What is Backorder ?

Backorder is what we call "out of stock" in layman
language. To be elaborate, Backorders are orders for goods that a company cannot fulfill at present because demand has outpaced supply. It may represent a product that is currently in production or it may represent one that has not yet begun production or it may represent orders partially built and waiting for a component to arrive.

Backorders should not be confused with “Out of Stock”. In case of “out of stock”, Supply or production may be uncertain. It may also be the end of the product's lifecycle and slated for discontinuance. Backorders on the other hand, are in-process or planned production that has encountered a lag due to several factors. The product will be made, it is simply not ready at the time the sales order is received.

Demerits of Backorder :-

High Frequency of backorders lead to order cancellations and thus the spending done by
the company to supply that product is wasted.
All this leads to losing market share, which can adversely impact on the company’s revenue.

Aim of the Project :-

To find out which factors are important to determine whether the product will go on backorder or not, by using a Machine Learning Model which will be trained on a dataset of 16,00,000 records and will classify products as ‘went_on_backorder’=0/1 on test data (unseen data) having 2,40,000 records.

Main aim of Model Training is to do a quantitative interpretation of features that determine the supply and demand of the product. Thus feature importance will be calculated, rather than focusing on optimizing recall/precision score.

MetaData

Data Fields	Meaning
sku	Stock Keeping Unit. (Unique number for every different kind of product.)
national_inv	Current inventory level of components. National Inv here means Inventory Level of all suppliers in a region, from where all other companies like ours purchase products.
lead_time	Purchase order lead time is the number of days from when a company places an order for supplies, to when those items arrive.
in_transit_qty	Quantity in transit.
forecast_x_month	Forecast sales for the next 3, 6, 9 months.
sales_x_month	Sales quantity for the prior 1, 3, 6, 9 months.
min_bank	Minimum recommended amount in stock.
potential_issue	Indicator variable noting potential issue with item.
pieces_past_due	Parts overdue from source.
perf_x_months_avg	Source performance in the last 6 and 12 months.
local_bo_qty	Amount of stock orders overdue.
Remaining Columns before target column	General Risk Flags/Constraints for the product's manufacturing or shipping/transportation (deck risk, oe constraint, ppap risk, stop_auto_buy, rev_stop,etc).
went_on_backorder	Product went on backorder or not.

Preprocessing Data

The Dataset is highly imbalanced Data, with 99.34% data points in class 0 and only 0.66% data of class 1.

The Dataset has many Categorical variables, therefore data was One Hot Encoded using the pandas get_dummies() function.

Challenges in the Dataset: -

1.) There are 100893 Null Values in the “lead_time” column.

2.) Columns “perf_6_month_avg” and “perf_12_month_avg” have most of values ranging from 0 to 1, but there are 1,20,000 plus data points for which “perf_6_month_avg” or “perf_12_month_avg” is ‘-99’. This irregularity was interpreted as, that the products for which the “perf_6_month_avg” or “perf_12_month_avg” was ‘-99’, meant that the product was not supplied by the current source for the last 6 or 12 months, therefore instead of 0, ‘-99’ was kept there.

To perform Feature Selection, Chi Squared Test was used between Categorical variables and the dependent variable (went_on_backorder_Yes) and dropped the variables having p value more than 0.02.

About Chi–Squared Test: -

The Chi-square test of independence is a statistical hypothesis test used to determine whether two categorical or nominal variables are likely to be related or not.
Here we used the Chi-Squared Test to quantify the relation between the categorical variables and the dependent variable.

Next, a heatmap was made which displayed correlation among all features. Many features were correlated which were either dropped or merged some features together by taking average.

--> “min_bank” was dropped as it was highly correlated with many features.
--> “forecast_3_month”, “forecast_6_month”, “forecast_9_month” were merged by taking the average, and the new column formed was named “forecast_sales”.
--> Same was done with columns like “sales_x_month” and with columns like “perf_x_month_avg”,
AVG(“sales_x_month”) = “sales_past”   AVG(“perf_x_month_avg”) = “perf_past”
--> “forecast_sales” was dropped, as it was highly correlated with “sales_past”.

Next, Null values were imputed in “lead_time” with “median” value of “lead_time”.
Median was used instead of mean because “lead_time” had many outliers, which would affect mean value.

Now we have Clean Data with No Null values in any of the columns, and where no two features are highly correlated.

Next RandomForest Model was trained with the value “balanced” in the “class_weight” attribute, which helps in nullifying the effect of imbalanced Data.

Parameters of RandomForest :-
max_samples=0.15
n_estimators=650
max_depth=10
min_samples_split=100
class_weight="balanced"

Using the above model a cross validated recall score of 0.83 was obtained.

Recall score on test data = 0.78

roc_auc_score on test data = 0.84

Then “feature_importances_” was calculated and the results came out to be :-

national_inv	51.85173
sales_past	24.99243
lead_time	5.005316
perf_past	4.665456
in_transit_qty	4.543062
local_bo_qty	4.255126
pieces_past_due	2.958315
deck_risk_Yes	1.080939
ppap_risk_Yes	0.585384
potential_issue_Yes	0.049014
oe_constraint_Yes	0.013216

It is observed that “national_inv” and “sales_past” together hold more than 75% of importance in the determination of backorder.

Further as per the above table of “feature_importances” insights were found as :-

1.) We can see that most of the products that went on backorder have “national_inv” near to 0.
2.) Products that went on backorder were having past sales generally less than 5000, while most of the products having sales much higher than 5000, didn’t go on backorder.
3.) We can see that “lead_time” of products that went on backorder is generally less than 17-18 days, while products that have very high “lead_time” generally didn’t go on backorder.

4.) We can see that products that went on backorder were having “in_transit_qty” generally less than 500. While most of the products having “in_transit_qty” higher than 500 didn’t go on backorder.