HIGH LEVEL DOCUMENT (HLD)
SHIPMENT PRICING PREDICTION

Project By :

Ritesh Pandey

Document Version control

Date issued	Version	Description	Author
23/06/2023	1	Initial HLD	Ritesh Pandey

Contents

Document Version Control 2
Abstract 4
1 Introduction 5
1.1 Why this High-Level Design Document? 5
1.2 1.2 Scope 5

2 General Description 6
2.1 Definitions 6
2.2 Product Perspective 6
2.3 Problem statement 6
2.4 Proposed Solution 6
2.5 Technical Requirements 7
2.6 Data Requirements 7
2.7 Tools used 8
2.7.1 Hardware Requirements 9
2.8 Constraints 9
2.9 Assumptions 9

3 Design Details 9
3.1 Process Flow 9
3.1.1 Model Training and Evaluation 10
3.1.2 Deployment Process 10
3.2 Event log 11
3.3 Error Handling 11

4 Performance 11
4.1 Reusability 11
4.2 Application Compatibility 11
4.3 Resource Utilization 11
4.4 Deployment 12

5 Dashboards 12
5.1 KPI's (Key Performing Indicators) 12

6 Conclusion 12

ABSTRACT

The goal of this project is to develop a model that can accurately predict supply chain shipment pricing based on a variety of factors. The market for supply chain analytics is expected to experience significant growth over the next few years, as organizations increasingly recognize the benefits of being able to forecast future events with a high degree of certainty. By accurately predicting supply chain pricing, supply chain leaders can address challenges, reduce costs, and improve service levels.

1. Introduction

1.1 Why this High-Level Document?

The purpose of this High-Level Design (HLD) Document is to add 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.

The HLD will:

1. Present all of design aspects and define them in detail
2. Describe all user interface being implemented
3. Describe the hardware and software interfaces
4. Describe the performance requirements
5. Include design features and architecture of the project

List and describe the non-functional attributes like:

1. Security
2. Reliability
3. Maintainability
4. Portability
5. Reusability
6. Application compatibility
7. Resource utilization
8. Serviceability

1.2 Scope

The HLD documentation presents the structure of the system, such as 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.

2. General Description

2.1 Definitions

TERM	DESCRIPTION
SPP	Shipment Price Prediction
IDE	Integrated Development Environment
API	Application Programming Interface
Postman	Postman is an API development tool

2.2 Product Description

The SPP is a Machine Learning based regression model which helps us to do predictive analysis on the Shipment Price of a consignment using certain parameters.

2.3 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.

2.4 Proposed Solution

Using all the standard techniques used in the life-cycle of a Data Science project starting from Data Exploration, Data Cleaning, Feature Engineering, Model Selection, Model Building and Model Testing and also building a frontend where a user can fill the information in the form input and receive instant output based on our prediction model. Throughout the project, we will utilize various techniques and methods to optimize and improve the accuracy of our model.

2.5 Further Improvements

The SPP can be easily embedded inside any website or an application and can be used to find out Shipment Price for a consignment from different manufacturing locations around the world.
This can also be improved further by feeding the model with more data, this data can be from various data repositories, Import-Export agencies, data scraped from internet, etc.

2.6 Data Requirements

Data requirement completely depend on our problem statement. We need the dataset from past few years, to train our model. Each record in the dataset must have the following features which are important to determine Shipment Pricing:

