

# Contents

Document Version Control  
Abstract

1. Introduction
  - 1.1 Why this High Level Design Document?
2. General Description
  - 2.1 Product Perspective
  - 2.2 Problem Statement
  - 2.3 Proposed Solution
  - 2.4 Technical Requirements
  - 2.5 Data Requirements
  - 2.6 Tools Used
  - 2.7 Constraints
3. Design Details
  - 3.1 Process Flow
  - 3.2 Deployment Process
  - 3.2 Event Log
  - 3.3 Error Handling
4. Performance
  - 4.1 Re-usability
  - 4.2 Application Compatibility
  - 4.3 Resource Utilization

High Level Design  
(HLD)

4.4 Deployment

4.5 User Interface

5. Conclusion

## **Document Version** **Control**

Date Issued	Version	Description	Author
6/09/2021	1	Initial HLD – V1.0.1	ChinmayMaganur

## Abstract

A backorder is an order for a good or service that cannot be filled at the current time due to a lack of available supply. The item may not be held in the company's available inventory but could still be in production, or the company may need to still manufacture more of the product. The backorder is an indication that demand for a company's product outweighs its supply. They may also be known as the company's backlog. Machine learning algorithms can help to predict whether the item will be backordered or not. This study demonstrates how different classification algorithms can predict the outcome. Different classification algorithms such as Logistic regression, Random Forest, Decision Tree, Xgboost, Support Vector Machine have been tested and compared to predict the better outcome of the model.

# 1. Introduction

## 1.1 Why this High-Level Design Document ?

The purpose of this High-Level Design (HLD) Document is to add the important details about this project. Through this HLD Document, I'm going to describe every small and big things about this project.

# 2. General Description

## 2.1 Product Perspective

The Back\_Order Prediction predict the wether backorder is possible or not using classification based Machine Learning algorithms.

## 2.2 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).

## 2.3 Proposed Solution

The proposed solution for this project is Machine learning algorithms can be implemented to predict backorder. Considering various features like inventory quantity, previous performance, minimum\_balance, forecast\_sales , actual\_sales etc as inputs from the web app, the

## High Level Design (HLD)

implemented classification model will predict the output Here, we have used Random Forest Classifier to predict . However, drawing a baseline model is important since it tells us how well other models have performed compared to base model. Here, the base model for Bacorder Prediction dataset is SVM,Decision\_Tree ,Logistic Regression.

## 2.4 Technical Requirements

In this Project the requirements to predict and through various platform. For that, in this project we are going to use different technologies. Here is some requirements for this project.

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

## 2.5 Data Requirements

Data Requirement completely depend on our problem.For training and testing the model, we are using Back\_Orderprediction dataset from kaggle.

National_inv	Present Inventory level
Lead_Time	Transit time of Product

## High Level Design (HLD)

In_transist_Qty	Amt of Product in Transit
Forecast_9_month	Forecast of sales
Slaes_9_month	Actual 9 month sale
Deck_Risk	True, False
Performance_9_month	Performance over 9 Months
Went_on_backordred	True, False

## 2.6 Tools Used



PyCharm is used as IDE.

For visualization of the plots, Matplotlib, Seaborn are used.

Heroku is used for deployment of the model.

MongoDB is used to retrieve, insert, delete, and update the database.

