DATA SCIENCE: MACHINE LEARNING PROJECT

Data Science Open Internship, INeuron.AI

High Level Design Report

On

BACKORDER PREDICTION

Made by**: Varun Sagar Theegala**

Main Technology**: Machine Learning**

Domain**: E-commerce**

Project Start Date**: 14th August 2021**

ABSTRACT

Backorders have increasingly become a very important area of concern & significance for E-commerce setups. Backorders refers to the order containing a product that is not in stock when the customer searches for it on an online platform. However, a customer is allowed to place an order for the product and wait for it to become available, after which it would be delivered

However, Backorders means a long waiting time for customers to receive their products. Also, this means a lot of additional tasks and burden for the company as it will need to get in contact with their suppliers, plan for the product, plan the storage of the product, costs and the logistics involved from ordering the product to its delivery. Sometimes, a long duration to fulfil backorders can make customers frustrated and turn them away to other alternate sellers of the product. This results in loss of potential sales and customers for a company.

Therefore, having a model in place to identify potential items which could become backorders would be useful for e-retailers to make necessary arrangements for the product before time to meet the demands from the customers and avoid losing sales opportunities, additional costs, debts and most importantly, customers.

TABLE OF CONTENTS

1. INTRODUCTION

1.1 What is a High-Level Design Document?

The purpose of a High-Level design (LLD) document is to provide additional details to the current project description given in the low-level document, architecture, and other reports.

This report contains information not only related to building the project and code, but also the various evaluations to assess the code written, how different tools and libraries interact other useful pieces of information to understand the engineering of the underlying programs and interfaces built for the project.

The HLD will help in the following ways –

- Describe the workflow adopted to build the project.
- Give a detailed description of the user interface built.
- List the required software i.e technologies, libraries in Python, etc.
- List and help understand performance evaluation measures for the project such as listed below:
    - Security
    - Reliability
    - Maintainability
    - Reusability
    - Application compatibility
    - Resource utilization
    - Serviceability

1.2 Scope of a High-Level Design Document

A Low-Level Design document will provide the details for the design of the project and the workflow adopted for various tasks such as model building, user interface development, deployment and use of technology.

For easy interpretation of the report by viewers, administrators and viewers from a non-technical background or different domain, mildly technical terms have been used at most in the report.

1.3 Definition

Term Used	Description
IDE	Integrated Development Environment
AWS	Amazon Web Services
UI	User Interface

2. General Description

2.1 Product Perspective

The Backorder Prediction web application is a machine learning model at the core which could help us identify in-demand or popular products that could become backorders.

2.2 Problem Statement

To create a Machine Learning predictive model which could determine whether a given product, based on its sales figures, inventory & other important information, could become a backorder or not.

2.3 Proposed Solution

The proposed solution for the problem statement is a Backorder Prediction model, which can be used to take specific decision with respect to inventory management, such as quantity to order, storage, inventory acquisition techniques, etc based on whether a product is identified by the model as a potential backorder or not. The model will be trained using a huge no. of product details from an e-commerce company. This model will be made accessible to user through its deployment on the internet and on cloud platforms in the form of a web application.

2.4 Further Improvements

The dataset used in the project has no information related to the seasonal sales, product type or suppliers. Adding such information to the dataset in the future could also lead to an opportunity to create multiple backorder prediction models for each product type or season for more accurate predictions due to learning specific patterns within the given cluster of data.

Also, adding such information could facilitate an in-depth analysis of the relationship between backordered product and other features currently present in the dataset based specific seasons or/and product types. Also, we could understand the no. of suppliers needed to supply required quantities of the product to meet demands.

2.5 Technical Requirements

To create the Backorder Prediction web application, we are going to make use of various technologies. Some of the major requirements for the completion of the project are –

- The project needs to be deployed on cloud platforms as well as locally on the internet.
- The user should be able to use the model by the medium of a UI.

