High Level Design (HLD)

Shipment Pricing Prediction

Problem Statement:

The market for supply chain analytics is expected to develop at a CAGR of 17.3 percent from 2019 to 2024, more than doubling in size. This data demonstrates how supply chain organizations are understanding the advantages of being able to predict what will happen in the future with a decent degree of certainty. Supply chain leaders may use this data to address supply chain difficulties, cut costs, and enhance service levels all at the same time. The main goal is to predict the supply chain shipment pricing based on the available factors in the dataset.

Dataset:

https://www.kaggle.com/datasets/divyeshardeshana/supply-chain-shipment-pricing-data

Why this High-Level Design Document?

The purpose of this High-Level Design (HLD) Document 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 a high level.

Benefits to build HLD:

- Describe the user interface that is being used Describe the hardware and software interfaces Present all of the design elements and define them in depth
- Describe the necessary performance standards.
- Include the project's architecture and design elements.
- List and explain the non-functional characteristics,
- such as security, Reliability, Maintainability, Reusability etc.

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 to mildly-technical terms which should be understandable to the administrators of the system.

Product Perspective

The Shipment Price Prediction system is a machine learning-based default prediction model which will help us to know the Shipping Price of Goods.

Solution Arch.

The solution proposed here is a BSS prediction model can be implemented to perform above mention use case. In this case, we have to entre a detail like Shipment Mode, Country, Brand, Unit of Measurement, Weight(Kg) Based on the above details, the model predicts the possible demand of bikes on that particular day.

As well as we have built a batch prediction where we can upload our entire dataset in a single shot and we will get our output in a CSV file.

Data Requirements:

- Data based upon our problem statement
- We need as much data as available.
- Mode of Transportation
- Freight Cost(USD)
- Weight of Items

Technology Used:

- Python Programming
- Library used
  - Numpy
  - Pandas
  - Matplotlib
  - Seaborn
  - Sklearn
- Machine Learning
  - Single Prediction
  - Batch Prediction
- MongoDB Database
- Data Drift Handling
  - Evidently
- Docker
- Flask API
- HTML, CSS, JS for Designing
- GitHub for version control

Logging:

The system should log every event so that the user will know what process is running internally.

Database

Database Creation and connection - Create a database with name passed. If the database is already created, open the connection to the database.

- Table creation in the database.
- Insertion of files in the table

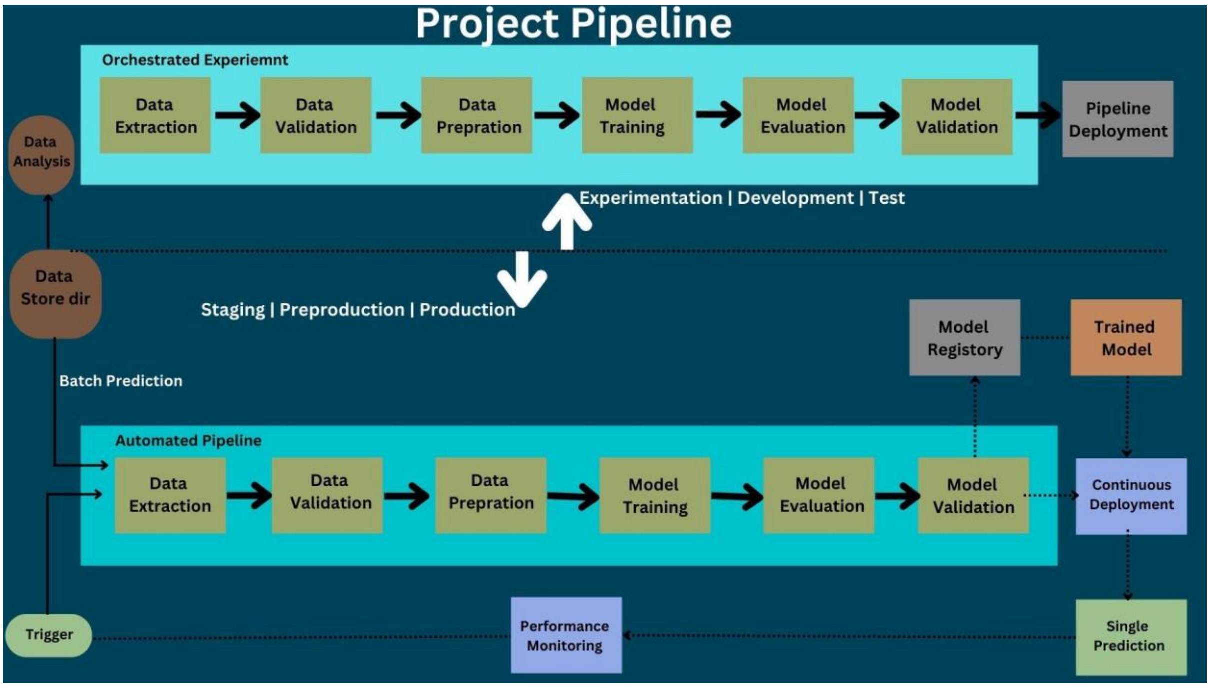
Constraints

This application must be user friendly, as automated as possible and users should not be required to know any of the workings.

Assumptions

The main objective of the project is to implement the use cases as previously mentioned (Problem Statement) for new dataset that comes through forms in application webpage. Machine Learning based prediction model is used for detecting the above-mentioned use cases based on the input data. It is also assumed that all aspects of this project can work together as the designer is expecting.

Project Arch. (Proposed methodology)



Performance

This application is used for prediction of the total number of bikes in demand by a bike sharing system for a particular day. As we already trained the model with sufficient data it predicts the default with high accuracy. Also, model retraining is very important to improve the performance.

Reusability
The code written and the components used should have the ability to be reused with no problems.

Resource Utilization
When any task is performed, it will likely use all the processing power available until that function is finished.

Thank you

Happy Learning!