High Level Design (HLD)

High Level Design (HLD)

BACK ORDER PREDICTION

Revision Number – 1.3

Last Date of Revision – 28/03/2022

Naveen Gupta

(Document Version Control

HIGH LEVEL DESIGN (HLD)

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Description | Author |
| 24-03-2022 | 1.0 | Abstract, | Naveen |
|  |  | Introduction |  |
|  |  | Problem |  |
|  |  | Statement |  |
| 23-03-2022 | 1.1 | Design Flow | Naveen |
| 22-03-2022 | 1.2 | Performance | Naveen |
|  |  | Evaluation |  |

HLD)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Contents** | | | | | | | | | | | | | | | | | | | | | |
| [**Abstract**](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) | | | | | | | | | |  | | | | | | | | | | | |
|  |  |  |  | |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | [**Why this HLD documentation?**](https://docs.google.com/document/d/1ZLlZEVdMuxYl3o0PziQjZcUbNmAcz-PR/edit#heading=h.1fob9te) | | | | | | | | | | | | | | | | | | | | |
|  |  |  |  | |  | |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |
| [**1 Description**](https://docs.google.com/document/d/1ZLlZEVdMuxYl3o0PziQjZcUbNmAcz-PR/edit#heading=h.2et92p0) | | | | | | | |  | | | | | | | | | | | | | |
|  |  | |  |  |  | |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |
|  | [**1.1**](https://docs.google.com/document/d/1ZLlZEVdMuxYl3o0PziQjZcUbNmAcz-PR/edit#heading=h.tyjcwt) | |  |  | [**Problem Perspective**](https://docs.google.com/document/d/1ZLlZEVdMuxYl3o0PziQjZcUbNmAcz-PR/edit#heading=h.tyjcwt) | | | | | | | | | | | | | | | | |
|  |  | |  | |  | |  | |  | |  |  |  |  |  |  |  |  |  |  |  |
|  | [**1.2 Problem Statement**](https://docs.google.com/document/d/1ZLlZEVdMuxYl3o0PziQjZcUbNmAcz-PR/edit#heading=h.3dy6vkm) | | | | | | | | | | | | | | |  | | | | | |
|  |  | | | | | |  | |  | |  |  |  |  |  |  |  | |  |  |  |
|  | [**1.3 Proposed Solution**](https://docs.google.com/document/d/1ZLlZEVdMuxYl3o0PziQjZcUbNmAcz-PR/edit#heading=h.1t3h5sf) | | | | | | | | | | | | | | |  | | | | | |
|  |  | | | | | |  | |  | |  |  |  |  |  |  |  | |  |  |  |
|  | [**1.4 Solution Improvements**](https://docs.google.com/document/d/1ZLlZEVdMuxYl3o0PziQjZcUbNmAcz-PR/edit#heading=h.4d34og8) | | | | | | | | | | | | | | | | | | | | |
|  |  | | | | | |  | |  | |  |  |  |  |  |  |  | |  |  |  |
|  | [**1.5 Technical Requirements**](https://docs.google.com/document/d/1ZLlZEVdMuxYl3o0PziQjZcUbNmAcz-PR/edit#heading=h.2s8eyo1) | | | | | | | | | | | | | | | | | | |  | |
|  | [**1.6 Data Requirements**](https://docs.google.com/document/d/1ZLlZEVdMuxYl3o0PziQjZcUbNmAcz-PR/edit#heading=h.17dp8vu) | | | | | | | | | | | | | | |  | | | | | |
|  |  | | | | | |  | |  | |  |  |  |  |  |  |  | | | |  |
|  | [**1.7 Tools Used**](https://docs.google.com/document/d/1ZLlZEVdMuxYl3o0PziQjZcUbNmAcz-PR/edit#heading=h.3rdcrjn) | | | | | | | | | |  | | | | |  | | | | | |
|  |  | |  |  | | |  | |  | |  |  |  |  |  | |  | | | |  |
|  | [**1.8**](https://docs.google.com/document/d/1ZLlZEVdMuxYl3o0PziQjZcUbNmAcz-PR/edit#heading=h.26in1rg) | |  | [**Constraints**](https://docs.google.com/document/d/1ZLlZEVdMuxYl3o0PziQjZcUbNmAcz-PR/edit#heading=h.26in1rg) | | | | | | | |  | | | | | | | | | |
