Backorders in Supply Chain

1 Problem Statement

Detection through machine learning of probable backorders(returned orders) on e-commerce products transactions

2 Explanation

Often we see that some products that are ordered from e-commerce websites get returned back to the vendor dude to various reasons like damaged goods, customer dissatisfaction etc. This creates a lot of disruption in the logistics and supply-chain management for the e-commerce service provider . If a product gets returned back, along with the obvious transportation charges, it involves a lot of planning for storing that product in the warehouses or accounting for it in the inventory etc. Hence it will prove to be very beneficial to an e-commerce service provider if such backorders could be anticipated by them and they could account for it's disruptions beforehand to maintain the smoothness of their operations.

3 Literature Review

Backorder detection is a classic classification problem where there is a target variable; namely 'went_on_backorders' and we have to classify the order as whether it went on backorder or not. For this, there are a lot of classification models available in machine learning like RandomForests, DecisionTrees, SVM etc. Owing to the fact that the data for this problem is highly imbalanced, a simple classification model like Logistic Regression would have proved to be a poor fit for it and hence a lot of steps were needed to find a good model fit.

4 Data

Part backorders is a common supply chain problem. Working to identify parts at risk of backorder before the event occurs so the business has time to react.
Training data file contains the historical data for the 8 weeks prior to the week we are trying to predict. The data was taken as weekly snapshots at the start of each week.

Columns are defined as follows:

sku - Random ID for the product
national_inv - Current inventory level for the part
lead_time - Transit time for product (if available)
in_transit_qty - Amount of product in transit from source
forecast_3_month - Forecast sales for the next 3 months
forecast_6_month - Forecast sales for the next 6 months
forecast_9_month - Forecast sales for the next 9 months

sales_1_month - Sales quantity for the prior 1 month time period
sales_3_month - Sales quantity for the prior 3 month time period
sales_6_month - Sales quantity for the prior 6 month time period
sales_9_month - Sales quantity for the prior 9 month time period
min_bank - Minimum recommend amount to stock
potential_issue - Source issue for part identified
pieces_past_due - Parts overdue from source
perf_6_month_avg - Source performance for prior 6 month period
perf_12_month_avg - Source performance for prior 12 month period
local_bo_qty - Amount of stock orders overdue
deck_risk - Part risk flag
oe_constraint - Part risk flag
ppap_risk - Part risk flag
stop_auto_buy - Part risk flag
rev_stop - Part risk flag
went_on_backorder - Product actually went on backorder. This is the target value.
The data has 1687861 rows and 23 features with a size of around 300 MB

5 Deliverable

The deliverables for this POC are the Precision-Recall and AUC-ROC plots which show the best suitable model for this problem and data.

6 Evaluation

Evaluation was done on the basis of AUC-ROC curve and Precision-Recall plots.

7 Data Ingestion

The data was taken in the form of a csv which was downloaded as a zip folder.

8 Data Analysis

On a glimpse, the data appears to be highly imbalanced along with some impurities in it which needed to cleaned. There did not appear to be a one single model which would always fit the data very well and hence, the decision to implement multiple models and compare their performances was taken.

9 Data Munging

In this section, we identify the impurities in the data and clean them to make it suitable for further processing. The techniques applied in this is rectifying

missing data and converting the string fields to numeric form with binarization. Also we are normalizing the data to give us better insights as to what steps to take for a better modelling.
Explaining some of the strategies used in the preprocessing:

1. Slice the dataset in half, cutting off items with no forecast or sales in the past 3 months. This strategy allows us to reduce 50% the original dataset, loosing only 300 items
2. Binaries were converted from strings ('Yes' and 'No') to 1 and 0.
3. The attributes related to quantities were normalized (std dev equal to 1) per row. Therefore, parts with different order of magnitudes are approximated. For example: 1 unit of a expensive machine may be different from 1 unit of a screw, but if we standard deviate all the quantities we have, we can get a better proportion of equivalence between those items.
4. Missing values for lead_time and perf_month_avg were replaced using series median and mean.

10 Data Exploration

The rows with missing values were treated by either imputing values wherever possible or by dropping the whole row itself.

11 Feature Engineering

All the features present in the dataset contribute to the target value. Moreover, as the dataset is highly imbalanced, the correlation between predictors and target is also biased.

12 Modeling

In this section we will be treating the Imbalance and fit different models to our data We adopted here 3 estimators. Note that we do not use cross-validation or parameter tuning yet.

1. DecisionTree (CART): This is our base estimator of all methods. Decision trees present great results in imbalanced domains, since it leafs are generated using 'gini' or 'entropy', not 'accuracy' as general learning algorithms.
2. RandomUnderSampling (RUS): One of the most common sampling techniques. Rexamples the dataset by eliminating random instances of majority class.
3. RandomForest (FOREST): Bagging-based ensemble.
4. GradientBoosting (GBOOST): Boosting-based ensemble.

5. UnderBagging - A different model, not currently implemented in sklearn. It works as a bagging of emphasized textundersampled sets. We used the sklearn.ensemble.BaggingClassifier and imblearn.ensemble.RandomUnderSampler to reproduce the UnderBagging model.

13 Optimization

Hyperparameter optimization for this problem has been avoided as the data is very highly imbalanced and the HPO on the model would have had very less effect on the final results but it would have greatly increased the computation cost.

14 Prediction

The area under the ROC curve and the precision values have been provided in the plots.

15 Visual Analysis

The ROC curves and Precision-Recall plots have been used for visual analysis.

16 Results