greatlearning
Learning for Life

CAPSTONE PROJECT REPORT

BACKORDER PREDICTION

PGPDSE – FT Bangalore Jul21

greatlearning

GREAT LAKES
INSTITUTE OF MANAGEMENT, CHENNAI
Global Mindset - Indian Roots

Submitted by

GR OUP 3

ANHITH ROHAN B N
ANKITA YADAV
MOHAMMED RINAS. S
PRANAV DESAI
SANDESH. S

MENTOR:

Jatinder Bedi

1. Abstract

A backorder is an order for a good or service that cannot be filled at the current time due to a lack of available supply. The item may not be held in the company's available inventory but could still be in production, or the company may need to still manufacture more of the product.The backorder is an indication that demand for a company's product outweighs its supply. They may also be known as the company's backlog.

The nature of the backorder and the number of items on backorder will affect the amount of time it takes before the customer eventually receives the ordered product. The higher the number of items backordered, the higher the demand for the item.

Backordered items often have a higher return rate. There are additional costs for expediting or air freighting backordered product, upgrading customer outbound shipments at their cost to get the backorders there faster, and for inventory control personnel spending significant time expediting product. Retaining customers can be a challenge faced by many businesses as customers might not prefer backorders in the future.

In this paper, we are trying to predict how likely the proposed product will go into backorder or not. For this, Machine Learning Algorithms are used to devise a predictive model for the imbalanced class problem. This is achieved by integrating the proposed profit-based measure into the prediction model and optimizing the decision threshold to identify the optimal backorder strategy.

Key Words- Back Order, Machine Learning, Prediction, Supply Chain Management

1.1 Industry Review - Current practices, Background Research

When a customer orders a product, which is not available in the store or temporary out of stock, and the customer decides to wait until the product is available and promised to be shipped, then this scenario is called backorder of that specific product. If backorders are not handled promptly, they will have high impacts on the respective company's revenue, share market price, customers' trust, and may end up losing the customer or sale order. On the other hand, the prompt actions to satisfy backorders put enormous pressure on different stages of the supply chain which may exhaust the supply chain processes or may appear with extra labour and/or production costs, and associated shipment expenses. Moreover, the uncertainty in customers' demands causes difficulty in forecasting the demand which makes the traditional supply chain management systems less effective in many ways such as inaccurate demand forecasting or misclassifying of back-ordered products. Nowadays, some companies predict the backorders of products by applying machine learning prediction processes to overcome the associated tangible and intangible costs of backorders.

However, a sudden change in the demand may raise other risk flags associated with the supply chain and may lead to a loss. To cope with the challenges of stochastic demand, a few researchers developed multi-objective inventory models. It has been proven mathematically that the hybrid backorder (i.e., fixed and time-weighted backorder) inventory model is more efficient than the fixed backorder inventory model in the market with volatile demand. To subside the stochastic demand problem, forecasting partial backorders based on the periodic count on the current stock level seems profitable, but this process may exhaust the local inventory system.

1.2 Literature Survey - Publications, Application, past and undergoing research

Machine Learning (ML) techniques enable us to forecast accurately multiple aspects related to supply chain management such as demand, sale, revenue, production, and backorder. ML approaches have been used to predict manufacturers' garbled demands where some researchers applied a representative set of ML-based and traditional forecasting methods to the data to compare the precision of those used methods (Carbonneau R, Vahidov R & Laframboise K, 2007).

An analysis of the supply chain's demand prediction was carried out by applying the Support Vector Regression (SVR) method in the paper of (Guanghui, 2012).

To minimize the supply chain and inventory control costs, a risk-based dynamic backorder replenishment planning framework was proposed by applying the Bayesian Belief Network (Shin K, Shin Y, Kwon JH, Kang SH, 2012).

The prediction of uncertainty in an inventory model, and proposed a framework to estimate the demand to obtain more accurate inventory decisions (Prak D & Teunter R, 2019).

Machine learning to predict and optimize product backorders using a stack-ensemble machine learning approach. The author also discussed the cost–benefit of early prediction of the backorder (Dancho M, 2017).

A Machine Learning framework for customer purchase evaluated the performance of ML models such as Lasso, Extreme learning machine, and Gradient tree boosting to forecast future purchase trends (Martínez A, Schmuck C, Pereverzyev S Jr, Pirker C & Haltmeier M, 2020).

The efficiency and impact of different types of forecasting methods were measured for promotional products in business in the research of De Baets and Harvey, 2019.

