Low Level Design (LLD)

Low Level Design (LLD)

BACK ORDER PREDICTION

Revision Number – 1.3

Last Date of Revision – 22/11/2022

Pichia Anil Naga Sai

1

Low Level Design (LLD)

Document Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Description | Author |
| 27-10-2022 | 1.0 | Abstract, | Naveen |
|  |  | Introduction |  |
|  |  | Architecture |  |
| 12-11-2022 | 1.1 | Data | Naveen |
|  |  | Preprocessing |  |
| 22-11-2022 | 1.2 | Deployment | Naveen |
|  |  | Unit Test |  |

2

Low Level Design (LLD)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Contents** | | | | | | | | | | | | | | | |  |  |  |  |
| [**Abstract**](https://docs.google.com/document/d/1ZLlZEVdMuxYl3o0PziQjZcUbNmAcz-PR/edit#heading=h.gjdgxs) | |  | | | | | | | | | | | | | | [4](https://docs.google.com/document/d/1ZLlZEVdMuxYl3o0PziQjZcUbNmAcz-PR/edit#heading=h.gjdgxs) | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| [**INTRODUCTION**](https://docs.google.com/document/d/1ZLlZEVdMuxYl3o0PziQjZcUbNmAcz-PR/edit#heading=h.30j0zll) | | | | |  | | | | | | | | | | | [5](https://docs.google.com/document/d/1ZLlZEVdMuxYl3o0PziQjZcUbNmAcz-PR/edit#heading=h.30j0zll) | | | |
|  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | [**Why this LLD documentation?**](https://docs.google.com/document/d/1ZLlZEVdMuxYl3o0PziQjZcUbNmAcz-PR/edit#heading=h.1fob9te) | | | | | | | | | | | | | | | [5](https://docs.google.com/document/d/1ZLlZEVdMuxYl3o0PziQjZcUbNmAcz-PR/edit#heading=h.1fob9te) | | |  |
|  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| [**1 Architecture**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.3znysh7) | | | | | | | | | | | | | | | | [**3**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.3znysh7) | | | |
|  |  | |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |
| [**2 Architecture Design**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.3dy6vkm) | | | | | | | |  | | | | | | | | [**6**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.3dy6vkm) | | | |
|  |  | | |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |
|  | [**2.1 Data Gathering from Main Source**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.1t3h5sf) | | | | | | | | | | | | | | | [**6**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.1t3h5sf) | | |  |
|  |  | | |  |  |  |  | |  |  |  |  | |  |  |  |  |  |  |
|  | [**2.2 Tools Used**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.17dp8vu) | | | |  | | | | | | | | | | |  | | [**6**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.17dp8vu) | |
|  |  | | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |
|  | [**2.3 Data Description**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.3rdcrjn) | | | | | | | | | | | | | | | [**7**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.3rdcrjn) | | |  |
|  |  | | |  |  |  |  | |  |  |  |  | |  |  |  |  | |  |
|  | [**2.4 Import Data into Database**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.26in1rg) | | | | | | | | | | | | |  | | [**7**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.26in1rg) | | |  |
|  |  | | |  |  |  |  | | |  |  |  | |  |  |  |  | |  |
|  | [**2.5 Export Data from Database**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.lnxbz9) | | | | | | | | | | | | | |  | [**8**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.lnxbz9) | | |  |
|  |  | | |  |  |  |  | | |  |  |  | | |  |  |  | |  |
|  | [**2.6 Data Pre-Processing**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.35nkun2) | | | | | | | | | |  | | | |  | [**8**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.35nkun2) | | |  |
|  |  | | |  |  |  |  | | |  |  |  | | | |  |  | |  |
|  | [**2.7 Modelling**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.1ksv4uv) | | | | | | | | | |  | | | | | [**8**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.1ksv4uv) | | | |
|  |  | | |  |  |  |  | | |  | |  | | | |  |  | |  |
|  | [**2.8 UI Integration**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.44sinio) | | | | |  | | | | | | | | | | [**8**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.44sinio) | | | |
|  |  | | | |  |  |  | | |  | |  | | | |  |  | |  |
|  | [**2.9 Data From User**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.2jxsxqh) | | | | | |  | | | | | | | | | [**8**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.2jxsxqh) | | | |
|  |  | | | |  | |  | | |  | |  | | | |  |  | |  |
|  | [**2.10 Data Validation**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.z337ya) | | | | | | | | |  | | | | | | [**8**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.z337ya) | | |  |
|  |  | | | |  | |  | | |  | |  | | | |  |  | |  |
|  | [**2.11 Rendering the Results**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.3j2qqm3) | | | | | | | | | | |  | | | | [**8**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.3j2qqm3) | | |  |
|  |  | | | |  | |  | | | | |  | | | |  |  | |  |
|  | [**3 Deployment**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.1y810tw) | | | |  | | | | | | |  | | | |  | [**8**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.1y810tw) | |  |
|  | [**3.1 Unit Test Cases**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.4i7ojhp) | | | | | |  | | | | | | | | | [**9**](https://docs.google.com/document/d/1CawpaB-mO_T9ew2gcRJPxTYNEbUemWHc/edit#heading=h.4i7ojhp) | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

3

Low Level Design (LLD)

**Abstract**

Inventory backorder prediction is widely recognized as an important component of inventory models. However, backorder prediction is traditionally based on stochastic approximation, thus neglecting the substantial amount of useful information hidden in historical inventory data. To provide those inventory models with a big data-driven backorder prediction, we propose a machine learning model equipped with an under sampling procedure to maximize the expected profit of backorder decisions. This is achieved by integrating the proposed profit-based measure into the prediction model and optimizing the decision threshold to identify the optimal backorder strategy. We show that the proposed inventory backorder prediction model shows better prediction and profit function performance than the state-of-the-art machine learning methods used for large imbalanced data. Notably, the proposed model is computationally effective and robust to variation in both warehousing/inventory cost and sales margin. In addition, the model predicts both major (non-backorder items) and minor (backorder items) classes in a benchmark dataset.

