

MACHINE LEARNING PROJECT

HLD Report On BACKORDER PREDICTION

Made by: Gaurav Singh
Domain: E-commerce

TABLE OF CONTENTS

1. Introduction.....	5
What is Low-Level design document?.....	5
Scope.....	5
Constraints... ..	6
2. General Description.....	7
Product Prospective.....	7
Problem Statement.....	7
Proposed Solution... ..	7
Technical Requirements.....	8
Data Requirements.....	8
Tool Used... ..	9
Constraints... ..	9
3. Design Flow	10
Process Flow	10
Prediction Flow.....	5
Event Log... ..	11
Error Handling	11
Performance.....	11
4. Re-usablity.....	12
5. Resource Utilization.....	12
6. Deployment.....	12

7. User Interface.	12
8. Conclusion.	14

ABSTRACT

Backorder is an order which can't be fulfilled at the given time due to lack of supply or the product is currently out of stock or not in inventory but can guarantee delivery of the goods or service requested by a certain date in the future because the production of goods or replenishment of inventory is underway. Unlike in the situation of Out-of-stock where the delivery date of the goods can't be promised, in the Backorder scenario the customers are allowed to shop for the products and order. Simply put Backorder can be thought of as an order with a delayed delivery date.

It 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.

1. Introduction

This document will be used for documenting High-level designs of project.

The HLD stands for **High-Level Design**, where the designer will only focus on the various models, like:

- **Decision Tables**
- **Decision Trees**
- **Flow Diagrams**
- **Flow Charts**
- **Data Dictionary**

Purpose of the Document

The solution architect develops the High-level design, which is used to specifies the complete description or architecture of the application.

The HLD involves **system architecture, database design, a brief description of systems, services, platforms, and relationships** among modules.

The HLD is also known as **macro-level or system design**. It changes the business or client requirement into a **High-Level Solution**.

The High-level design is created before the Low-Level Design.

Objective of HLD

1.HLD presents all of the design aspects (taken from business requirements and expected outcome) and defines them in form of a diagram.

2.It describe the user interface being implemented and description of hardware and software interfaces.

3.It describe the performance requirements and flow of user's daily process.

4.HLD includes design features and the architecture of the project.

Scope of HLD

The High-Level Design documentation presents the structure of the system as the application/database architecture, application flow and technology architecture. High-Level Design documentation may use some non-technical terms unlike Low Level design which should be strictly technical jargon.

2 General Description

Product Prospective

The Backorder prediction app is used to find prospective backorders to avoid any loss in sales or business.

Problem statement

To build a model which will be able to predict whether an order for a given product can go on backorder or not. A backorder is the order which could not be fulfilled by the company. Due to high demand of a product, the company was not able to keep up with the delivery of the order.

Proposed Solution

In this machine learning classifiers are investigated in order to propose a predictive model for this imbalanced class problem, where the relative frequency of items that goes into backorder is rare when compared to items that do not. Specific metrics such as area under the Receiver Operator Characteristic and precision-recall curves, sampling techniques and ensemble learning are employed in this particular task.

Technical Requirements

- Model should be exposed through API or User Interface, so that anyone can test model.
- Model should be deployed on cloud (Azure, AWS, GCP).
- Cassandra database should be integrated in this project for any kind of user input.

Data Requirements

Following Data fields are used after feature selection on which the model is trained and tuned. These fields are taken as an input in the app which tells you its backorder or not.

```
predictors = [  
    'forecast_6_month',  
    'forecast_9_month',  
    'sales_3_month',  
    'national_inv',  
    'sales_6_month',  
    'sales_9_month',  
    'perf_6_month_avg',  
    'perf_12_month_avg',  
    'forecast_3_month',  
    'in_transit_qty',  
    'potential_issue'  
]
```