Based on the literature review, there are some research gaps. To our knowledge, just a few researchers have explored the negative values in the supply chain data. Besides, very few papers have considered flexible inventory control and cost minimization techniques separately, but none of them provided the probable backorder scenarios in inventory management.

2. Data set and Domain

The Dataset which we choose for our Capstone Project is “Backorder Prediction” belongs to the domain supply chain department.

```
# TRAINING DATASET

train_data= pd.read_csv("backorder_prediction-training data.csv")
train_data.head()
```

	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	...
0	1026827	0.0	NaN	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
1	1043384	2.0	9.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
2	1043696	2.0	NaN	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
3	1043852	7.0	8.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
4	1044048	8.0	NaN	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...

5 rows × 23 columns

We are reading the dataset 'backorder_prediction-training data' and displaying all the first five rows and columns of the dataset using head() .

2.1 Basic Statistics of Dataset

```
print('Shape of the Dataset : ', train_data.shape)
print('Number of Rows : ', train_data.shape[0])
print('Number of Features : ', train_data.shape[1])
```

```
Shape of the Dataset :  (1687861, 23)
Number of Rows :  1687861
Number of Features :  23
```

2.2 Data Dictionary

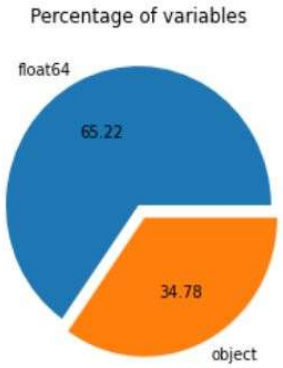
```
columns= train_data.columns

print('The Features of the dataset are :')
for i,j in enumerate(columns):
    print(i+1,'.', j)
```

```
The Features of the dataset are :
1 . sku
2 . national_inv
3 . lead_time
4 . in_transit_qty
5 . forecast_3_month
6 . forecast_6_month
7 . forecast_9_month
8 . sales_1_month
9 . sales_3_month
10 . sales_6_month
11 . sales_9_month
12 . min_bank
13 . potential_issue
14 . pieces_past_due
15 . perf_6_month_avg
16 . perf_12_month_avg
17 . local_bo_qty
18 . deck_risk
19 . oe_constraint
20 . ppap_risk
21 . stop_auto_buy
22 . rev_stop
23 . went_on_backorder
```

2.3 Feature description

Columns	Description
Sku (Categorical)	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 (Categorical)	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 (Categorical)	Whether the product is vulnerable to deck (transit) based risks.
oe_constraint(Categorical)	General part risk flag
ppap_risk (Categorical)	Whether the product is vulnerable to production quality based risks.
stop_auto_buy (Categorical)	Whether the product has automated stock warning system.
rev_stop (Categorical)	General part risk flag
went_on_backorder (Categorical)	Product actually went on backorder. This is the target value.

3. Variable categorization (count of numeric and categorical)



Numerical variables count – 15

Categorical variables count – 8

we can observe that about 65.2% of the distribution is numerical and only 34.78% of the distribution is categorical data in the Dataset.

4. Pre-Processing Data Analysis (count of missing/ null values, redundant columns, etc.)

Now, we know the shape of the data and also, regarding type of the features that are present in the data. let's get into the information regarding the datThere is function called info() in pandas which

```
train_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1687860 entries, 0 to 1687859
Data columns (total 22 columns):
 #   Column             Non-Null Count    Dtype
---  ------             --------------    -----
 0   national_inv       1687860 non-null  float64
 1   lead_time          1586967 non-null  float64
 2   in_transit_qty     1687860 non-null  float64
 3   forecast_3_month   1687860 non-null  float64
 4   forecast_6_month   1687860 non-null  float64
 5   forecast_9_month   1687860 non-null  float64
 6   sales_1_month      1687860 non-null  float64
 7   sales_3_month      1687860 non-null  float64
 8   sales_6_month      1687860 non-null  float64
 9   sales_9_month      1687860 non-null  float64
 10  min_bank           1687860 non-null  float64
 11  potential_issue    1687860 non-null  object
 12  pieces_past_due    1687860 non-null  float64
 13  perf_6_month_avg   1687860 non-null  float64
 14  perf_12_month_avg  1687860 non-null  float64
 15  local_bo_qty       1687860 non-null  float64
 16  deck_risk          1687860 non-null  object
 17  oe_constraint      1687860 non-null  object
 18  ppap_risk          1687860 non-null  object
 19  stop_auto_buy      1687860 non-null  object
 20  rev_stop           1687860 non-null  object
 21  went_on_backorder  1687860 non-null  object
dtypes: float64(15), object(7)
memory usage: 296.2+ MB
```

