DATA SCIENCE: MACHINE LEARNING PROJECT

Data Science Open Internship, INeuron.AI

Architecture 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 in order 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 Purpose of Low-Level Design Report

The purpose of preparing a low-level design report for the project is to impart necessary information to concerned and prospective stakeholders and management personnel, in respect to the design adopted to build the project, the interface created to interact with the model and the system & input requirements that need to be satisfied in order for the model to perform as needed.

Also, the document will act as a source of guidance to potential programmers who wish to replicate the project for their personal or commercial use as they could follow the steps mentioned to rebuild the machine learning model and user interface from scratch to the final state.

1.2 Scope:

The scope of the project has several applications.

- An e-commerce platform can use the model to identify the in-demand products they should order and keep ready to fulfil the needs of the customer in the shortest amount of time.
- Using the model will facilitate reduction of unnecessary costs, backtracking with suppliers or strain on transportations that would need to be undertaken to complete such orders.
- Identifying potential backorder products in time will prevent the firm for facing a large no. of order cancellations, which would mitigate the risk of debts that could occur in case huge sums are borrowed and invested in hope of generating huge sales from potential backorder products.
- Inventory management operations could improve as management can precisely plan the purchase of the products & required quantities to be ordered from various suppliers within a given period and their storage in the warehouses in the most effective manner.
- Most importantly, identification of backorder product could allow to keep optimum level of stock ready to meet the demand for the product & avoid losing out on sales opportunities, which could otherwise fall in the hands of rivals.

This product will be deployed in the form of a web application to make it an easily accessible service for users, who would only require a stable internet connection and key information related to their concerned products to get the required output.

1.3 Constraints

There are a few constraints associated with the project which potential users need to be cautious of, mainly –

- The users need an internet connection to access the application for usage.
- The users require to have specific details regarding the product such as its past sales, forecasted sales figures, lead time, backorder quantity for the product due, current inventory level of the product, specific risks like part production approval, etc. Gathering such pieces of information could be time consuming.
- Information collected must be in the correct format i.e numbers, string, etc for the model to accept and process.

1.4 Risks:

The only risk involved in the entire process for the user is the interpretation of one of the inputs required by the model i.e average performance of the product over last 6 months. Despite best efforts to find accurate meaning of the feature, its true meaning couldn’t be found. It was assumed that the feature is related to % of forecasted quantity and hence its unit is taken as % Hence the value entered by the user for this input could impact the output predicted by the model.

1.5 Out of Scope:

The information of the product type, location to be delivered, quantity ordered by the customers or no. of suppliers for the product is not present in the dataset and hence outside the scope of the model.

2. TECHNICAL SPECIFICATIONS

2.1 Dataset

The dataset used for building the model originally contained close to 16,87,860 records and 23 variables in total, of which 22 were potential predictors and 1 i.e went on backorder, was the target variable to predict.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	sku	national_inv	lead_time	in_transit_qty	forecast_3_month	forecast_6_month	forecast_9_month	sales_1_month	sales_3_month	sales_6_month	sales_9_month	min_bank	potential_issue	pieces_past_due	perf_6_month_avg	perf_12_month_avg	local_bo_qty	deck_risk	oe_constraint	ppap_risk	stop_auto_buy	rev_stop	went_on_backorder
2	1026827	0		0	0	0	0	0	0	0	0	0	No	0	-99	-99	0	No	No	No	Yes	No	No
3	1043384	2	9	0	0	0	0	0	0	0	0	0	No	0	0.99	0.99	0	No	No	No	Yes	No	No
4	1043696	2		0	0	0	0	0	0	0	0	0	No	0	-99	-99	0	Yes	No	No	Yes	No	No
5	1043852	7	8	0	0	0	0	0	0	0	0	1	No	0	0.1	0.13	0	No	No	No	Yes	No	No
6	1044048	8		0	0	0	0	0	0	0	4	2	No	0	-99	-99	0	Yes	No	No	Yes	No	No
7	1044198	13	8	0	0	0	0	0	0	0	0	0	No	0	0.82	0.87	0	No	No	No	Yes	No	No
8	1044643	1095		0	0	0	0	0	0	0	0	4	No	0	-99	-99	0	Yes	No	No	Yes	No	No
9	1045098	6	2	0	0	0	0	0	0	0	0	0	No	0	0	0	0	Yes	No	Yes	Yes	No	No
10	1045815	140		0	15	114	152	0	0	0	0	0	No	0	-99	-99	0	No	No	No	Yes	No	No
11	1045867	4	8	0	0	0	0	0	0	0	0	0	No	0	0.82	0.87	0	No	No	No	Yes	No	No
12	1045918	0	2	0	0	0	0	0	0	0	0	0	No	0	0.91	0.82	0	No	No	No	Yes	No	No
13	1047146	20		0	0	0	0	0	0	0	0	0	No	0	-99	-99	0	Yes	No	No	Yes	No	No
14	1047199	18		0	0	0	0	0	0	0	0	0	No	0	-99	-99	0	Yes	No	No	Yes	No	No
15	1047661	29		0	0	0	0	0	0	0	0	0	No	0	-99	-99	0	Yes	No	No	No	No	No
16	1049160	10		0	0	0	0	0	0	0	0	0	No	0	-99	-99	0	Yes	No	No	Yes	No	No
17	1049468	11	8	0	0	0	0	0	0	0	0	0	No	0	0.82	0.78	0	No	No	No	Yes	No	No
18	1050390	12	2	0	0	0	0	0	0	0	0	0	No	0	1	0.98	0	No	No	No	Yes	No	No
19	1050440	169	2	0	0	0	0	0	0	0	0	0	No	0	1	1	0	No	No	No	Yes	No	No
20	1117808	4		0	0	0	0	0	0	0	0	0	No	0	-99	-99	0	No	No	No	Yes	No	No
21	1050856	147	8	0	0	0	0	0	0	0	0	0	No	0	-99	1	0	No	No	Yes	Yes	No	No
22	1118063	8		0	0	0	0	0	0	0	0	1	No	0	-99	-99	0	Yes	No	No	Yes	No	No
23	1127154	18		0	0	0	0	0	0	0	0	0	No	0	-99	-99	0	Yes	No	No	Yes	No	No
24	1127410	9		0	0	0	0	0	0	0	0	1	No	0	-99	-99	0	Yes	No	No	Yes	No	No
25	1128393	33		0	0	0	0	0	0	0	0	1	No	0	-99	-99	0	Yes	No	Yes	Yes	No	No
26	1051516	82	8	0	0	0	0	0	0	0	0	0	No	0	-99	1	0	No	No	No	Yes	No	No
27	1128591	42		0	0	0	0	0	0	0	0	1	No	0	-99	-99	0	Yes	No	No	Yes	No	No

