High Level Design (HLD)

BackOrder Prediction

x

Document Version Control

Date Issued	Version	Description	Author
24/06/2022	1	Initial HLD-V1.0	Umang Tank
3/07/2022	1.1	HLD V1.0	Umang Tank

Contents

Abstract

In today's world, data is power. With News companies having terabytes of data stored in servers, everyone is in the quest to discover insights that add value to the organization. With various examples to quote in which analytics is being used to drive actions, one that stands out is news article classification.

1 Introduction

1.1 Why this High-Level Design Document?

The purpose of this High-Level Design (HLD) Document is to add necessary details to the current project description to represent a suitable model for coding. This model 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.

The HLD will:

- Present all of the design aspects and define them in details.
- Describe the user interface being implemented - Describe the hardware and software interfaces
- Describe the performance and 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 structure of the system, such as the database architecture, application architecture, application flow (Navigations), and technology architecture. The HLD uses non-technical to mildly-technical term which should be understandable to the administrator of the system.

2 General Description

2.1 Product Perspective

The Backorder prediction problem 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 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 solution here is a Classification based Machine Learning model. It can be implemented by different classification algorithms (like Logistic Regression, Random forest, Decision tree , XGBoost and so on.Here First we are performing Data preprocessing step, in which Data Profiling ,feature engineering, feature selection, feature scaling ,PCA steps are performed and then we are going to build model.

2.4 Technical Requirements

In this Project the requirements to check to predict the backorder sales for a particular product according to the provided dataset. For that, in this project we are going to use different technologies. Here is some requirements for this project.

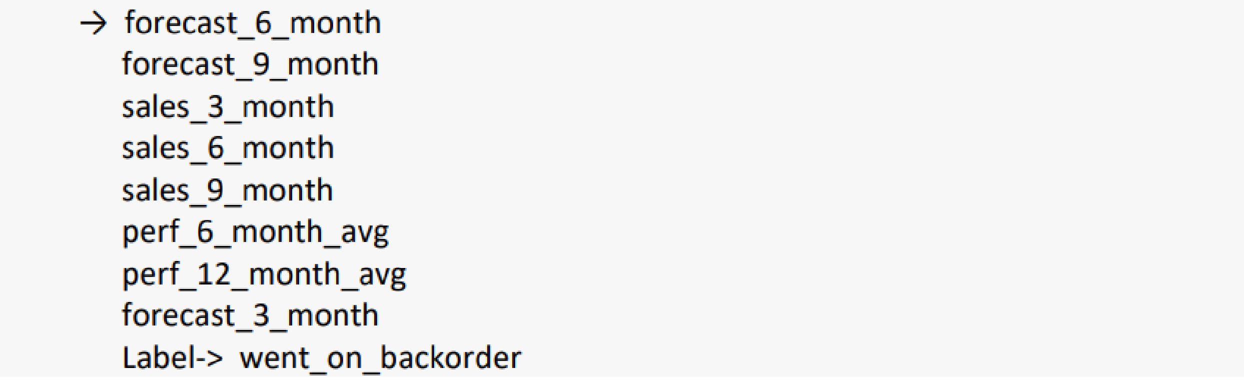
• Model should be exposed through API or User Interface, so that anyone can test model.

• Model should be deployed on cloud (Azure, AWS, GCP).

• mongoDB database should be integrated in this project for any kind of user input.

2.5 Data Requirement

- Data Requirement completely depend on our problem. • For training and testing the model, we are using Backorder prediction dataset that is provided by Ineuron Company.
- From user we are taking following input:)



→ forecast_6_month
forecast_9_month
sales_3_month
sales_6_month
sales_9_month
perf_6_month_avg
perf_12_month_avg
forecast_3_month
Label-> went_on_backorder

2.6 Tools used

- VS-Code 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, Bootstrap.
- Flask is used for backend development and for API development.
- GitHub is used as version control system.