From the info() we get to know that,

- There are null values/missing values present in the dataset
- We have range of index from 0 to 1687859
- Total memory usage is 296.2+ MB
- There are 15 floating datatypes and 8 object. i.e., categorical

4.1 Statistics about the numerical features in the dataset

	count	mean	std	min	25%	50%	75%	max
national_inv	1687860.0	496.111782	29615.233831	-27256.0	4.00	15.00	80.00	12334404.0
lead_time	1586967.0	7.872267	7.056024	0.0	4.00	8.00	9.00	52.0
in_transit_qty	1687860.0	44.052022	1342.741731	0.0	0.00	0.00	0.00	489408.0
forecast_3_month	1687860.0	178.119284	5026.553102	0.0	0.00	0.00	4.00	1427612.0
forecast_6_month	1687860.0	344.986664	9795.151861	0.0	0.00	0.00	12.00	2461360.0
forecast_9_month	1687860.0	506.364431	14378.923562	0.0	0.00	0.00	20.00	3777304.0
sales_1_month	1687860.0	55.926069	1928.195879	0.0	0.00	0.00	4.00	741774.0
sales_3_month	1687860.0	175.025930	5192.377625	0.0	0.00	1.00	15.00	1105478.0
sales_6_month	1687860.0	341.728839	9613.167104	0.0	0.00	2.00	31.00	2146625.0
sales_9_month	1687860.0	525.269701	14838.613523	0.0	0.00	4.00	47.00	3205172.0
min_bank	1687860.0	52.772303	1254.983089	0.0	0.00	0.00	3.00	313319.0
pieces_past_due	1687860.0	2.043724	236.016500	0.0	0.00	0.00	0.00	146496.0
perf_6_month_avg	1687860.0	-6.872059	26.556357	-99.0	0.63	0.82	0.97	1.0
perf_12_month_avg	1687860.0	-6.437947	25.843331	-99.0	0.66	0.81	0.95	1.0
local_bo_qty	1687860.0	0.626451	33.722242	0.0	0.00	0.00	0.00	12530.0

- There are Outliers in the features
- Through five-point summary we get to know the count, mean, std, min, max values .
- We get to know that there are missing values by looking at the count column.

	Total Missing count	Missing value %
sku	0	0.000000
national_inv	1	0.000059
lead_time	100894	5.977625
in_transit_qty	1	0.000059
forecast_3_month	1	0.000059
forecast_6_month	1	0.000059
forecast_9_month	1	0.000059
sales_1_month	1	0.000059
sales_3_month	1	0.000059
sales_6_month	1	0.000059
sales_9_month	1	0.000059
min_bank	1	0.000059
potential_issue	1	0.000059
pieces_past_due	1	0.000059
perf_6_month_avg	1	0.000059
perf_12_month_avg	1	0.000059
local_bo_qty	1	0.000059
deck_risk	1	0.000059
oe_constraint	1	0.000059
ppap_risk	1	0.000059
stop_auto_buy	1	0.000059
rev_stop	1	0.000059
went_on_backorder	1	0.000059

- The last row of the data set is having all null values, which can be removed.
- And the rest of the null values were also dropped as the null value percentage is less than 6% of the total data.

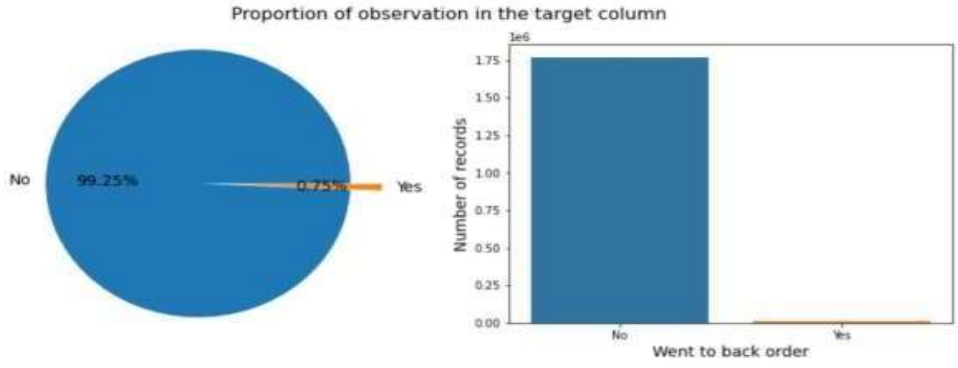
5. Project Justification Project Statement, Complexity involved, Project Outcome

5.1 Project Statement: -

Determining beforehand, whether or not a product will go to backorder or not based on the provided historical data.