2.6 Data Requirements

To use the model to get the predictions for their product, the user needs to fill the input fields in the web application with the following information. These are the same features using which the final model has been trained.

S.no	Input	Units	Format	Note
1	Total Sales for past 6 months	Amount	Float / Decimal or Integer	Required
2	Forecasted Sales for next 6 months	Amount	Float / Decimal or Integer	Required
3	Average Product Performance in past 6 months	Percentage	Float / Decimal or Integer	Required
4	Present Inventory Level	Units Sold	Integer	Required
5	Minimum Stock recommended	Units Sold	Integer	Required
6	Level of Stock in transit	Units Sold	Integer	Required
7	Present Backorder quantity for product	Units Sold	Integer	Required
8	Lead time for the product	Days	Integer	Required
9	Any Deck risk associated	Binary i.e Yes or No	- (Options provided)	Default option selected - Yes
10	Any Part Production Approval risk associated	Binary i.e Yes or No	- (Options provided)	Default option selected - Yes

2.7 Tools Used

The Backorder Prediction web application is a project built using several technologies and programming libraries. The various tools used in the project are displayed below

- Programming Language – **Python**
- IDE – **Amazon Sagemaker**
- Text Editor – **Atom**
- Python DataFrame manipulation library(s) – **Numpy & Pandas**
- Python Data Visualization library(s) - **Matplotlib & Seaborn**
- Python Model Building library(s) – **Scikit Learn**
- UI Development – **HTML, CSS & JavaScript**
- UI & Model Integration – **Flask**
- Internet Deployment – **GitHub & Heroku**
- Cloud Deployment – **AWS Ec2**

2.8 Constraints

For the successful use of the solution the following constraints were or need to be met:

- The Backorder Prediction model requires users to field inputs for specific set of questions to generate predictions. The questions should be understandable and easy to collect information for.
- The entire solution must be simple, informative & user friendly so that the user can navigate through the application without guidance and get required output.
- To access the solution, the users need to have access to the internet or cloud.

2.9 Assumptions

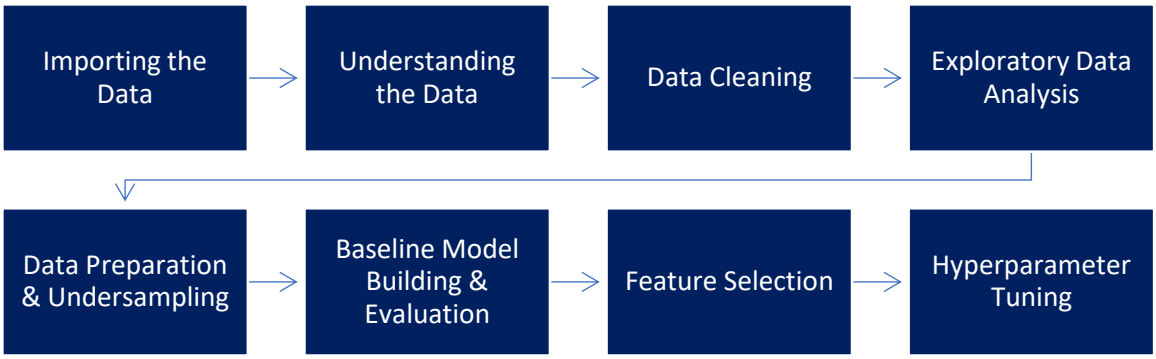
There are a set of assumptions being made during the deployment and use of the application:

- To use the machine learning model, the user to need provide necessary details related to a product. It is assumed that the user has access to the required information, which help the model make a meaningful prediction.
- At the start of the application, all elements used to design the UI function properly.
- All elements of the project should work in the manner as described in the reports i.e The applications URLs should be valid and start the application, HTML, CSS, and JavaScript should render the UI, user should be able to fill the input fields & the model created using Python programming and libraries should be running, processing the input supplied by user and generate results.

