High-Level Design (HLD)

Backorder Prediction

Table of Content

Abstract

There are several steps between the time a product is manufactured and the time it reaches the customer, including production, logistics, and transportation. Occasionally due to a communication breakdown or for any other reason, a retailer may order a product that the supplier is out of stock, which can put an unexpected kink in the entire process. This article outlines the deployment of a system that uses various data produced by ERP (Enterprise Resource Planning) to forecast where a product may go on backorder. and make plans for it properly. Identify the goods as being back ordered based on historical inventory, supply chain, and sales data (Yes or No).

1. Introduction

1.1 why this High-Level Design Document?

This High-Level Design (HLD) Document aims 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.

This HLD will:

- Present all of the design aspects and define them in detail
- Describe the user interface is implemented
- Describe the hardware and software interfaces
- 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 system's structure, 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 system's administrators.

1.3 Definitions

Term	Description
ML	Machine Learning
DL	Deep Learning
IDE	Integrated Development Environment

2 General Description

2.1 Product Perspective

The Backorder Prediction system is a Machine Learning based model which helps us to estimate whether a product can go in backorder or not

2.2 Problem Statement

To create an ML solution for a predictive model to forecast backorders and plan accordingly can be constructed.

2.3 Proposed Solution

An ML-based application that can resolve the problem statement is proposed to assist in this case. In order to do so, we tested a number of ML algorithms on the supplied data, and based on a defined parameter, the algorithm that can fulfill the parameter is chosen and implemented in the application so that users may make predictions. Additionally, the user may add any new algorithm that wasn't previously examined and contrast it with current algorithms by changing the Model Config. If the results are better, the new algorithm is used.

2.4 Further Improvements

The following improvements can be made to the system:

- Along with the ML algorithm DL algorithm can also be used
- Deploying the solution on a faster system so that the algorithm can be selected much quicker
- Improving the data distribution better the target classes

2.5 Technical Requirements

This Document addresses the requirement to access the project. You need a System with a decent configuration.

Minimum Requirement:

- CPU: Intel Core i3-3210 3.2 GHz / AMD A8-7600 APU 3.1 GHz or equivalent
- CPU SPEED: 1 gigahertz(GHz) or faster processor or System on a Chip(SoC)
- RAM: 2 GB
- OS: Windows 7 and up
- FREE DISK SPACE: At least 1 GB
- A decent Internet Connection

2.6 Data Requirements

Data requirement completely depend on our problem statement. We need data from the ERP system mostly structured data in form of rows and columns. The data is based on past data from inventories, supply chains, and sales. 2 separate files must be provided 1 each for training and testing

2.7 Tools Used

Python programming language and frameworks such as NumPy, Pandas, and Scikit-Learn are used to build the whole model

- VS Code is used as IDE
- Evidently is used for checking Data Drift
- Docker is used to create a container of the whole project
- Heroku is used to Host deploy the docker container
- Front-end development is done using HTML/CSS
- Flask is used for Backend Development
- GitHub is used as a version control system

2.8 Constraints

The Application must be user friendly, as automated as possible and the user should not be required to know any of the workings