5.2 Complexity Involved:

Since this is an imbalanced classification problem having positive class points very less we have to choose performance metrics accordingly. For this case false negative is a bigger concern than false-positive. Hence for this recall is more important for us than Precision. This is because it is okay to predict a product will go to back order than it is actually to back order as it can be dealt with easily. But it failed to predict product going to back order when it actually went to back order, it will be negatively impacting the company's sales and its reputation along with an additional pressure on the whole supply chain.

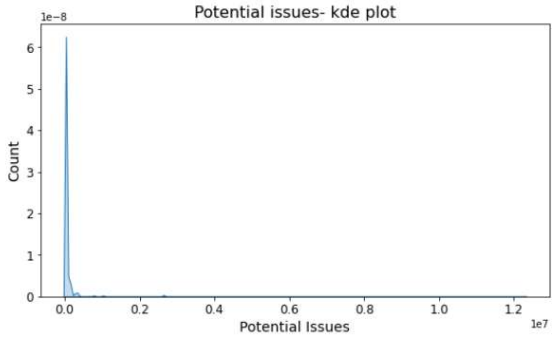


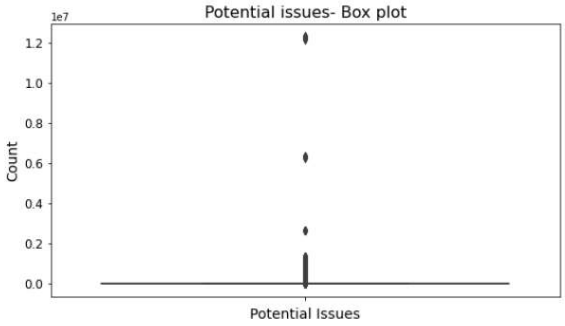
From the data proportions of the target variable, we can see that there is a high imbalance in the dataset.

6. Data Exploration (EDA)

6.1 Relation between variables - Univariate Analysis for numerical features

Numeric Feature: 'national_inv'

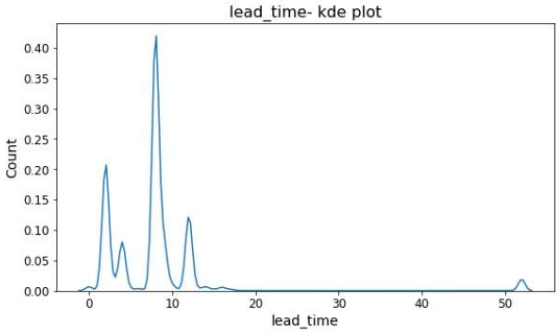


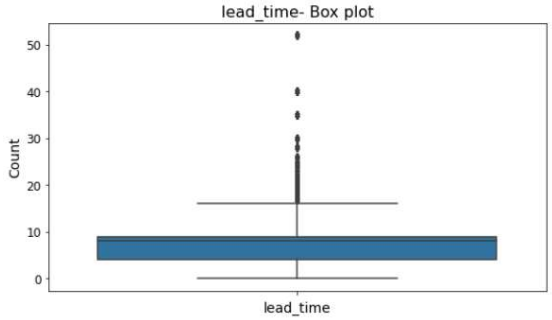



```
Skewness = 340.2858003326191
Kurtosis = 131276.59257932162
```

- IQR of the 'national_inv' is small as compared to the range of its values.
- Feature heavily right skewed as it has high positive skewedness value.
- Kurtosis value is very high implying that there are a lot of values located in the tail part of the distribution.

Numeric Feature: 'lead_time'



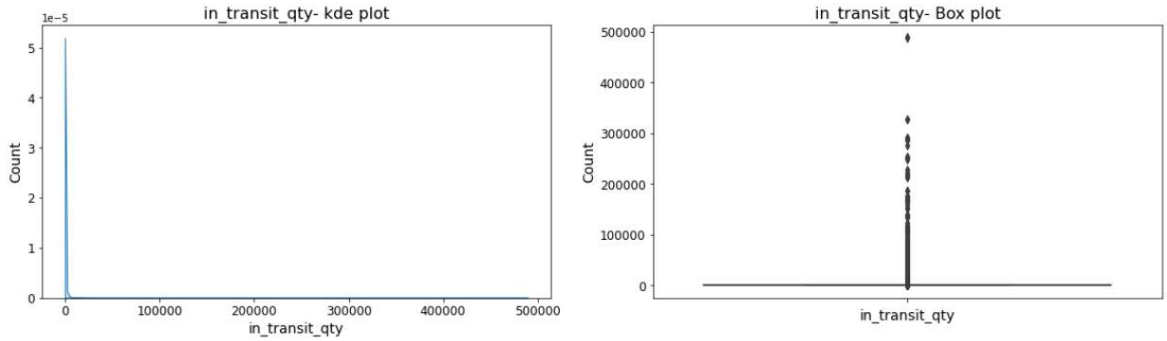



```
Skewness = 4.556295427885091
Kurtosis = 26.23722750420738
```

