High Level Design

BACK ORDER PREDICTION

WRITTEN BY M.SHABARISH

Table of Contents

Abstract

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. Material backorder is a common supply chain problem, impacting an inventory system service level and effectiveness. Identifying parts with the highest chances of shortage prior its occurrence can present a high opportunity to improve an overall company's performance. 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). The Problem is to prevent the unexpected strain on production, logistics and transportation. From then problem statement we need reduce the both False Positive and False Negative. False Negative reduction will help us to prevent the unexpected strain on production, logistics and transportation false positive is not that much important to the company because even if it is predicted as product is going to be backordered but actually is not backordered. The macro-average is not recommended since it gives equal weight to both classes, which is not desirable when dealing with imbalanced data. The weighted average and micro-average metrics give more weight to the minority class, which is important when evaluating the classifier's performance on imbalanced data.

1. Introduction

1.1 Why this HLD document?

A High-Level Design (HLD) document is a technical document that describes the architecture of a software system or application at a high level. The purpose of an HLD document is to provide a blueprint for developers and stakeholders to understand the high-level technical details of the system, including the architecture, components, modules, interfaces, and other high-level design details.

The HLD document has several purposes:

- Establishing the system's architecture.
- Defining the components and modules.
- Providing a common understanding.
- Facilitating communication and collaboration.
- Guiding the low-level design.
- List and describe the non - functional attributes like:
  1. Security
  2. Reliability
  3. Maintainability
  4. Portability
  5. Reusability
  6. Resource
  7. Utilization

1.2 Scope

An HLD document is an essential document that helps ensure the successful development of a software system or application by providing a clear and high-level technical specification. The HLD document is often used as a starting point for the development process and serves as a foundation for the more detailed low-level design documents.

1.3 Definitions

TERM	DESCRIPTION
BO	Backorder
S3	Amazon Simple Storage Service (Amazon S3) is an object storage service
IDE	Integrated Development Enviroment.
Stream lit	Streamlit’s open-source app framework
VS Code	VS Code stands for Visual Studio code and it is an IDE.

2. General Description

2.1 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. Material backorder is a common supply chain problem, impacting an inventory system service level and effectiveness. Identifying parts with the highest chances of shortage prior its occurrence can present a high opportunity to improve an overall company's performance.

2.2 Proposed Solution

Machine learning models can provide more accurate results than traditional methods, as they can learn from data and improve over time. To predict the unusual backorders to reduce the strain on the logistics proposing an Machine Learning is very robust to predict the backorders. Backorders are predicted using several features like lead time, forecast of sales and performance of sales for prior months, quantity of the sales and based on some risk factors.

2.3 Further Improvements

Backorder use case can be improved by adding several features like intelligent estimation of the quantity of the sales to avoid backorders, based the available inventory we can add features like estimation of workers required to process the orders.
Continuous monitoring and improvement of the model can help ensure that it remains accurate and relevant over time. Techniques such as model retraining and updating can be applied to keep the model up-to-date with the latest data and trends.

2.4 Technical Requirements

As the project requirements we don't need any hardware requirements.
In Backend we use streamlit to deploy our application.

2.5 Data Requirements

The dataset is available as a raw file in Github. To reduce the latency saving the dataset as a parquet file in Amazon s3 buckets.
To process the data we need the data in the form of (.csv) file

2.6 Tools Used

- Python 3.8 is used while creating the environment.
- Amazon S3 buckets are used to store the data.
- VS Code is used as IDE.
- GitHub is used as code repository.
- Streamlit application is used to host the application.

