Predicting Back Order Situation in Supply Chain Management

Y. Fan

May 6th, 2020

1. Introduction

1.1 Background

Back order, is a definition in supply chain management, which indicates the customer order that cannot be fulfilled by the company due to a lack of stock. Back order is a common situation in a company's daily business, especially in the retail business.

Back order can upset the companies, as they might directly or indirectly lead to a decrease in revenue (due to customer order cancellation and decreased customer loyalty) and an increase in extra cost (because the company needs to pay more in purchasing the missing parts).

The company wants to avoid backorders, but also avoid overstocking for the products (which leads to higher inventory costs). And this requires the company to precisely identify the backorder situation in advance and estimate the number of items needed in inventory to satisfy customer demand.

In this study, a prediction model will be developed using the machine learning technique. The companies, especially the ones running a retail business, can be interested in this topic, because they can use the prediction results as suggestions to help them to identify potential backorder situations, so that they can take action in advance to prevent backorder in the real case and save cost in that way.

1.2 Data description

The data which is used in this analysis comes from an open source supply chain dataset (Link: https://data.world/amitkishore/can-you-predict-products-back-order).

The provided dataset consists of 2 parts (train and test part). In this study, these 2 parts of data will be first merged together to proceed further data cleaning. The data is stored in CSV file and has a volume of 1.929.937 rows and 23 columns.

1.3 Methodology

The prediction model is developed using the machine learning technique – Classification.

As classification is the method which attempts to learn the relationship between a set of feature variables and the target variable (categorical target variable such like Yes or No) and to predict the target variable based on what has been learnt, it is suitable to use classification to carry out the prediction for the backorder situation.

2. Data processing

The data is imported into dataframe in Python and the data type can be seen as in the following diagram. The feature “went_on_backorder” is the target value which indicates if a certain part will go on backorder in the future, and this is also the value that should be predicted using the data model.

```
sku                 object
national_inv       float64
lead_time          float64
in_transit_qty     float64
forecast_3_month   float64
forecast_6_month   float64
forecast_9_month   float64
sales_1_month      float64
sales_3_month      float64
sales_6_month      float64
sales_9_month      float64
min_bank           float64
potential_issue     object
pieces_past_due    float64
perf_6_month_avg   float64
perf_12_month_avg  float64
local_bo_qty       float64
deck_risk           object
oe_constraint       object
ppap_risk           object
stop_auto_buy       object
rev_stop            object
went_on_backorder   object
dtype: object
```

Before the prediction model can be built up, row data must be cleaned and transformed into certain format. The data processing part includes the following steps:

2.1 Remove duplicates

Duplicate rows are removed or merged according to the unique part number (“sku”).

2.2 Evaluate and handle missing data

Missing data is identified and handled according to the situation:

- The blank rows are removed;

- The rows with dummy “perf_6_month_avg” and “perf_12_month_avg” information are removed;

- The missing “lead_time” values are replaced with its mean value.

2.3 Convert categorical attributes to binary variables

In the dataset, the categorical attributes (“deck_risk”, “oe_constraint”, “ppap_risk”, “stop_auto_buy”, “rev_stop” and “went_on_backorder”) are converted into binary variables according to the rule Yes/No to 1/0.

2.4 Data normalization

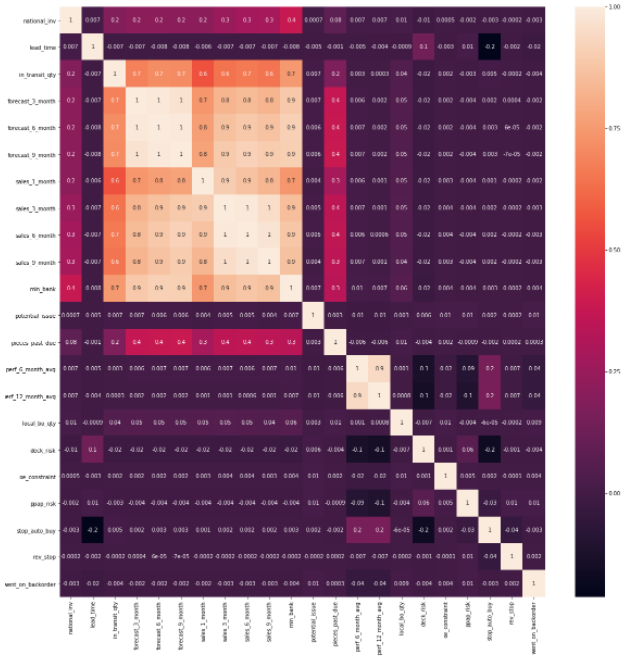
Data normalization is conducted to transform the values of different features into a similar range. For example: the variable “inventory level” has a range from 0 to 10 million (with the unit of “piece”), and the variable “lead time” only has a range from 0 to 50 (with the unit of “day”). To make sure that both features have the same impact on the result, normalization is done to convert all values to a range between 0 to 1 (using Min-Max Approach).