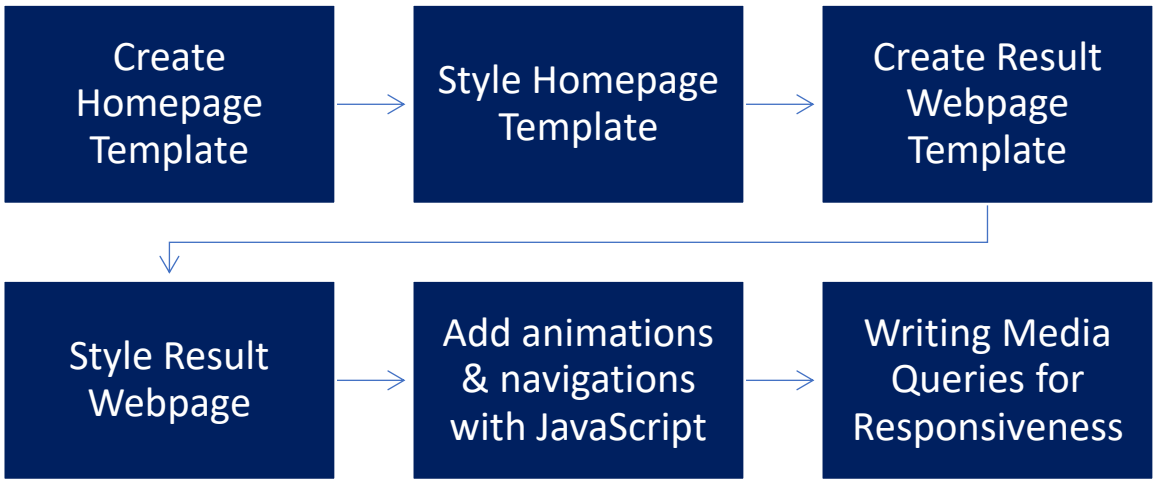
3. DESIGN DETAILS

3.1 Process Flow

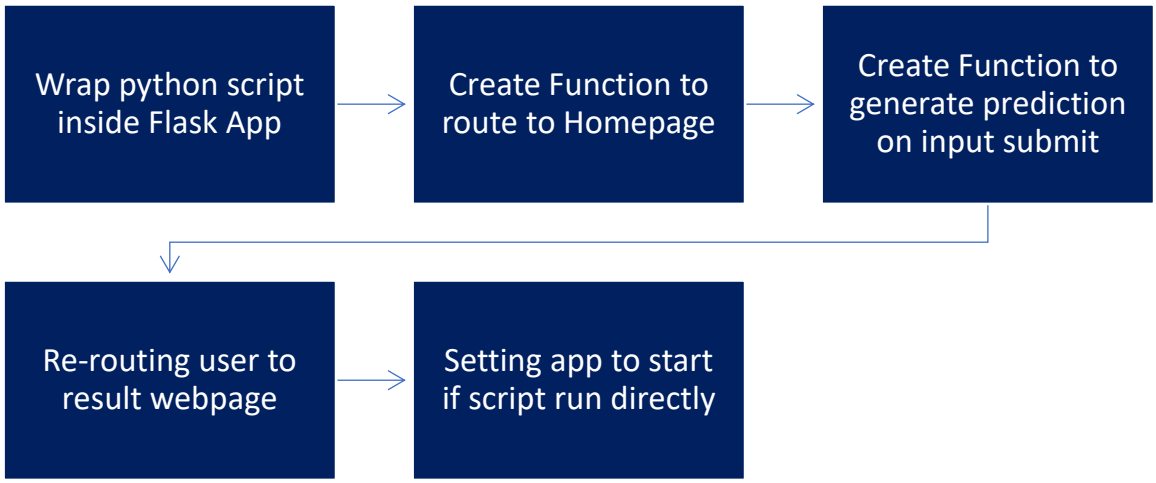
Model Building Flow



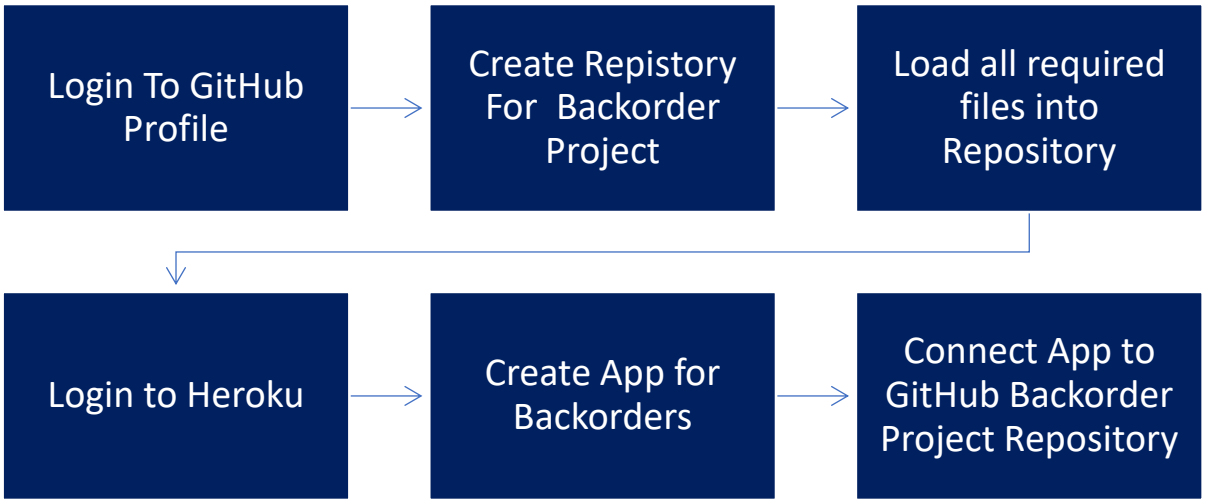
UI Development Flow



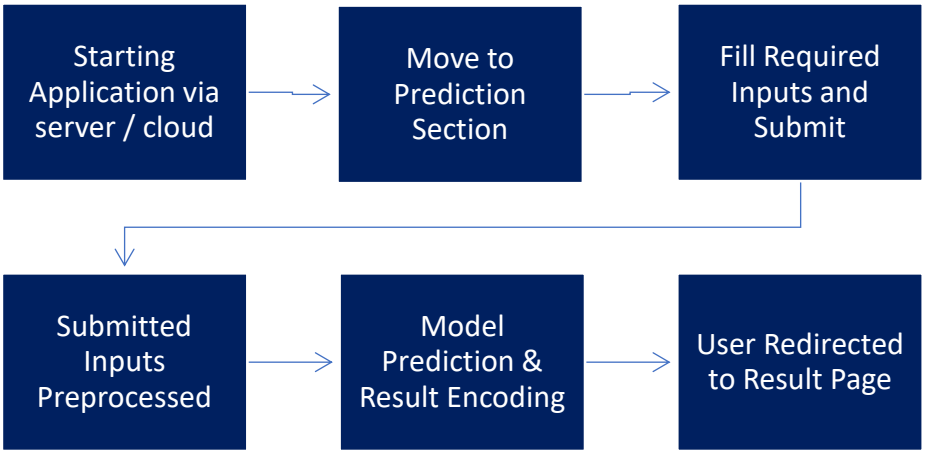
App Development Flow



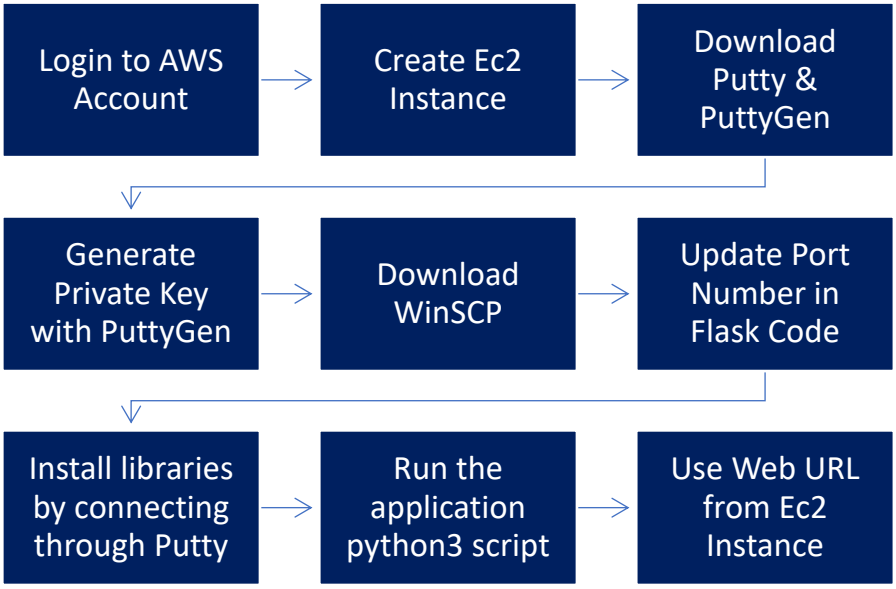
Deployment Flow (GitHub & Heroku)



Deployment Flow (Cloud using AWS)



Prediction Flow



3.2 Event Log

A logging mechanism has been put in place so that all important steps during the model building process and all predictions made can be tracked.

- Important steps performed during phases such as Exploratory Data Analysis, Data Preparation, Model Building, Feature Selection & Hyperparameter Tuning were tracked.
- Dates and time at which step was successfully completed has been recorded for any potential error debugging purposes.
- When model successfully makes the prediction, the completion & predicted value is logged for record purposes.
- The log entries will be transferred to a text file.

3.3 Error Handling

The Logging mechanism implemented and other pieces of code for building the UI have been written in a way to smoothly encounter and make users aware of errors to debug.