1. **pq #:** Price quote (PQ) number
2. **asn/dn # :**Shipment number: Advanced Shipment Note (ASN) for Direct Drop deliveries, or Delivery Note (DN) for from RDC deliveries.
3. **Country:** Destination country.
4. **Managed by** : SCMS managing office: either the Program Management Office (PMO) in the U.S. or the relevant SCMS field office.
5. **fulfil via:** Method through which the shipment was fulfilled: via Direct Drop from vendor or from stock available in the RDCs.
6. **vendor Inco term:** The vendor INCO term (also known as International Commercial Terms) for Direct Drop deliveries.
7. **Shipment mode:** Method by which commodities are shipped.
8. **Pq first sent to client date**: Date the PQ is first sent to the client.
9. **Po sent to vendor date**: Date the PO is first sent to the vendor.
10. **Scheduled delivery date**: Current anticipated delivery date.
11. **Delivered to client date**: Date of delivery to client.
12. **Delivery recorded date**: Date on which delivery to client was recorded in SCMS information systems.
13. **Product group**: Product group for item, i.e., ARV, HRDT.
14. **Sub classification**: Identifies relevant product sub classifications, such as whether ARVs are pediatric or adult, whether a malaria product is an artemisinin-based combination therapy (ACT), etc.
15. **Vendor**: Vendor name.
16. **Item description**: Product name and formulation from Partnership for Supply Chain Management (PFSCM) Item Master.
17. **Molecule/test type**: Active drug(s) or test kit type.
18. **Brand**: Generic or branded name for the item.
19. **Dosage**: Item dosage and unit.
20. **Dosage form**: Dosage form for the item (tablet, oral solution, injection, etc.).
21. **Unit of measure (per pack)**: Pack quantity (pills or test kits) used to compute unit price.
22. **Line-item quantity**: Total quantity (packs) of commodity per line item.
23. **Line-item value**: Total value of commodity per line item.
24. **Pack price**: Cost per pack (i.e., month s supply of ARVs, pack of 60 test kits).
25. **Unit price**: Cost per pill (for drugs) or per test (for test kits).
26. **Manufacturing site**: Identifies manufacturing site for the line item for direct drop and from RDC deliveries.
27. **First line designation**: Designates if the line in question shows the aggregated freight costs and weight associated with all items on the ASN DN.
28. **Weight (kilograms)**: Weight for all lines on an ASN DN.
29. **Freight cost (USD)**: Freight charges associated with all lines on the respective ASN DN.

30. **Line-item insurance (USD)**: Line-item cost of insurance, created by applying an annual flat rate to commodity cost.

2.7 Tools used

Python programming language and frameworks such as NumPy, Pandas, Scikit-learn, Matplotlib, Seaborn, Flask ,Jupyter Notebook, Visual Studio Code and a few other libraries were used to build the whole model.

1. Google colab and Visual Studio code were used as IDE.
2. For visualization tasks, Matplotlib, Seaborn and plotly were used.
3. Flask were used for building the web application and server to run the code.
4. GitHub is used as version control system.
5. NumPy and Pandas were used to clean and interpret data.
6. Scikit learn was used to cross validate and compare different models.

2.7.1 Hardware Requirements

1. Windows Server, Linux, or any operating system that can run as a webserver, capable of delivering HTML5 content.
2. Minimum 1.10 GHz processor or equivalent.
3. Between 1-2 GB of free storage
4. Minimum 512 MB of RAM
5. 3 GB of hard-disk space

2.8 Constraints

The front-end must be user friendly and should not need any one to have any prior knowledge in order to use it.