4

Low Level Design (LLD)

**1 Introduction**

**1.1 Why this Low-Level Design Document?**

The main purpose of this LLD documentation is to feature the required details of the project and supply the outline of the machine learning model and also the written code. This additionally provides the careful description on however the complete project has been designed end-to-end.

**1.2 Architecture**

PYTHON

SQL SERVER

Data (CSV)

EDA

DATA PREPROCESSING

IMPORT PYTHON LIBRARIES & READ DATA

FEATURE SELECTION

FITTING MODEL WITH LIGHT GBM

FEATURE ENGINEERING

CREATING A WEB PAGE FOR DEPLOYMENT

SAVE MODEL IN PICKLE FILE

HYPER PARAMETER TUNNING

EXPORT DATA BACK TO SQL SERVER

DEPLOYMENT ON LOCAL HOST USING FLASK AND VS CODE

DEPLOYMENT ON HEROKU

CREATING POWER BI REPORT

INPUT VALUE & PREDICT FINAL RESULT

ARCHITECTURE

**2. Architecture Design**

This project is to make associate interface for the user to grasp their approximate Back Order Prediction worth, additionally to the present, it would like of obtaining the important time project expertise we have a tendency to square measure mercantilism the gathered information into our own information then begin the project from the scratch.

5

Low Level Design (LLD)

**2.1. Data Gathering**

The data for the current project is being gathered from Kaggle dataset, the link to the data is:

https://www.kaggle.com/competitions/untadta/data

**2.2. Tool Used**

• Python 3.9 is employed because the programming language and frame works like numpy, pandas, sklearn and alternative modules for building the model.

* Visual Studio Code is employed as IDE.
* For visualizations seaborn and components of matplotlib are getting used
* For information assortment prophetess info is getting used version

management.

• Heroku is employed for deployment

* SQL SERVER IS USED FOR DATABASE.
* Power Bi is used for creating a report.

**2.3 Data Description**

There are about 1m+ records of sales information such as sku,national\_inv,lead\_time,min\_bank,forecast\_3\_month,sales\_3,6,9\_months etc.

HLD

**2.4 Import Data into Database**

* Created associate api for the transfer of the info into the SQL SERVER info, steps performed are:
* Connection is created with the info.
* Created a info with name back\_order.
* Create command is written for making the info table with needed parameters.
* And finally, a insert command is written for uploading the knowledgeset into data table by bulk insertion.

**2.5 Export Data into Database**

In the above created api, the download url is also being created, which downloads the data into a csv

file format.

**2.6 Data Preprocessing**

Steps performed in pre-processing are:

* First the info sorts square measure being checked and located solely the value column is of sort number.
* Checked for null values as there square measure few null values, those rows square measure born.
* Converted all the desired column into the date time format.
* Performed one-hot cryptography for the desired columns.
* Scaling is performed for needed information.
* And, the info is prepared for passing to the machine learning formula

7

Low Level Design (LLD)

**2.7 Modelling**

The pre-processed information is then envisioned and everywhere the specified insights are being drawn. Though from the drawn insights, the info is at random unfold however still modelling is performed with completely different machine learning algorithms to form positive we tend to cowl all the chances and eventually, for sure random forest regression performed well and any hyperparameter calibration is finished to extend the model’s accuracy.

**2.8 UI Integration**

Both CSS and HTML files are being created and are being integrated with the created machine learning model. All the required files are then integrated to the app.py file and tested locally

**2.3 Data from User**

The data from the user is retrieved from the created HTML web page.

**2.4 Data Validation**

The data provided by the user is then being processed by app.py file and validated. The validated data is then sent for the prediction.

**2.11 Rendering Result**

The data sent for the prediction is then rendered to the web page. And predicted result will be added in new table and export to sql server and the powerbi report is created through making connection of sql server with powerbi.

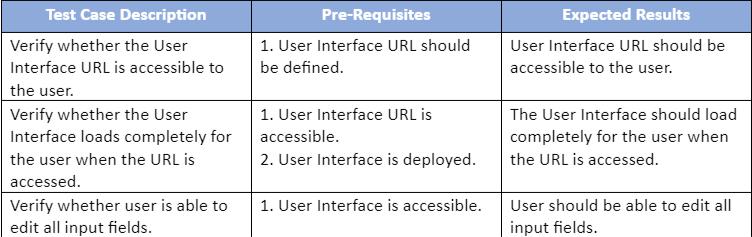
**3. Deployment**

The tested model is then deployed to Heroku. So, users can access the project from any internet devices.

8

Low Level Design (LLD)

**3.1 Unit Test**



9