However, during the training phase, due to heavy imbalance in representation of target classes in the dataset, the majority class was under-sampled and eventually the final model created was trained with a total of 56,465 records.

2.2 User Interface

The user interface i.e the web application, designed to enable interaction of the user with the model has been built using HTML, CSS, JavaScript & Flask

The user interface contains the following –

- Landing Page: This is the first page that is visible to the user on starting the application. Consists of navigation pane and buttons to allow users to scroll to desired sections.
- Backorders Description Section: This is the 1st section on the web page, which provides important information to users about backorders and its significance/ risks posed to e-commerce platforms.
- Prediction Section: This is the 2nd section, with a form containing several fields that need to be filled by the users according to the questions asked and specifications i.e units and format. On successfully filling all information, the user can hit the submit button which will enable to form to process the information provided and pass it to the model.
- Results Page: This is a 2nd and final page of the application where the result generated by the model is displayed to the users.

How the following technologies were used to build the interface are given below:

HTML – The front-end interface for the application i.e webpages, have been designed using Hyper Text Markup Language. HTML forms have been used to create a form that consists of inputs and options for the user to pass information about the product, using which the model would produce the required output

CSS – The webpages for the application have been designed using Cascading Style Sheets. Media Queries have been used to make the application and its interface look presentable & feel usable across several devices like Mobile phones, Tablets and Laptops

JavaScript – In order to facilitate scrolling across the web application to different sections and provide a navigation pane for the users, JavaScript has been used.

Flask – In order to connect the front-end user interface with the predictive model, the Flask framework has been utilized. The model has been wrapped inside a flask application, which was connected to the front-end interface using flask.

2.3 Input Schema

The model working at the backend, which will make the prediction, requires a total of 10 inputs from the user. The features and the format of the input required is given as follows:

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

2.4 Logging

Through the model building phase and generating predictions for user inputs after deployment, logging was utilized to capture & track important information or steps being performed.

Such information was logged and stored directly into a text file along with the timestamp and specific description, which would allow the viewers to understand steps, identify any specific errors and locate the error with respect to time of occurrence.

2.5 Predictions

- The application contains a prediction section, which include a form where several fields that need to be filled by the users, are displayed to the user.
- The users are assisted in respect to unit associated with each input required i.e units sold, sales amount, days, etc
- With proper formats set for each input using HTML, any wrong formatted input entered by user will be blocked by the input fields
- 8 out of the 10 fields need to be mandatorily filled by the user to submit the form.

- 2 fields, i.e binary questions, are set to “Yes” by default.
- On filling all required fields and submitting, the input is processed by the model and output by the model is produced in 0 or 1, with 0 representing non backorders and 1 being backorders.

2.6 Deployment

- The model has been deployed on the web server through Heroku. This is accessible to everyone by using the URL for the web application.
- The model has been hosted on the cloud using AWS Ec2 instance. To create a virtual machine to push the application to, Ubuntu was selected due to its low cost and since our model requirements were not many and mainly basic. To use the machine, authentication using public and private keys was performed with the help of PuTTy, PuTTyGen and WinSCP. This cloud-based application is accessible to cloud users using the application’s public cloud address.

