which we got in the model building stage. We can define base accuracy of the model and if model accuracy is higher then base accuracy, then only our model will accept otherwise it will be rejected.

3.7 Model Deployment
Here model will be deployed to Streamlit cloud platform.

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 application	1. Application is accessible 2. User is logged in to the application	User should be able to see input fields on application
Verify whether user is able to edit all input fields	1. Application is accessible 2. User is logged in to the application	User should be able to edit all input fields
Verify whether user gets Submit	1. Application is accessible 2. User is logged in to the application	User should get Submit button to submit the inputs
Verify whether user is getting predicted results on clicking submit	1. Application is accessible 2. User is logged in to the application	User should be presented with predicted results on clicking submit