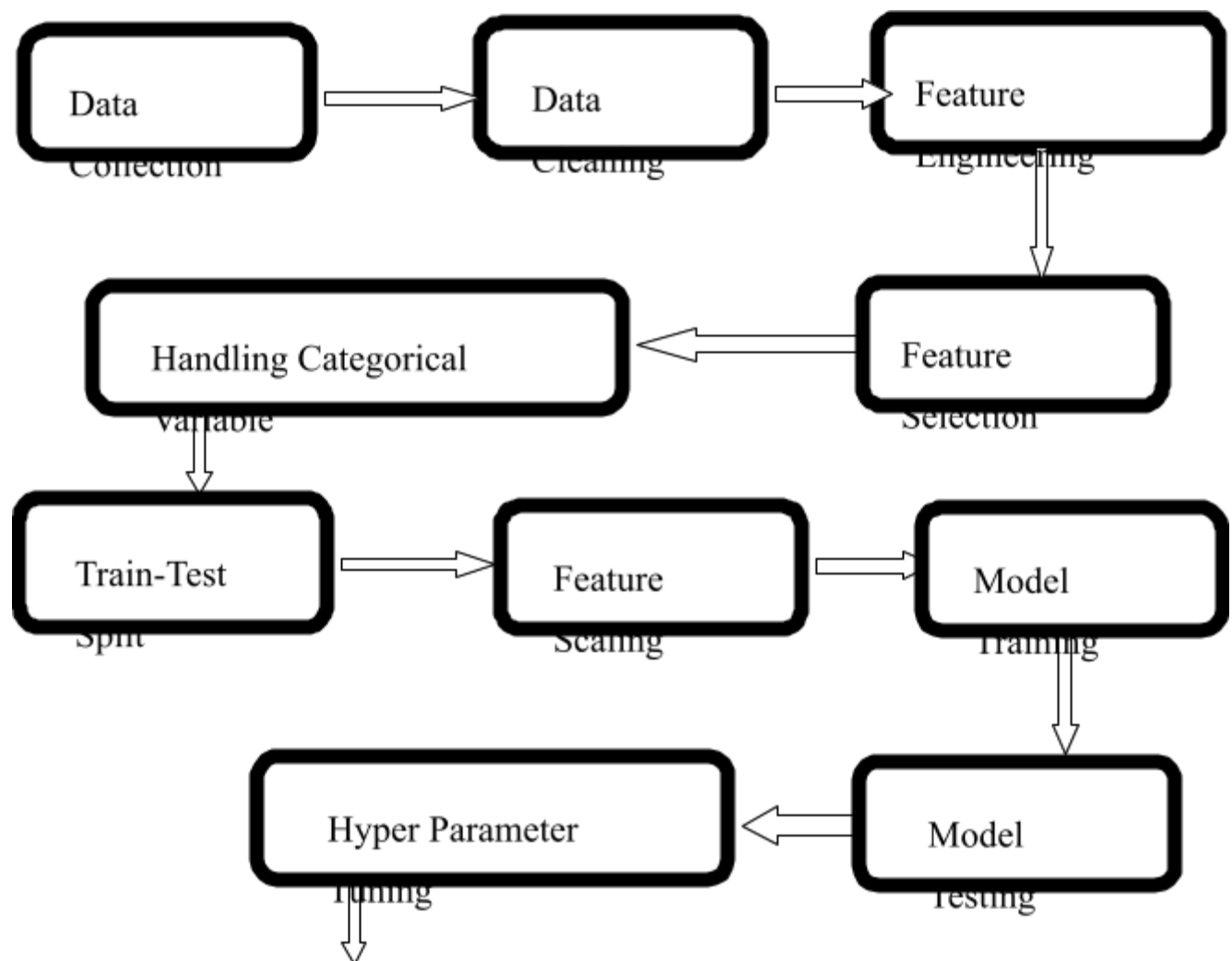
Front end development is done using HTML/CSS, Flask is used for

## High Level Design (HLD)

backend development and for API development.  
GitHub is used as version control system.