3. TECHNOLOGY STACK

Task	Tool/ Platform/ Service Used
Machine Learning Model Building & Other Backend Programming	Python
Connecting Model to User Interface	Flask
User Interface Creation	HTML, CSS & JavaScript
Deployment	Heroku & AWS Ec2

4. PREDICTION MODEL SOLUTION

- To predict, for given set of information related to a product, whether a product could become a backorder for the e-commerce platform or not, a Machine Learning model is used. Since there are 2 outcomes, we are concerned with i.e if a product is a backorder or not, the Supervised Classification Machine Learning algorithms are used.
-
- There were several Classification models that were tested such as Logistic Regression, Nearest Neighbours Classifier, Support Vector Classifiers, Decision Tree and Ensemble techniques such as Bagging (Random Forest) & Boosting (Adaboost, XGB & Gradient Boosting).
-
- From the several baseline models built, Random Forest was chosen and taken ahead for feature selection & Hyperparameter tuning.
-
- The final model built was a baseline Random Forest model with 10 features.

5. MODEL TRAINING & WORKFLOW

Importing the Dataset	• Downloading the dataset from source
Converting Data into DataFrame	• Importing the dataset into Python in the form of Dataframe
Data Understanding	• Checking dataset size, shape, type of variables, no. of unique values & target class proportions.
Data Cleaning	• Checking for lead spaces in columns names, category names & dropping empty records.
Exploratory Data Analysis	• Performing Univariate, Bivariate & Multivariate Analysis, Statistical Analysis & Treating Multicollinearity
Data Preparation	• Splitting data into train & test samples, treating missing values, outliers, normality & category encoding.
Undersampling Dataset / Reducing Imbalance	• Combining both samples, undersampling majority class and re-splitting into train and test sample
Model Building	• Building Baseline models, evaluating models, cross validating models for stability & calculating latency.
Feature Selection & Hyperparamter Tuning	• Performing Backward & Recursive Feature Elimination, Feature Importances & Grid Search for best parameters.
User Interface Development & Integration of Front & Backend	• Building User Interface with HTML, CSS & JavaScript and connecting to model using Flask.
Deployment	• Deployed on local server using Heroku and pushed to Cloud platforms using AWS Ec2 instance.

6. USER I/O WORKFLOW

Starting Application	• User can start application in local server or cloud and scroll down to prediction section
Filling Input	• Filling required input fields in correct formats and units and submitting.
Input Pre-processing	• Numerical data transformed using Power Transformer & Categorical data encoded.
Model Prediction	• Input fed to model and output generated in 0 or 1.
Transforming Result	• Output from model encoded as "Not a Backorder" if 0 else "A Backorder" if 1.
Result Displayed	• User re-directed to results webpage where final transformed result from model is displayed.

7. KEY PERFORMANCE INDICATORS

During the model building and tuning phase, a set of 10 features were identified which were driving the predictions made.

These features are the key performance indicators based on which a product's status as backorder or not is predicted –

- Sales of the Product in the past 6 months
- Forecasted Sales of the Product in the following 6 months.
- Product Performance against forecasted units to sell in last 6 months on an average.
- Total level of Inventory for Product present with the company.
- Total level of Inventory in transit / transportation.
- Minimum level of Inventory recommended to hold.
- Total quantity of product that is currently in backorder status.
- The lead time i.e days between placing order for product by company to supplier & receiving the product from suppliers.
- Any risk associated with respect to the part production approval for the product / ppap risk flag.
- Any risk associated with respect to storage of product / deck risk flag

8. TEST CASES

Test Case Description	Pre-Requisite	Expected Result
Access to Application URL	1. Application URL should be defined on project platform like GitHub or in LinkedIn posts related to the project.	Application URL should be accessible to the user.
Application Loading	1. Application URLs for the cloud and server should be available. 2. The Application should have been successfully deployed using Heroku. 3. The application should have been successfully hosted in cloud using Ec2.	On using the URLs, the web application should start.
Interface Performance	1. The links to the folders for CSS, templates and JavaScript should be present in the Python file. 2. The folders should be properly pushed to the cloud platform and GitHub platforms.	On the start of the application, the web pages should load properly. The styling of the pages should appear as per mentioned in CSS file. Scrolling mechanism and navigation bar should be visible.
Input Filling	1. The Application should have been started. 2. The user should have scrolled to the Prediction section	On filling the input fields in the correct form, the user should see values in the field.
Input Submission	1. All mandatory Inputs fields should be filled	On pressing the submit button, the inputs should get submitted and user should get redirected to another webpage.

Prediction Generation & Display.	1. The Inputs should get successfully submitted. 2. Inputs should be appropriately transformed based on their type (i.e numerical or categorical) 3. The model should successfully process the inputs and generate a result. 4. The results should be encoded.	The user should be redirected to a new webpage where they should see either “Not a Backorder” if model has originally predicted 0 or “A Backorder” if model has originally predicted 1.