Tools Used

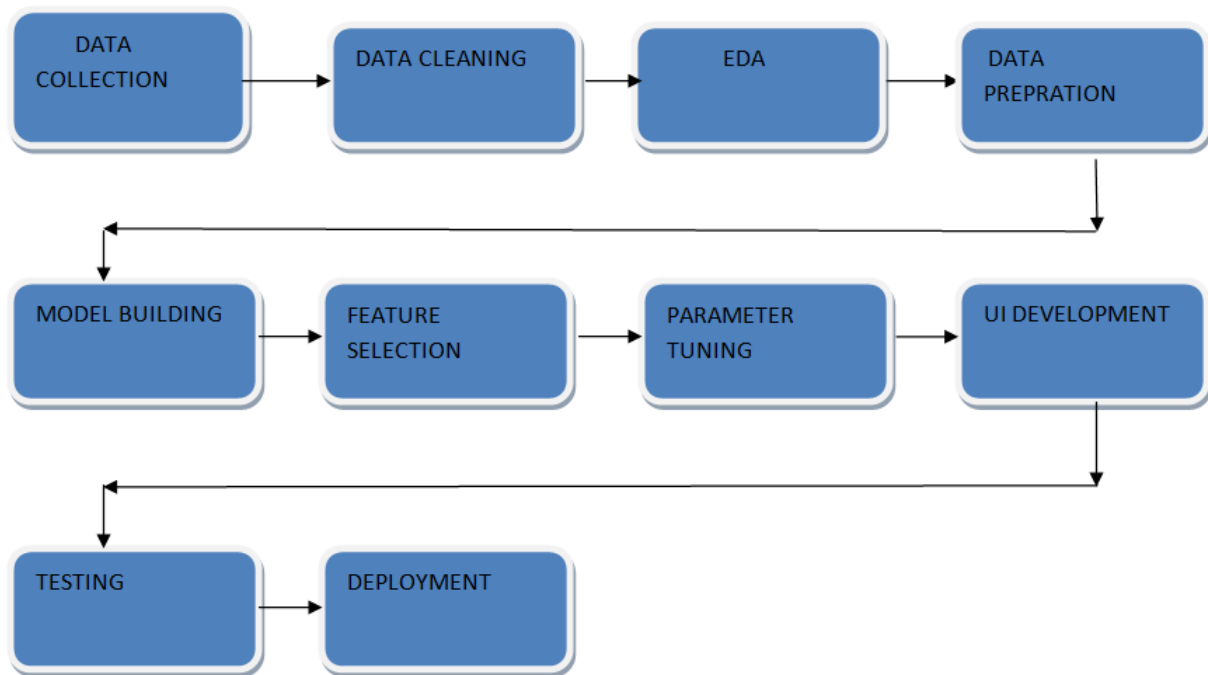
- Pycharm - IDE
- Visualization - Matplotlib , Seaborn
- Database - Cassandra
- Front End - HTML , CSS , JS
- Framework - Flask
- VCS - GIT
- Deployment - Heroku
- Data Wrangling - Numpy,Pandas
- Scikit-learn - ML Models
- Python

Constraints

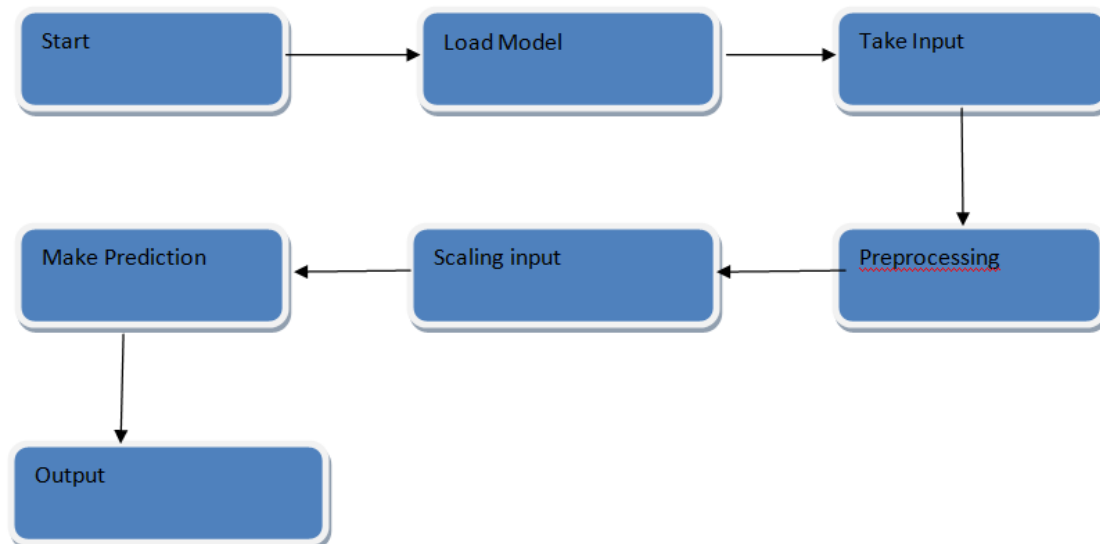
The Backorder prediction system must be user friendly, errors free and users should not be required to know any of the back end working.