|  |  | | | | | |  | |  | | |  |  |  |  | |  | | | |  |
|  | [**1.9 Assumptions**](https://docs.google.com/document/d/1ZLlZEVdMuxYl3o0PziQjZcUbNmAcz-PR/edit#heading=h.lnxbz9) | | | | | | | | | | | | | | | | | | | | |
|  | |  | | | | |  | |  | | |  |  |  |  | |  | | | |  |
| [**2 Design Flow**](https://docs.google.com/document/d/1ZLlZEVdMuxYl3o0PziQjZcUbNmAcz-PR/edit#heading=h.35nkun2) | | | | | | | | |  | | | | | | | | | | | | |
|  |  | | | | | |  | |  | | |  |  | |  | |  | | | |  |
|  | [**2.1 Modelling Process**](https://docs.google.com/document/d/1ZLlZEVdMuxYl3o0PziQjZcUbNmAcz-PR/edit#heading=h.1ksv4uv) | | | | | | | | | | | | | | | | | | | | |
|  |  | | | | | |  | | | | |  |  | |  | |  | | | |  |
|  | [**2.2 Deployment Process**](https://docs.google.com/document/d/1ZLlZEVdMuxYl3o0PziQjZcUbNmAcz-PR/edit#heading=h.44sinio) | | | | | | | | | | | | | | | | | | | | |
|  | |  | |  | | |  | | | | |  |  | | | |  | | | |  |
| **2.3** | | | | **Logging** | | | | | | | | | | | | | | | | | |
| **2.2** | | | | **Error Handling** | | | | | | | | | | | | | | | | | |
| [**3 Performance Evaluation**](https://docs.google.com/document/d/1ZLlZEVdMuxYl3o0PziQjZcUbNmAcz-PR/edit#heading=h.4i7ojhp) | | | | | | | | | | | | | | | | |  | | | | |
|  | | |  |  | | |  | | | | |  |  | | | |  | | | |  |
|  | [**3.1**](https://docs.google.com/document/d/1ZLlZEVdMuxYl3o0PziQjZcUbNmAcz-PR/edit#heading=h.2xcytpi) | |  | [**Reusability**](https://docs.google.com/document/d/1ZLlZEVdMuxYl3o0PziQjZcUbNmAcz-PR/edit#heading=h.2xcytpi) | | | | | | | |  | | | | |  | | | | |
|  |  | |  |  | | |  | | | | |  |  | | | |  | | | |  |
|  | [**3.2**](https://docs.google.com/document/d/1ZLlZEVdMuxYl3o0PziQjZcUbNmAcz-PR/edit#heading=h.1ci93xb) | |  | [**Application Compatibility**](https://docs.google.com/document/d/1ZLlZEVdMuxYl3o0PziQjZcUbNmAcz-PR/edit#heading=h.1ci93xb) | | | | | | | | | | | | | | | | | |
|  | [**3.3**](https://docs.google.com/document/d/1ZLlZEVdMuxYl3o0PziQjZcUbNmAcz-PR/edit#heading=h.3whwml4) | |  | [**Resource Utilization**](https://docs.google.com/document/d/1ZLlZEVdMuxYl3o0PziQjZcUbNmAcz-PR/edit#heading=h.3whwml4) | | | | | | | | | | | | |  | | | |  |
|  |  | | | | | |  | | | | | |  | | | |  | | | | |
|  | [**3.2 Deployment**](https://docs.google.com/document/d/1ZLlZEVdMuxYl3o0PziQjZcUbNmAcz-PR/edit#heading=h.2bn6wsx) | | | | | | | | | | | | | | | |  | | | | |
|  |  | | | | | |  | | | | | |  | | | | | | | | |
| [**Conclusion**](https://docs.google.com/document/d/1ZLlZEVdMuxYl3o0PziQjZcUbNmAcz-PR/edit#heading=h.qsh70q) | | | | | | | | | | | | | | | | | | | | | |

**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.

**1 Introduction**

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

The main purpose of this HLD 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 Description**

**Problem Perspective**

The Back Order prediction may be a machine learning model that helps Stores to predict the back order of the store’s products and helps the users to manage the inventory of the business product’s.

