High Level Design (HLD)

Revision Number: 1.0
Last date of Revision: 06/01/2024

Document Version Control

Date Issued	Version	Description	Author
06/01/2024	1	Initial HLD- V1.0	Ajith Kumar V, HariHara Sudan R

Abstract

Nowadays, traffic is a major issue for everyone, and it is a source of stress for anyone who has to deal with it on a daily basis. The growth of the population delays traffic and makes it worse day by day. The settlement of modern civilization looks at it, but it is unable to act in such a way as to protect people. We can watch traffic, collect data, and anticipate the next and subsequent observations using a variety of approaches and patterns. The observation agency then makes observations, which are then required out and predictions are made. Being stuck in a cosmopolitan city's traffic is the most common occurrence in one's life.

The goal of this project is to build a prediction model using multiple machine learning techniques and to use a template to document the end-to-end stages. We're trying to forecast the value of a continuous variable with the Metro Interstate Traffic Volume dataset, which is a regression issue.

1. Introduction

1.1. Why these High-Level Design Documents?

The purpose of this High-Level Design (HLD) Documents is to add necessary details to the current project description to represent a suitable for coding. This document is also intended to help detect contradictions before coding. And can be used as a reference manual for how the modules interact at a high level.

The HLD will be:

- Present all of the design aspects and define them in detail.
- Describe the user interface being implemented.
- Describe the needed Python libraries for the coding.
- 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 and mildly-technical terms which should be understandable to the administrators of the system

1.3 Definition

TERM	DESCRIPTION
LSTM	Long Short-term Memory

2. General Description

2.1. Product Perspective

The Traffic Volume Prediction is a ML based Web application that is able to predict future traffic by analyzing the past records. It will give the traffic volume at a particular time.

2.2. Problem Statement

To build a prediction model that can forecast traffic volume in a particular area based on the given information. We have to build an app that uses machine learning techniques to forecast and provide an estimate to the user.

2.3 Proposed Solution

We will be using LSTM model for our regression task to predict the future traffic. The user will fill the required feature as input and will get results through web application. The system will get features and it will be passed into the backend where the features will be validated and preprocessed and then it will be passed to LSTM model to predict the final outcome. MLOps pipeline has also been added to the application will a single click user can fetch data from database, create a model, train it based on hyperparameters, evaluate it and use it in production.

2.4 Data Requirements

The Metro Interstate Traffic Volume dataset required for building of the project is used from UC Irvine website and ingested into Cassandra/Astra DB. For building the ml model we will use this dataset and it consist of 48203 rows and various information about weather and holidays which acts as features.

2.5 Tools Used

The programming language is python that is used here, also we will use some other python-based libraries like tensorflow for model development, numpy, pandas for data manipulation, flask for web frameworks, dvc for data versioning, cassandra-driver for connecting with database, Visual Studio Code as python IDE for coding and storing all code files in Github

2.6 Constraints

The system should be user friendly, the user should get all proper messages while using the web app. He/She should get a proper error message if he/she has done something wrong on the web-app page. All the errors and results should be delivered in the easiest possible way and all the buttons are going to insert on the webpage should be labeled properly so user is not confused to use the system.

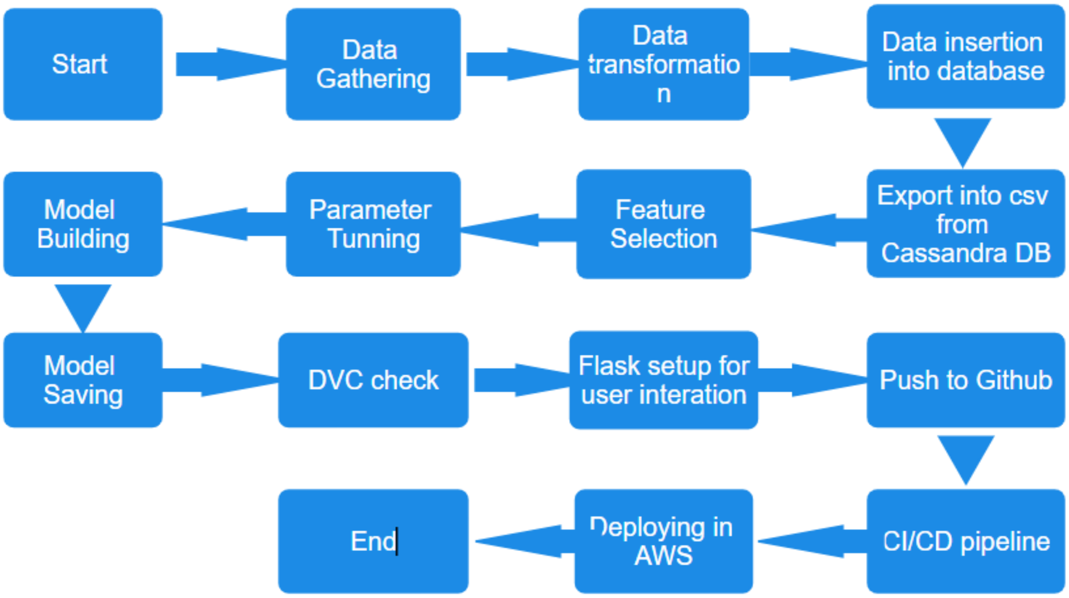
2.7 Assumptions

The main objective is to implement a system that will produce approximate future traffic volume in a particular hour of a day.