3.Design Flow

3.1 Process flow



3.2 Prediction Process



3.3 Event Log

In this Project we are logging every process so that the user will know what process is running internally.

Initial Step-By-Step Description:

- In this Project we defined logging for every class.
- By logging we can monitor every insertion, every flow of data in database.
- By logging we are monitor every step which may create problem or every step which is important in file system.
- We have designed logging in such a way that system should not hang even after so many logging's, so that we can easily debug issues which may arises during process flow.

3.4 Error Handling / Exception Handling

We have added Exceptional Handling in a way to handle any issue which arises due to some function call.

3.5 Performance

Model	Hyperparameters	Best Value	Accuracy	Precision	Recall
Logistic Regression	eta0/penalty	[1e-05, 'l2']	0.845	0.916	0.761
Decision Tree	max_depth	6	0.879	0.902	0.851
Random Forest	n_estimators/max_depth	[75, 50]	0.906	0.909	0.903
Gradient Boosted Decision Tree	n_estimators/max_depth	[35, 20]	0.898	0.906	0.889

4. Re-usability

We have done programming of this project in such a way that it should be reusable. So that anyone can add and contribute without facing any problems.

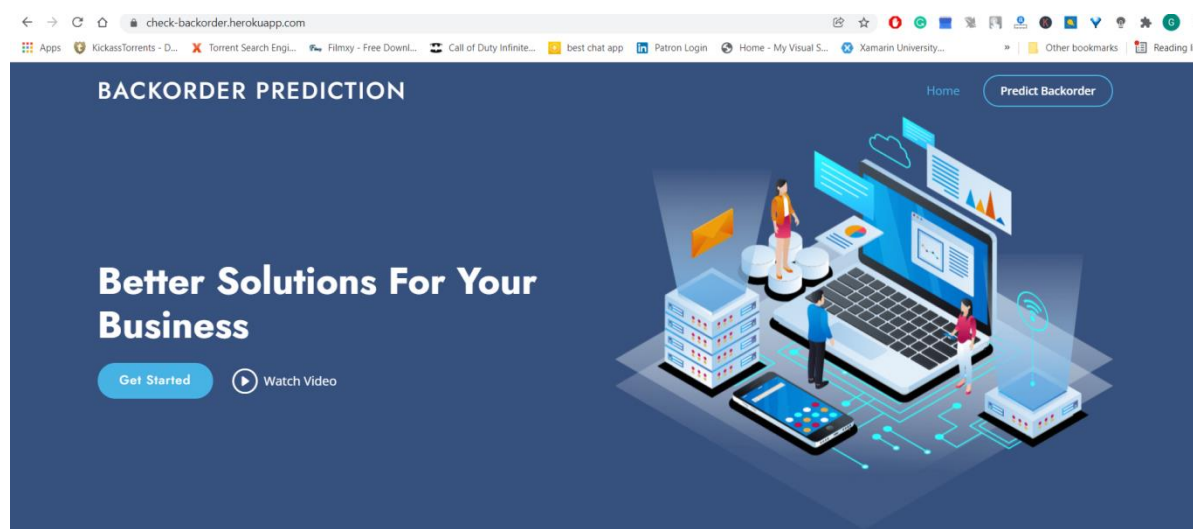
5.Resource Utilization

In this project, when any task is performed, it will likely that the task will use all the processing power available in that particular system until it's job finished. By keeping this in mind, In this project we have used the concept of multithreading.

6.Deployment

We have deployed this on Heroku Cloud.

7.User Interface



BACKORDER PREDICTION

Home Predict Backorder

Enter Backorder Prediction Details

Enter National Inventory

Enter Sales Qty For Last 3 Month Time

Enter Sales Qty For Last 6 Month Time

Enter Sales Qty For Last 9 Month Time

Enter Forecast Sales For Next 3 Months

Enter Forecast Sales For Next 6 Months

Enter Forecast Sales For Next 9 Months

Enter Source Avg Performance For Last 6 Month

Enter Source Avg Performance For Last 12 Month

Enter In Transit Qty

Enter if any potential_issue (0-No , 1-Yes)

BACKORDER PREDICTION

Back

Better Solutions For Your Business

Get Started

Watch Video

Result

Yes it's in Backorder

8. Conclusion

The Model which we have created will tell the owner in advance of the products which will be in backorder. It will help in not losing sales due to product unavailability.