**1.3 Problem Statement**

Backorders are unavoidable, but by anticipating which things will be backordered, planning can be streamlined at several levels, preventing unexpected strain on production, logistics, and transportation. ERP systems generate a lot of data (mainly structured) and also contain a lot of historical data; if this data can be properly utilized, a predictive model to forecast backorders and plan accordingly can be constructed. Based on past data from inventories, supply chain, and sales, classify the products as going into backorder (Yes or No).

**1.4. Project Solution**

Project requires the desired input of user from the created interface and method all the provided information to satisfy the wants of the machine learning model and at last show the expected output .

**1.5 Answer enhancements**

We will even predict that how much order of a customer can be gone in a backorder and it will more help for the business to manage the inventory properly and having an advance inventory management in hand for the customer’s order to deliver it on a time.

**1.6 Technical needs**

There are not any hardware needs needed for victimization this application, the user should have AN interactive device that has access to the web and should have the fundamental understanding of providing the input. And for the backend half the server should run all the package that's needed for the process and provided information to show the results.

**1.7 Information needs**

The info demand is totally supported the matter statement. and also, the information set is accessible on the Kaggle within the type of standout sheet(.xlsx), because the main theme of the project is to induce the expertise of real time issues, we have a tendency to once more mercantilism {the information into the prophetess data base and commerce it into csv format.

**1.8Tools 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.
* Front end development is completed victimization HTML/CSS.
* Flask is employed for each information and backend readying.
* GitHub is employed for version management.
* Heroku is employed for deployment.
* SQL SERVER IS USED FOR DATABASE.
* Power Bi is used for creating a report.

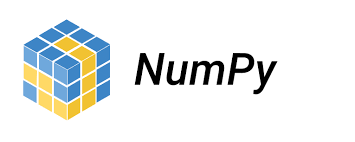
**1.9 Constraints**

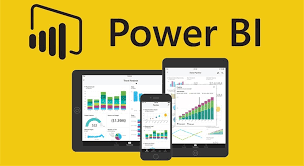
The Back Order Prediction answer should be user friendly, as automatic as attainable and also the user should not be needed to understand any of the operating.

**1.10 Assumptions**

The most objective of the project is to implement the utilization cases as for the new dataset that user provides through the programme. Machine learning model is employed for process the on top of computer file. It's additionally assumed that each one aspects of this project have the flexibility to figure along within the approach as the designer is expecting.





**2.1 and 2.2 Design Flow and Deployment Process**

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

ARCHITEC

**2.3 Logging**

Each step is being logged within the system that runs internally, that shows the date time and therefore the processed that has been performed, work is completed in several layers as information, DEBUG, ERROR, WARNINGS. this provides US the perceive of the logged info.

**2.4 Error Handling**

Once ever a slip is occurred, the reason are logged in its several log file, in order that the developer will rectify the error.

ARCHITECTURE

**3 Performance analysis**

**3.1 Reusability**

Elements of the code written is accustomed different applications and therefore the rest is changed and be reused.

**3.2 Application Compatibility**

The various parts for this project are exploitation python as associate interface between them. Every element can have its own tasks to perform, and it's the work of the python to make sure correct transfer of data.

**3.3 Resource Utilization**

Once any task is performed, it'll doubtless; use all the process power offered till that performs is finished.

**3.4 Deployment**

The model is being deployed on Heroku.

**Conclusion**

The back order prediction will predict the worth supported the trained knowledge set within the rule. Therefore, the user will recognize the approximate result for his or her product.