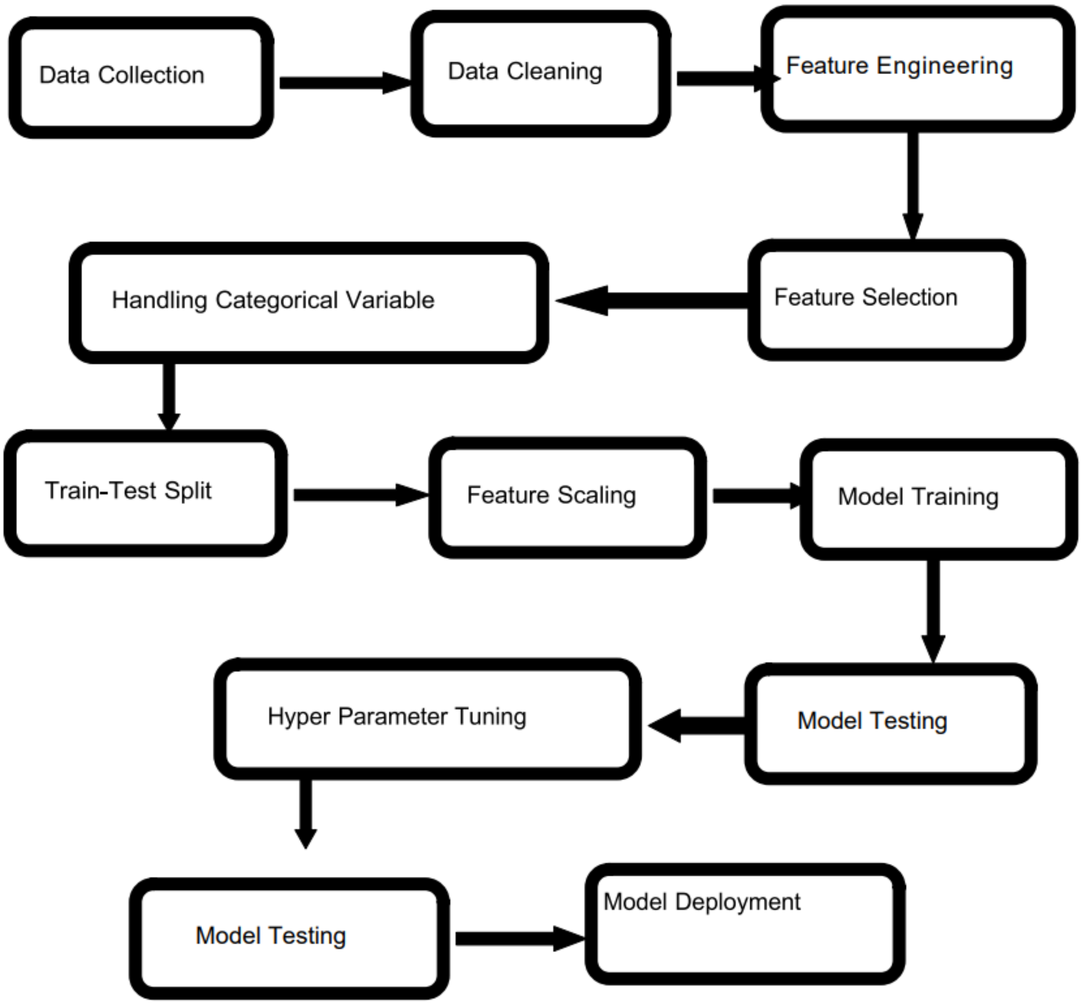
2.7 Constraints

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

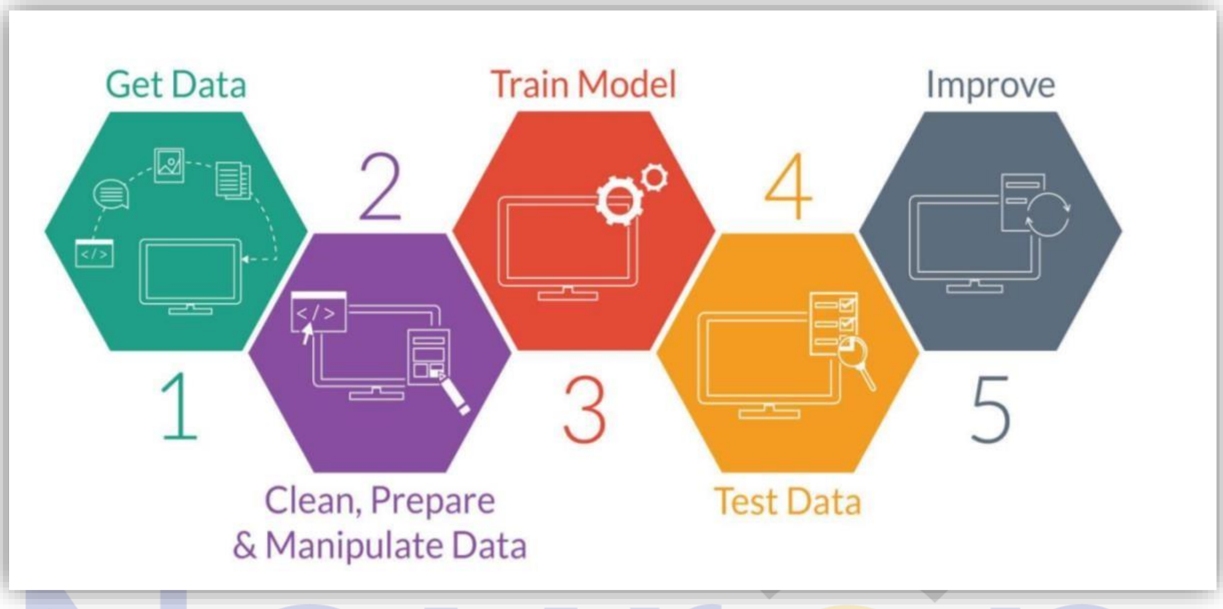
3. Design Details

3.1 Process Flow

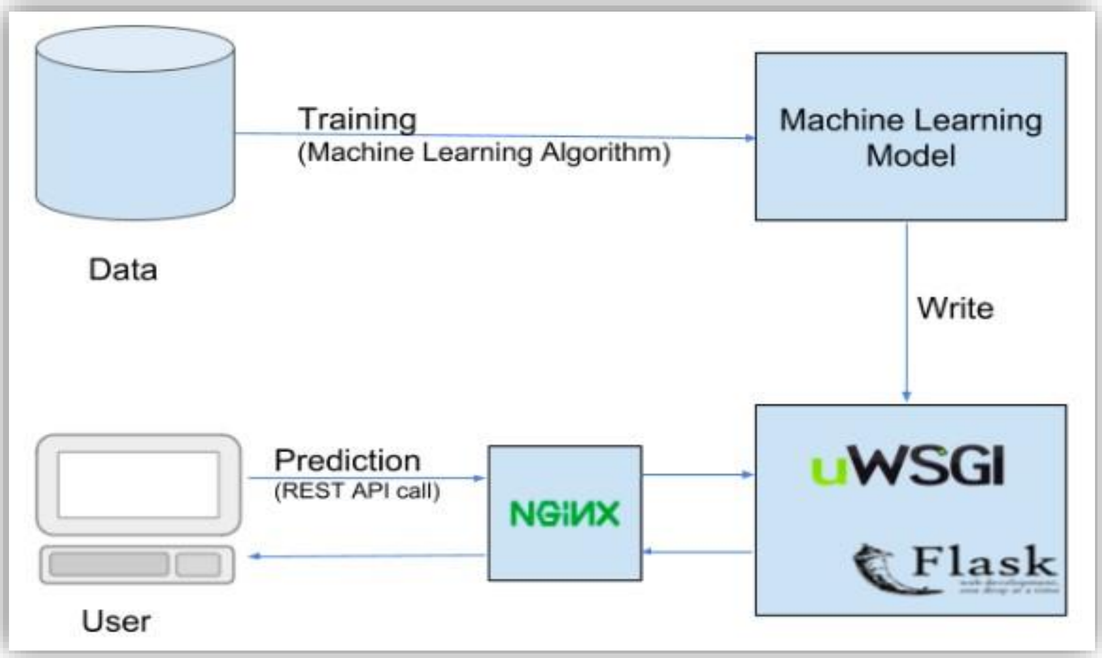
.


Data Collection
Data Cleaning
Feature Engineering
Feature Selection
Handling Categorical Variable
Train-Test Split
Feature Scaling
Model Training
Model Testing
Hyper Parameter Tuning
Model Testing
Model Deployment

3.1.1 Model Training and Evaluation



3.1.2 Deployment Process



3.2 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 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.

3.4 Reusability

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.

3.5 Application Compatibility

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

3.6 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..

4 Conclusion

Background In this project, five Classification models are evaluated for individual News Article data. It has been found thatMultinomialNB model which is built is the best performing model.