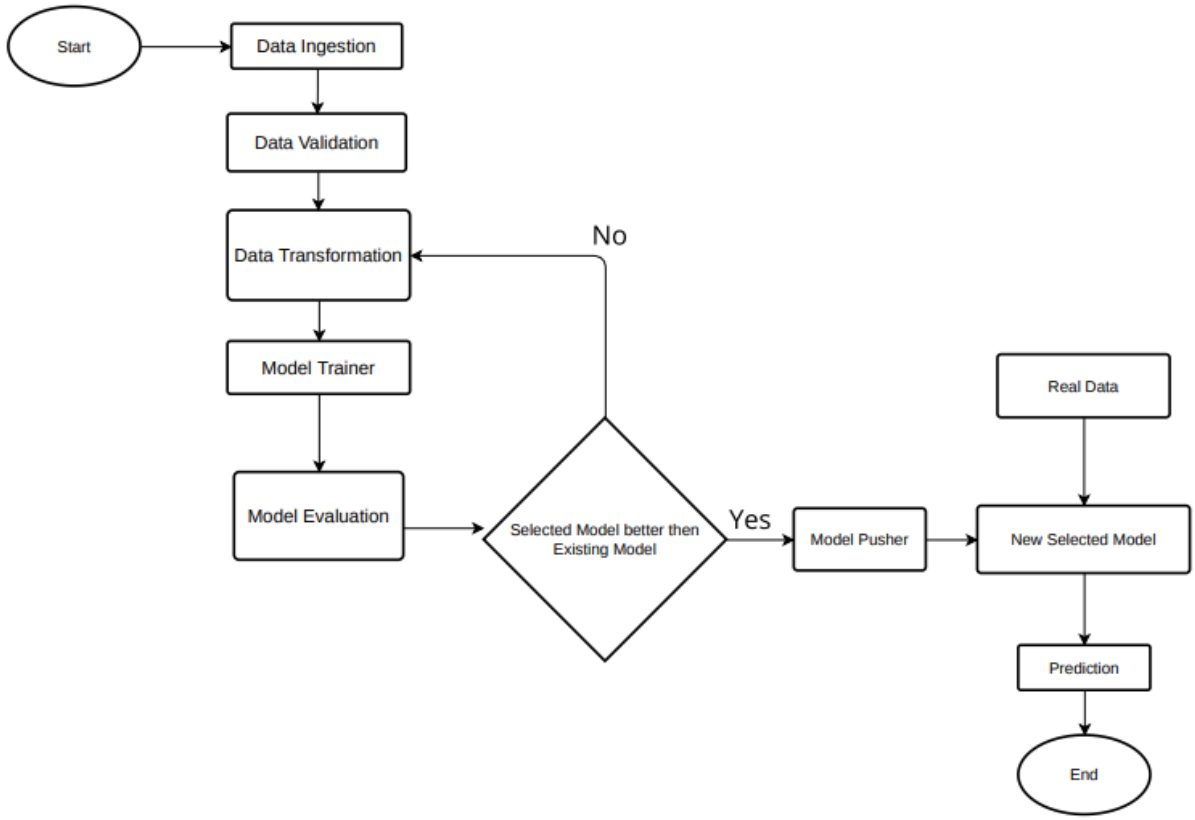
2.9 Assumptions

The Main objective of the project is to implement the use case as previously mentioned (2.2 Problem Statement) for the new dataset that comes through. Machine Learning based model is used for predicting the above-mentioned use cases based on the input data. It is also assumed that all aspects of this project have the ability to work together in the way the designer is expecting

3 Design Details

3.1 Process Flow

For identifying the different types of anomalies, we will an ML-based model. Below is the process flow diagram



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 step logging is required
2. The System should be able to log each and every system flow.
3. Developer can choose the logging method. You can choose database logging/ File logging as well.
4. System should not hang even after using so many loggings. Logging is just because we can easily debug issues so logging is mandatory to do.

3.3 Error Handling

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

3.4 Performance

The Model should take able to make quick predictions in a few seconds

3.5 Reusability

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

3.6 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 the proper transfer of information.

3.7 Resource Utilization

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

3.8 Deployment

This project has been dockerized so it can be deployed anywhere like

4 KPIs (Key Performance Indicators)

Key indicators displays a summary of the backorder prediction in the retail/supply chain.

- Better preparation for a product supply chain
- Making a correct prediction on backorder products
- Having an easy-to-understand and interactive interface for the customers

5 Conclusion

The Application will predict the backorder on a product based on ERP parameters which contain a lot of historical data. like inventories, supply chains, and sales. So that we can plan accordingly can be constructed.to whether order more of that product and