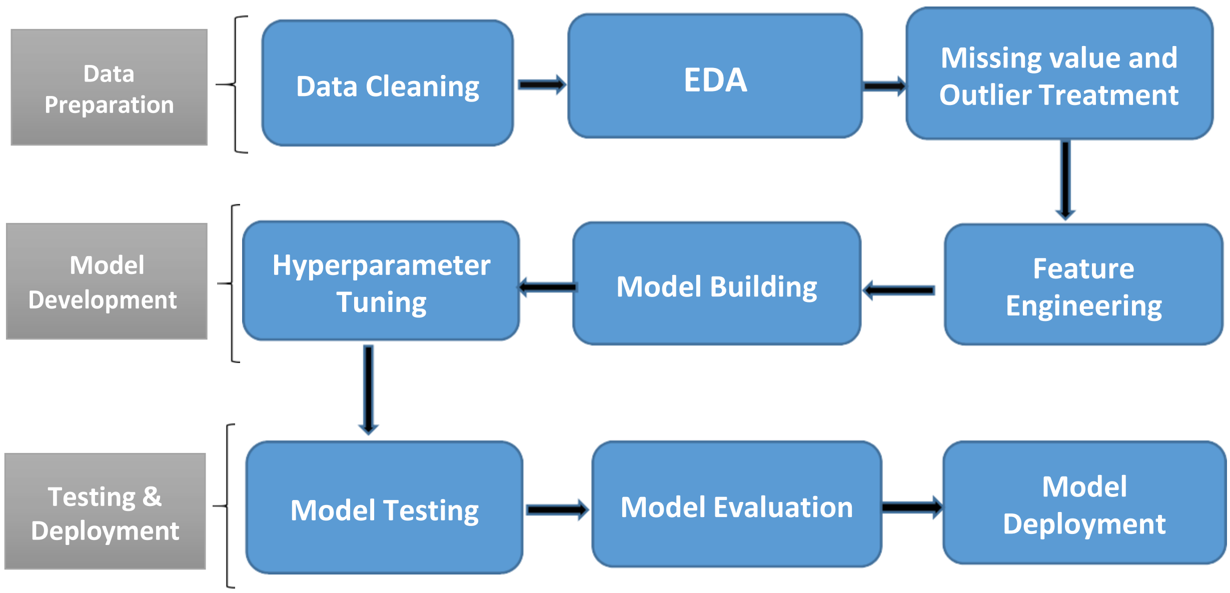
2.9 Assumptions

The main objective of this project is to implement the use case as previously mentioned (2.3 problem statement) for new dataset that comes through the UI. It is assumed that all aspects of this project have the ability to work together as the designer is expecting and also the data on which our model is trained is as correct as possible.