- Description of Error occurred will be recorded and displayed to user for easy debugging.
- Field input fields have been prepared in a way to accept only specific type of input. Any other input format followed by user will not be accepted and displayed inside input box.

4. PERFORMANCE

4.1 Reusability

Several measures have been made to ensure the entire program can be rebuilt and all code written, and components used can be re-used for building the same application.

- The Backorder Prediction web application has a requirements text file containing the list of all interdependencies i.e libraries and respective versions, needed for model building and flask application python code.
- Detailed architecture, low-level and high-level design reports have been prepared with details on all steps required to follow to build the application.
-

4.2 Latency

The machine learning model created has been built in a way to reduce the amount of time taken to generate a prediction for the given input from the user. On the whole, the model takes a total of 0.01158 seconds to generate a prediction, which is less than a second.

4.3 Application Compatibility

The several tools used in this project have been connected with each other, using Python as the interface, to build the end-to-end ML Backorder prediction web application. The coding in Python has been done to ensure all required information, if needed, are transferred carefully and accurately, from one component (ex – Input fields on HTML Page) to another (Prediction model).

4.4 Resource Utilization

When different tasks are being performed, the server is likely to use required or all of the processing power available to complete that task.

4.5 Deployment

The Backorder Prediction web application has been deployed on cloud using Amazon Web Services Ec2 and on the internet using Heroku App so that users have several methods to access the solution.

5. CONCLUSION

The Backorder Prediction model will identify whether a product, based on its sales, forecasts, levels of inventory and associated risks, will become a backorder or not. This prediction will help E-commerce entities in recalculating their forecasted sales, improve inventory management operations like ordering, storage, etc, mitigate risk of bad debts against backordered products cancellation and most importantly, meet the demand of customers, thereby never losing out on potential sales opportunities.