- IQR of the 'lead_time' is medium as compared to the range of its values.

- Feature heavily right skewed as it has high positive skewedness value.

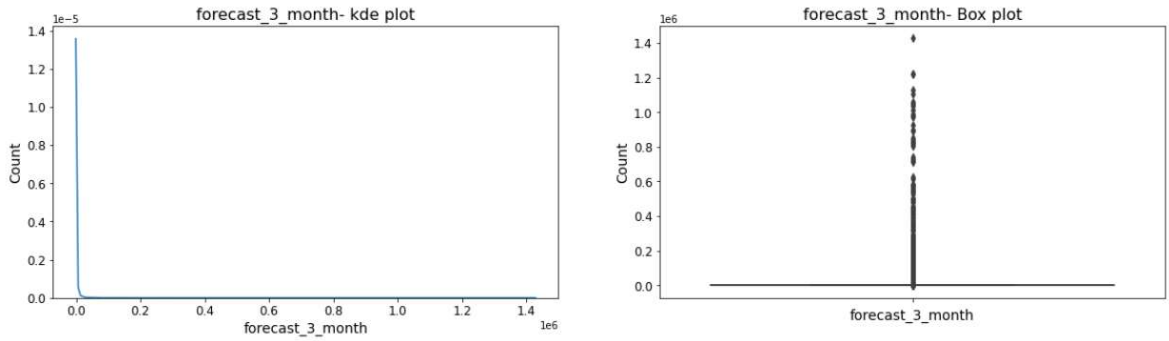
Numeric Feature: 'in_transit_qty'




```
Skewness = 166.18340424761558
Kurtosis = 39606.10405290813
```

- IQR of the 'in_transit_qty' is medium as compared to the range of its values.
- Feature heavily right skewed as it has high positive skewedness value.
- Kurtosis value is very high implying that there are a lot of values located in the tail part of the distribution.

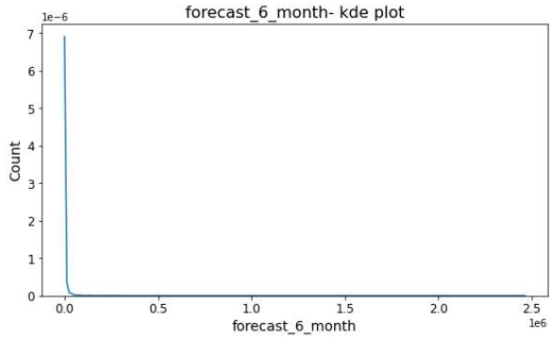
Numeric Feature: 'forecast_3_month'

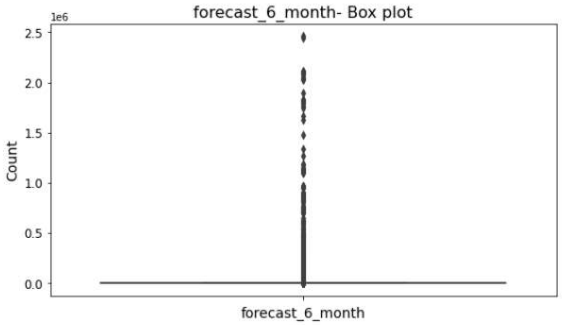



```
Skewness = 138.96832519579834
Kurtosis = 25637.55029993227
```

- IQR of the 'forecast_3_month' is small as compared to the range of its values.
- Feature heavily right skewed as it has high positive skewedness value.
- Kurtosis value is very high implying that there are a lot of values located in the tail part of the distribution.