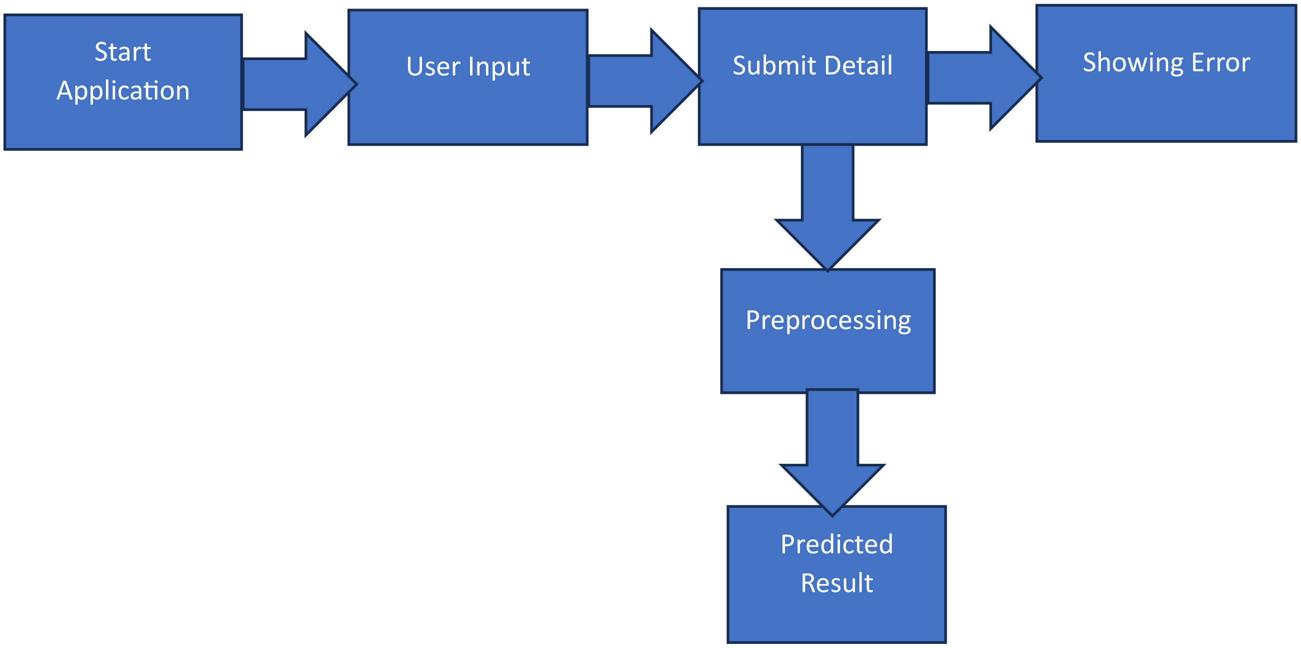
3. Design Details

3.1 Process Flow

We will be using following process flow for this project. The process will be based on modular coding i.e using oops concept to build the entire project from start to end.



3.2 Deployment Process



3.3 Error Handling

If any error occurred in the processing way then the error message should be shown to the user in a completely non-technical way that can be understandable by any person and meaningful error message should be shown. All the errors that might occur and handled properly and we have logged every event and errors that might occur in the application.

4 Performance

The Traffic Volume Prediction app is dependent on LSTM model and we will train and get the model when the validation loss has converged. Our system performance will be based on the data we are going to feed the algorithms and the model which we are going to use.

4.1 Reusability

The code and the module are created during the time of building the project and the coding guidelines are maintained and its written in modular fashion. Our system should have the flexibility to work properly from any location and it should handle any improper input value from the user and even if more rows are added in db the model can handle it and run.

4.2 Application Compatibility

Different libraries and python programming language are used to build the system. Every library has its own functionality and it should work properly without fluctuation. Flask will be used for making the web API and web application. All the components of the application should work properly and it should produce a result without any interpretation.

4.3 Resource Utilization

Our application should utilize the give resource properly and it should use a minimal amount of internet to work and call the APIs on the web page. Our system does not use much computational resources hence it will make the application slow. Out application will be deployed in cloud platform and it should utilize the resource given on the cloud and work properly.

5 Deployment

For the deployment process, we will be using AWS EC2 and AWS ECR for hosting our application. The cloud platform will run the system and it will give the flexibility to use our application globally.

6 Conclusion

The Traffic Volume prediction app will help the commuters to predict hourly forecast of traffic and help them in their day-to-day life. We have past record about weather, holiday and traffic volume information. We will analyze the past data and will build an ML model that can identify the internal pattern and be able to predict the traffic volume in future.