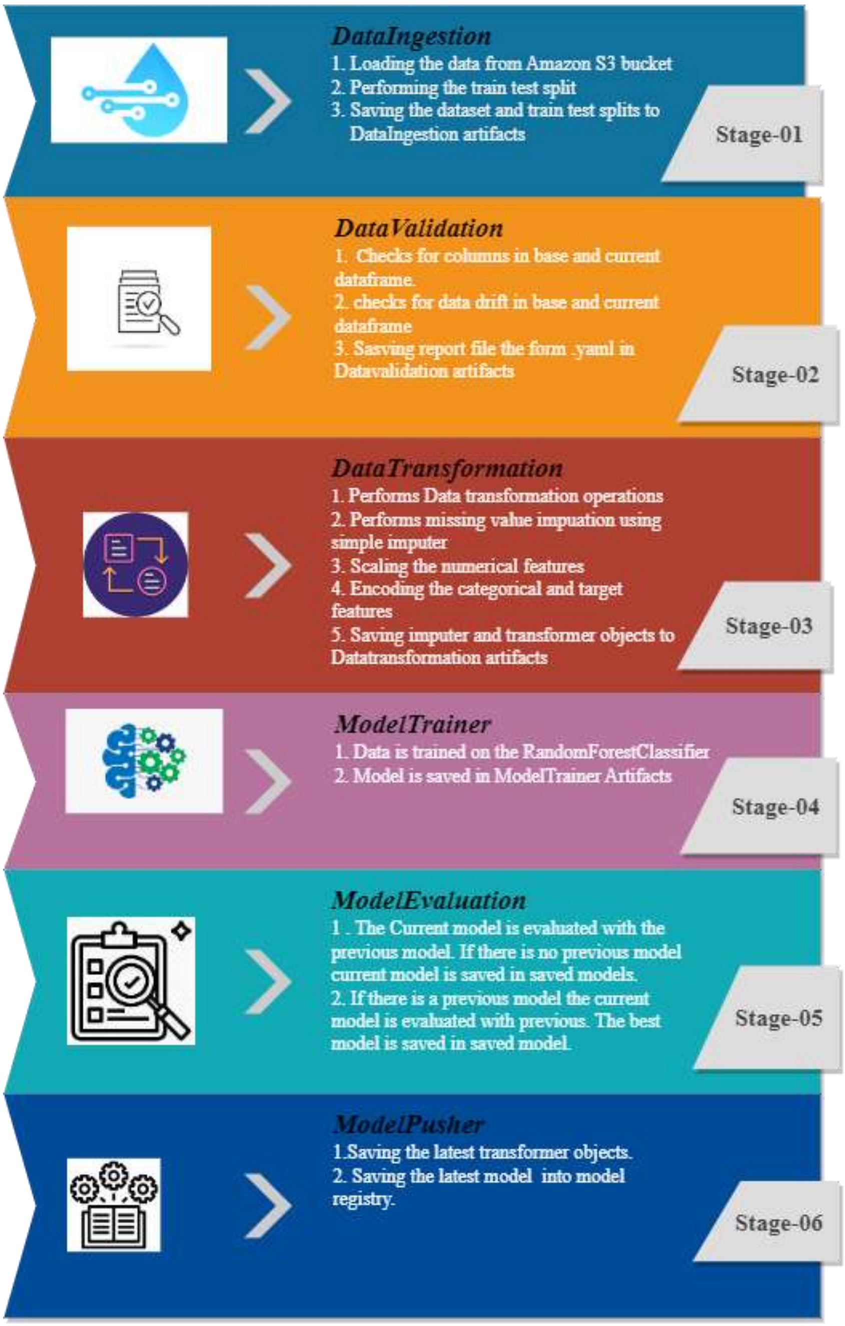
And other libraries like Pandas, Numpy, Matplotlib, Seaborn, Scikit-Learn are used to build the whole model.

2.7 Assumptions

The main objective of the project is to implement a classification model. It is assumed that all functionalities are working as it is expected.

3. Design Flow

3.1.1 Machine Learning Pipeline stages



DataIngestion

1. Loading the data from Amazon S3 bucket
2. Performing the train test split
3. Saving the dataset and train test splits to DataIngestion artifacts

Stage-01

DataValidation

1. Checks for columns in base and current dataframe.
2. checks for data drift in base and current dataframe
3. Sasving report file the form .yaml in Datavalidation artifacts

Stage-02

DataTransformation

1. Performs Data transformation operations
2. Performs missing value impuation using simple imputer
3. Scaling the numerical features
4. Encoding the categorical and target features
5. Saving imputer and transformer objects to Datatransformation artifacts

Stage-03

ModelTrainer

1. Data is trained on the RandomForestClassifier
2. Model is saved in ModelTrainer Artifacts

Stage-04

ModelEvaluation

1 . The Current model is evaluated with the previous model. If there is no previous model current model is saved in saved models.
2. If there is a previous model the current model is evaluated with previous. The best model is saved in saved model.

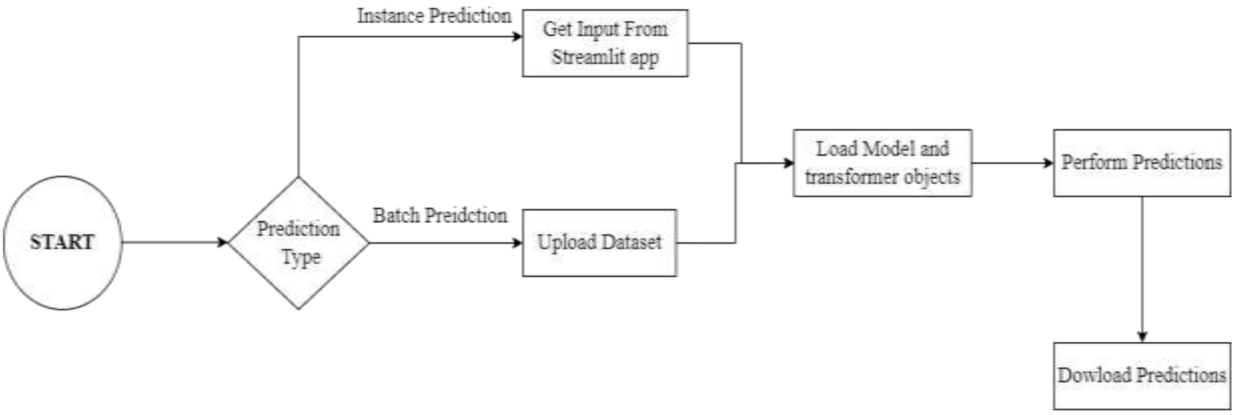
Stage-05

ModelPusher

1.Saving the latest transformer objects.
2. Saving the latest model into model registry.

Stage-06

3.1.2 Deployment Process



3.2 Logging

All the execution process is logged into a log with timestamps. This will help to find the error very quickly.

3.3 Error Handling

The Exception handling is implemented to find the errors quickly and can easily be detected any wrong as has happened.

4 . Performance

4.1 Reusability

The code has be written as per the Industry standards and can be reused without any interruption.

4.2 Application compatability

The various components of this project is built using python programming language. Each component has it own task to perform according to the pipeline.

4.3 Deployment

This Model is deployed using streamlit application.