Numeric Feature: 'forecast_6_month'

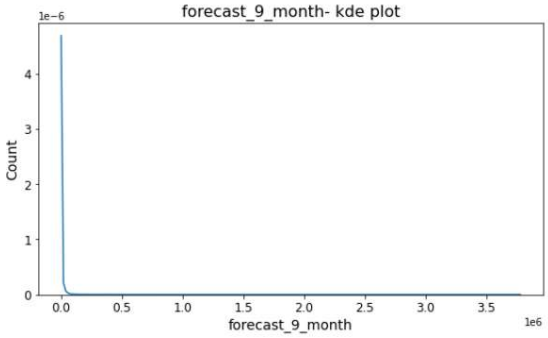


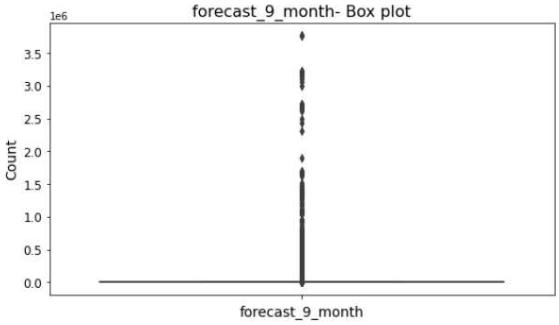



```
Skewness = 138.96142721254265
Kurtosis = 25189.903788272073
```

- IQR of the 'forecast_6_month' is small as compared to the range of its values.
- Feature heavily right skewed as it has high positive skewedness value.
- Kurtosis value is very high implying that there are a lot of values located in the tail part of the distribution.

Numeric Feature: 'forecast_9_month'

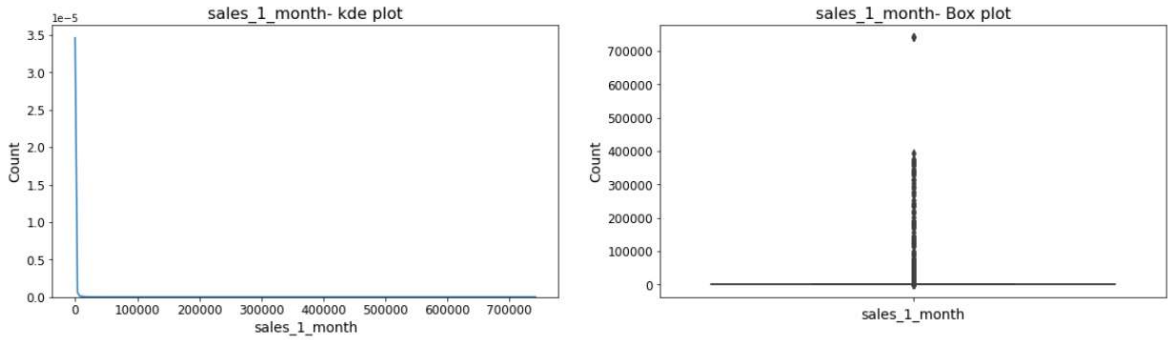





```
Skewness = 143.298874740098
Kurtosis = 27048.452312581445
```

- IQR of the 'forecast_9_month' is small as compared to the range of its values.
- Feature heavily right skewed as it has high positive skewedness value.
- Kurtosis value is very high implying that there are a lot of values located in the tail part of the distribution.

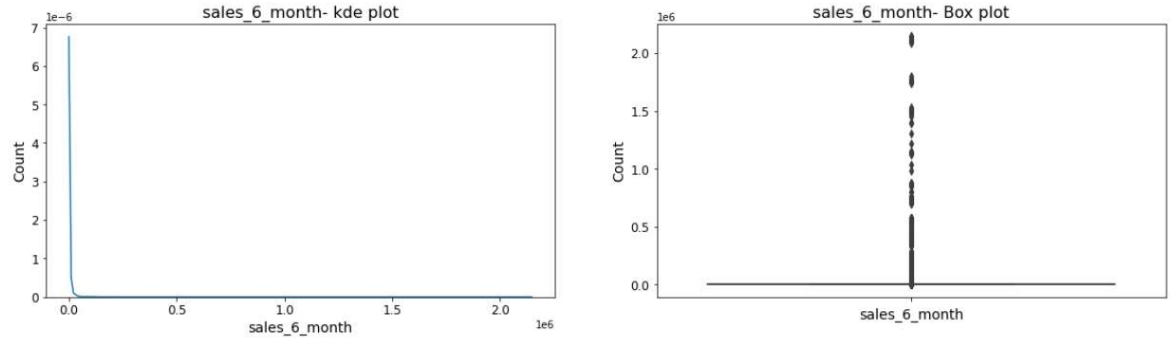
Numeric Feature: 'sales_1_month'




```
Skewness = 196.1199898556541
Kurtosis = 53855.92556025887
```

- IQR of the 'sales_3_month' is very small as compared to the range of its values.
- Feature heavily right skewed as it has high positive skewedness value.
- Kurtosis value is very high implying that there are a lot of values located in the tail part of the distribution.