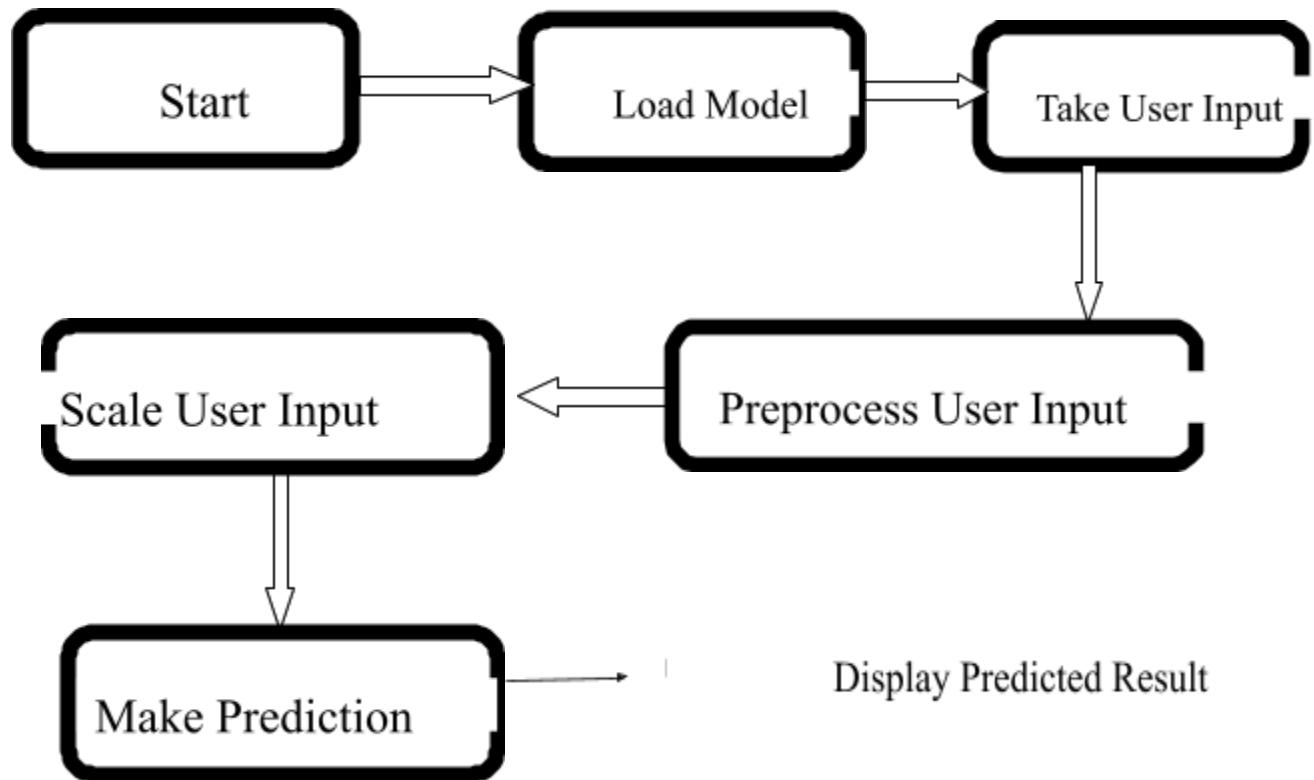
## 3 Design Details

### 3.1 Process Flow





## 3.2 Deployment Process



## 3.3 Event Log

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

Step-By-Step Description:

In this Project we defined logging for every function, 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

## High Level Design (HLD)

hang even after so many logging's, so that we can easily debug issues which may arises during process flow.

### **3.3 Error Handling**

We have designed this project in such a way that, at any step if error occur then our application should not terminate rather it should catch that error and display that error with proper explanation as to what went wrong during process flow.

## 4. Performance

Solution of Back Order prediction is used to predict whether it's out of stock or not in advance, so it should be as accurate as possible so that it should give as much as possible accurate prediction.

That's why before building this model we followed complete process of Machine Learning . Here are summary of complete process:

1. First we cleaned our dataset properly by removing all null value and duplicate value present in dataset.
2. Performed EDA on whole data set. Did type conversion of some columns .Tried to find the pattern
3. Then I handled categorical variable by performing One-Hot encoding.
4. With the help of Correlation plot we found that Sales(3,6,9 month), Forecast(3,6,9) and Performance(6,9) columns were heavily correlated.Hence dropped some columns from those
5. Since data was highly imbalance, performed sampling on data
6. Then I split the hole data set train-test split. After that I performed scaling on X\_train and X\_test
7. After performing above step I was ready for model training. In this step, I trained my dataset on different algorithm(Logistic, Random-Forest, SVM, DecisionTreeRegression,Random-Forest and XGB). After training the dataset on different algorithms I got highest accuracy of 97% on RandomForestRegression.
8. After that I applied hyper-parameter tuning on all model which I have described above. Here also I got highest accuracy of 98% on test dataset by same RandomForestRegression.
9. After that I saved my model in pickle file format for model deployment.
10. After that my model was ready to deploy. I deployed this model on various cloud storage( heroku)

## 4.1 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.

## 4.2 Application Compatibility

The different module of this project is using Python as an interface between them. Each modules have it's own job to perform and it is the job of the Python to ensure the proper transfer of information.