2.5 Dimensionality Selection

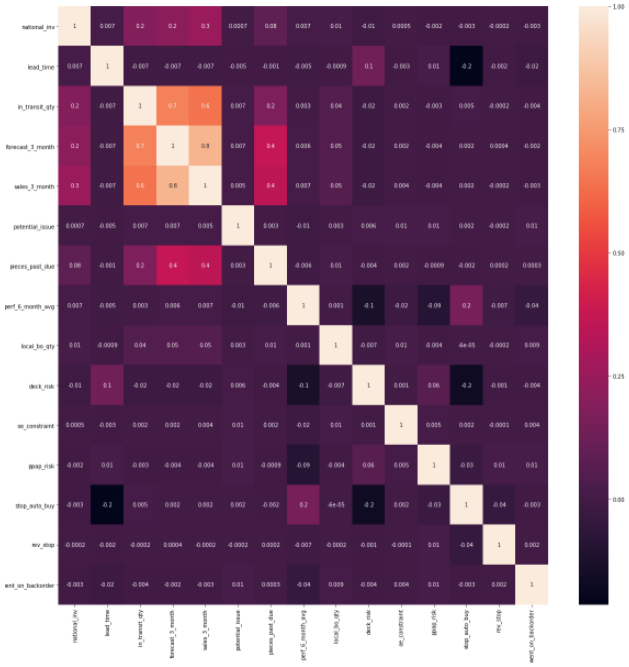
Based on the correlation heatmap, it can be shown that some features are highly correlated with each other. Considering that these features might have the same contribution to the final result, thus they have been from the scope.

Dropped features: “forecast_6_month”, “forecast_9_month”, “sales_1_month”, “sales_6_month”, “sales_9_month”, “min_bank”, “perf_12_month_avg”.

Correlation heatmap before dimensionality selection:



Correlation heatmap before dimensionality selection:



2.6 Data balancing

One of the biggest problem in the given dataset is the data imbalance: the majority class (the data with “went_on_backorder” is “No”) significantly outweighs the minority class (the data with “went_on_backorder” is “Yes”).

```
0    1173958
1      12134
Name: went_on_backorder, dtype: int64
```

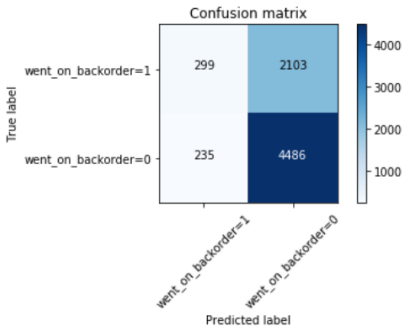
Here the undersampling method is used to randomly pick samples from the majority class, in order to achieve an equivalence in both classes (majority : minority = 2:1).

3. Prediction Modelling and Evaluation

The dataset is splitted into 2 parts: 80% is used for training and 20% is used for testing.

At first, all 14 features are included to train the model. The machine learning technique classification (Logistic Regression) is used to develop the prediction model.

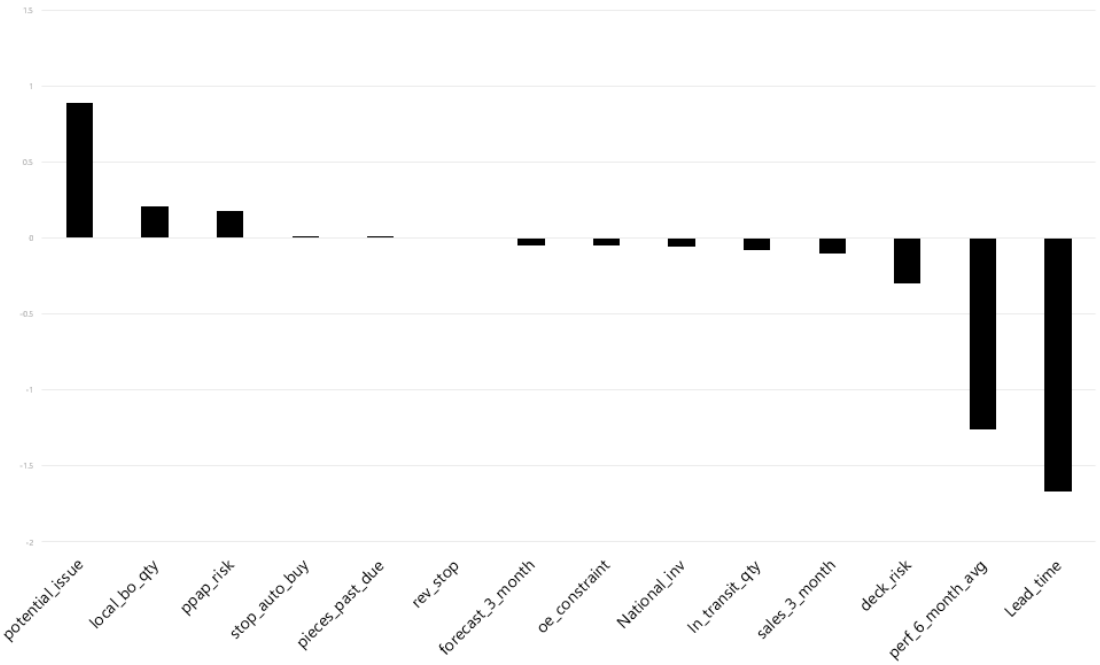
The confusion matrix below shows how the data model can predict the backorder situation:



	precision	recall	f1-score	support
0	0.68	0.95	0.79	4721
1	0.56	0.12	0.20	2402
micro avg	0.67	0.67	0.67	7123
macro avg	0.62	0.54	0.50	7123
weighted avg	0.64	0.67	0.59	7123

As shown in the graph, the data model performances quite well in predicting the “no” situation (with a f1-score of 0.79), but not that well in predicting the “yes” situation (with the f1-score of only 0.20).

Besides, the 14 features have different impacts on predicting the target value:



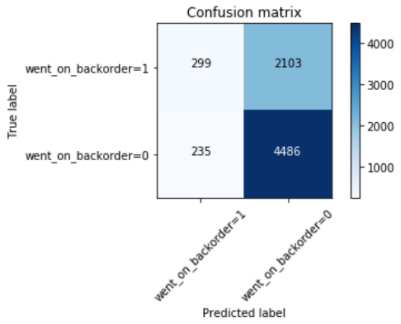
The top 7 features in predicting backorder situation are:

Lead_time, Perf_6_month_avg, Potential_issue, Deck_risk, Local_bo_qty, Ppap_risk, Sales_3_month. They are either positively or negatively correlated to the target value "went_on_backorder". For example:

- Lead time is negatively correlated with the target value. The higher the lead time (from supplier to the company) is, the less likely there would be a backorder for the part;
- Part performance in 6 month is negatively correlated with the target value. The better the part performanced in the past 6 month, the less likely there would be a backorder for the part;
- Part issue is positively correlated with the target value. If the part has more issue in the past, there tends to be more backorder situation in the future.
- Local backorder quantity is positively correlated with the target value. If there are more local backorder quantities related to the part, there tends to be more backorder situation in the future.

In order to simplify the data model without reducing its performance, in the next step, the top 7 features are selected to build the model. The classification method Logistic Regression is applied.

After the model fitting and testing, we have the following evaluation result, which is exactly the same as the one with 14 features. It shows that the top 7 features play a dominant role in predicting the backorder, whereas the other 7 features have a minimal contribution to it.



	precision	recall	f1-score	support
0	0.68	0.95	0.79	4721
1	0.56	0.12	0.20	2402
micro avg	0.67	0.67	0.67	7123
macro avg	0.62	0.54	0.50	7123
weighted avg	0.64	0.67	0.59	7123

4. Discussion

According to the evaluation result, an improvement has been achieved in predicting the backorder situation:

The model has successfully predicted ~12% of the backorder “yes” situation and ~95% of the backorder “no” situation. In these 12% / 299 cases, the companies can take action in advance such as increase the inventory level, set up demand plan to avoid backorder, winning customer orders back and saving extra cost for the companies.

However, it is obvious that the data model performances much better in predicting the “no” situation than in predicting the “yes” situation. A reason can be: there is a huge data imbalance existing in the dataset (which is mentioned in Chapter 2.6), so the data containing the “yes” information is not sufficient. The prediction model can be further improved if the data is provided in a balanced way.

Furthermore, the data model still has improvement space regarding the wrong predictions. It is considered if more part related information can be included in the dataset, then the wrong prediction ratio can be decreased.

5. Conclusion

In this study, a prediction model is built up to analyze the relationship between backorder and different features related to the part. It is identified that lead time, part performance in the past, potential issue on the part, deck risk, local backorder quantity, ppap risk, part sales in the past are the most important features that affect the target value (backorder yes or no). This data model can be very useful for the retail companies and the warehouse operaters, to help them to identify the potential backorder situation in the future, to adjust the demand and storage plan of the parts according to the prediction result, so that backorder situation can be avoided and cost can be saved.