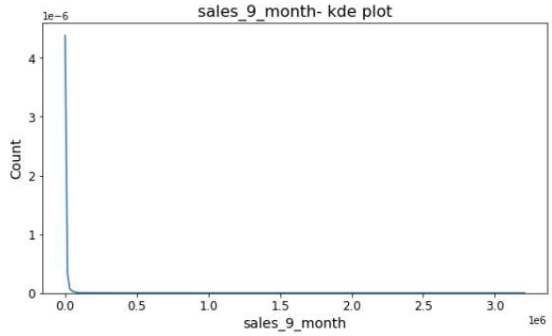
Numeric Feature: 'sales_6_month'

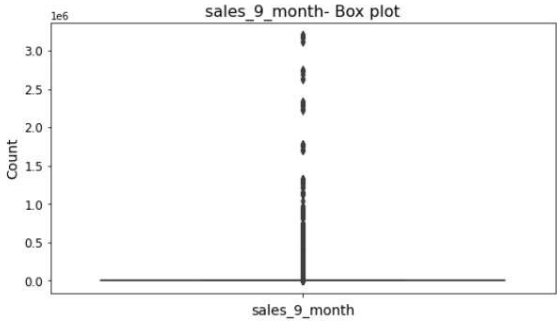



```
Skewness = 139.17671201086372
Kurtosis = 24305.44501338931
```

- IQR of the 'sales_6_month' is very small as compared to the range of its values.
- Feature heavily right skewed as it has high positive skewedness value.

Numeric Feature: 'sales_9_month'

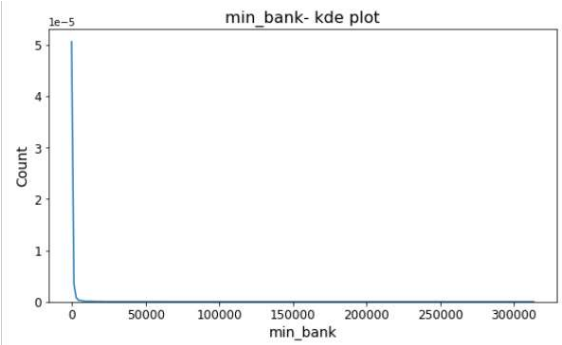


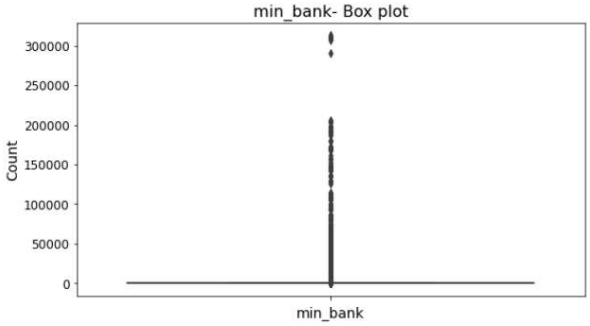



```
Skewness = 135.05419147168155
Kurtosis = 22844.80574661239
```

- IQR of the 'sales_9_month' is very small as compared to the range of its values.
- Feature heavily right skewed as it has high positive skewedness value.
- Kurtosis value is very high implying that there are a lot of values located in the tail part of the distribution.

Numeric Feature: 'min_bank'

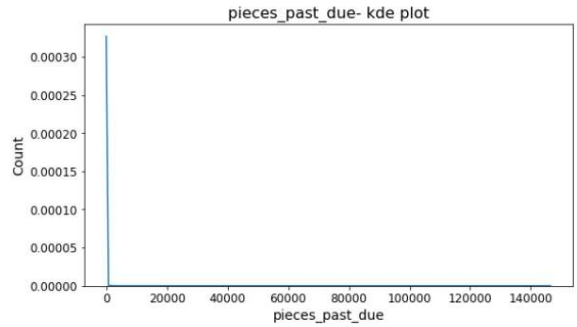


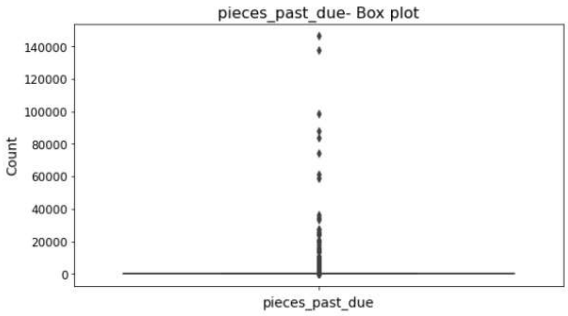



```
Skewness = 131.21264893012795
Kurtosis = 23549.240091008585
```

- IQR of the 'min_bank' is very small as compared to the range of its values.
- Feature heavily right skewed as it has high positive skewedness value.
- Kurtosis value is very high implying that there are a lot of values located in the tail part of the distribution.