## 4.3 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 multi-threading.

## 4.4 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 multi-threading.

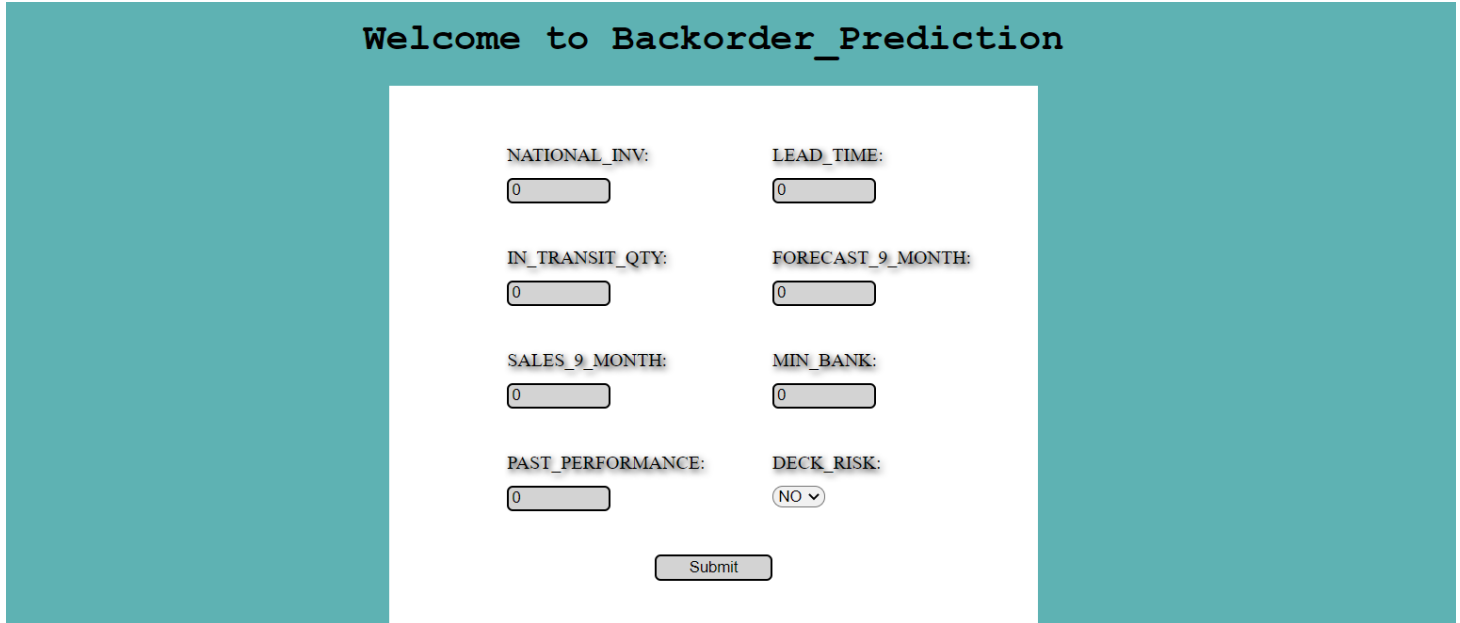
## 4.5 Deployment

we have deployed this on cloud Heroku



## 4.6 User Interface

We have created an UI for user by using HTML and CSS.



The screenshot displays a web application titled "Welcome to Backorder\_Prediction". The interface is set against a teal background. A central white rectangular area contains the input fields. There are eight input fields arranged in two columns. The left column contains: "NATIONAL\_INV:" with a text box containing "0", "IN\_TRANSIT\_QTY:" with a text box containing "0", "SALES\_9\_MONTH:" with a text box containing "0", and "PAST\_PERFORMANCE:" with a text box containing "0". The right column contains: "LEAD\_TIME:" with a text box containing "0", "FORECAST\_9\_MONTH:" with a text box containing "0", "MIN\_BANK:" with a text box containing "0", and "DECK\_RISK:" with a dropdown menu showing "NO". Below these fields is a "Submit" button.

Welcome to Backorder\_Prediction

NATIONAL_INV:	LEAD_TIME:
<input type="text" value="0"/>	<input type="text" value="0"/>
IN_TRANSIT_QTY:	FORECAST_9_MONTH:
<input type="text" value="0"/>	<input type="text" value="0"/>
SALES_9_MONTH:	MIN_BANK:
<input type="text" value="0"/>	<input type="text" value="0"/>
PAST_PERFORMANCE:	DECK_RISK:
<input type="text" value="0"/>	<input type="text" value="NO"/>

Submit

## 5. Conclusion

- The BackOrder prediction model will predict the backorder in prior so that retailers can replenish the stock as soon as possible