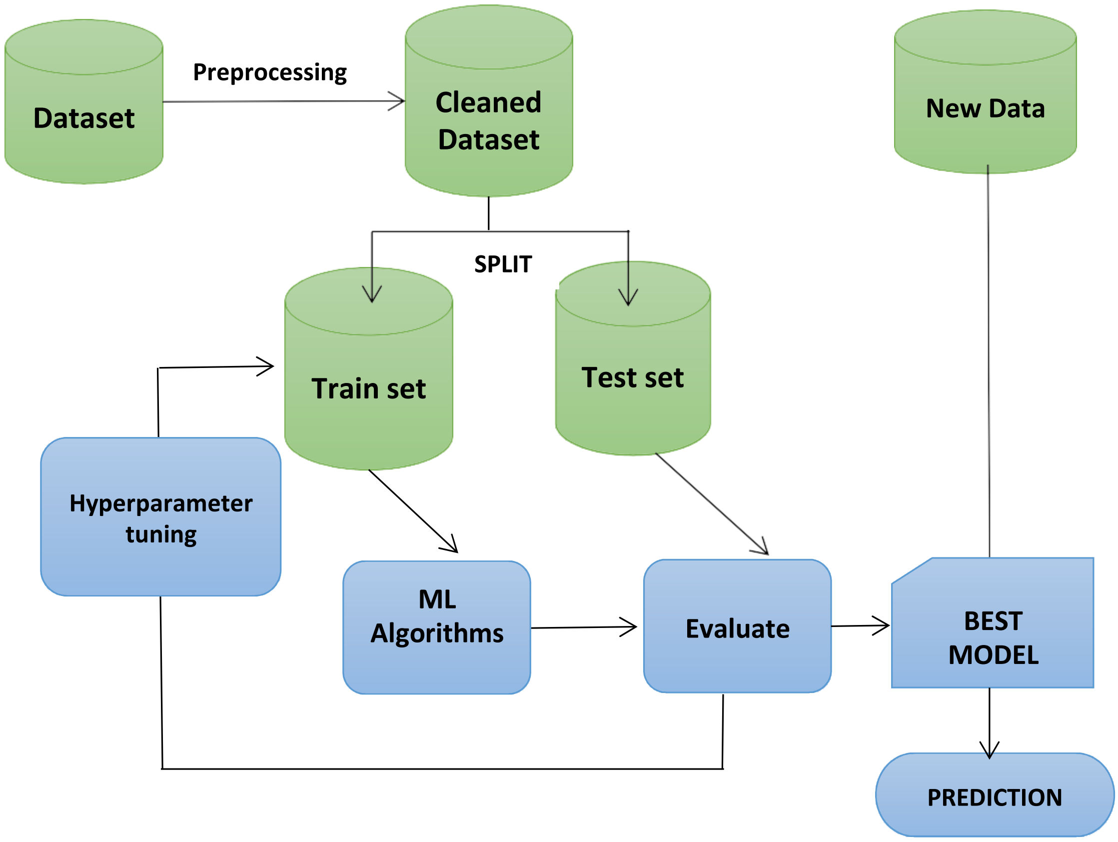
3. Design Details

3.1 Process Flow

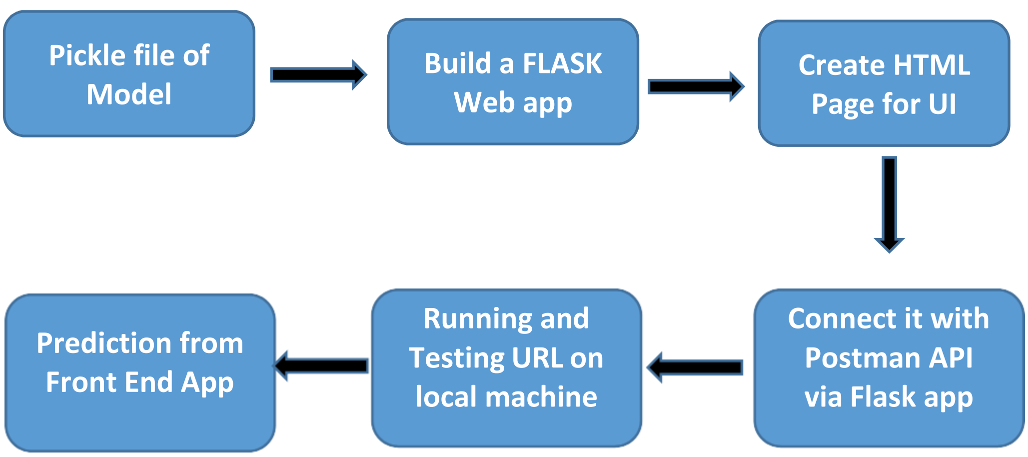
For accomplishment of the task, we will use a trained Machine Learning model. The process flow diagram is shown below :



3.1.1 Model Training and Evaluation



3.1.2 Deployment Process



3.2 Event Log

The system should log every event so that the user will know what process is running internally.
Initial step-by-step description :
1. The system identifies at what level logging is required
2. The system should be able to log each and every system flow
3. Developer can chose logging method. You can chose database logging/ File logging as well
4. System should not hang even after so many loggings. Logging just because we can easily debug issues so logging is mandatory to do

3.3 Error Handling

Errors should be encountered, an explanation will be displayed as to what went wrong ? An error will be defined as anything that falls outside the normal intended usage.

4 Performance

The SPP tool is used to predict whether a shipment consignment pricing. So this is made keeping in mind that if it will be used by Shipment corporations, Import-Export firms, private agencies then it is supposed to be as accurate as possible. So that it doesn't predict cost with high error. Also model retraining is very important to further enhance its performance.

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

4.2 Application Compatibility
The different components for this project will be using Python as an interface between them, Each component will have its own task to perform, and it is the job of Python to ensure proper transfer of information.

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

4.4 Deployment

5 Dashboards

Dashboards will be implemented to display and indicate certain KPIs and relevant indicators for the Supply Chain bottlenecks that if not addressed in time would cause demand-supply imbalance.

5.1 KPIs (Key Performance Indicators)

1. Key Performance Indicators of SPP.
2. Top destination countries.
3. Production volume wise Manufacturing Plant.
4. Tracking various Shipment Modes.

6 Conclusion

The SPP will give the Shipment Pricing predictions of a consignment instantly and has the potential to help various organization, agencies, companies, etc around the world for smoother supply chain management and to learn from the analytics.