Numeric Feature: 'pieces_past_due'

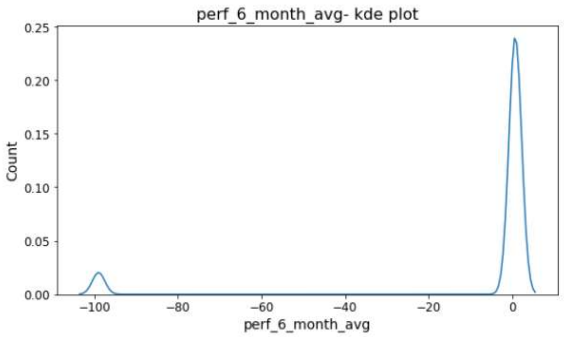


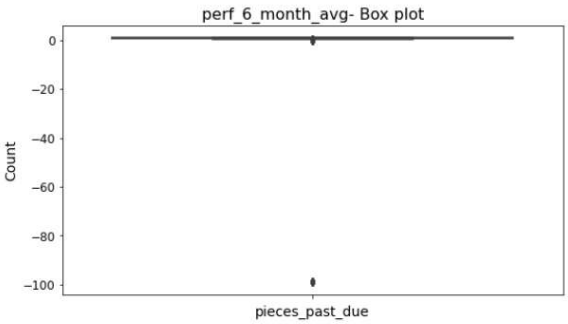



```
Skewness = 412.39190039252696
Kurtosis = 207663.2258415861
```

- IQR of the 'pieces_past_due' is very small as compared to the range of its values.
- Feature heavily right skewed as it has high positive skewedness value.
- Kurtosis value is very high implying that there are a lot of values located in the tail part of the distribution.

Numeric Feature: 'perf_6_month_avg'

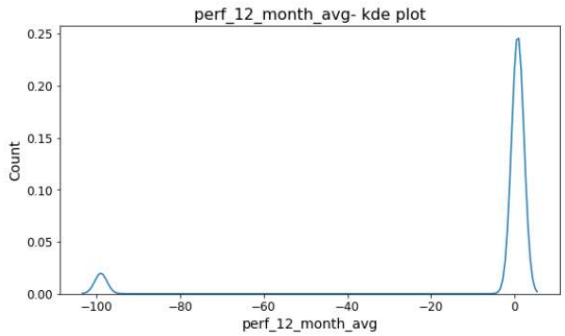


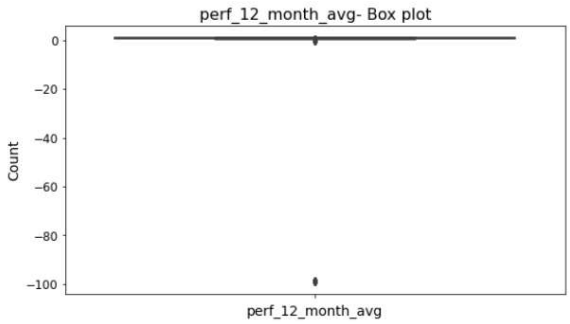



```
Skewness = -3.180621807495058
Kurtosis = 8.11739511529576
```

- IQR of the 'perf_6_month_avg' is large as compared to the range of its values.
- Feature left skewed as it has low negative skewedness value.
- Kurtosis value is low implying that there are a some values located in the tail part of the distribution.

Numeric Feature: 'perf_12_month_avg'

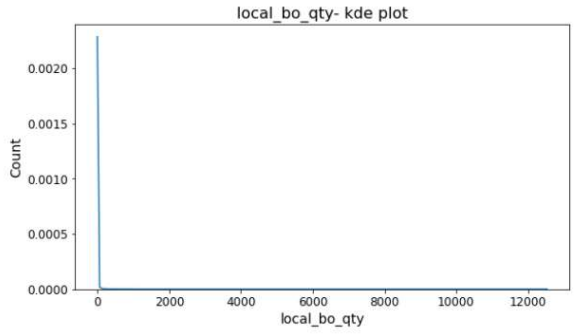


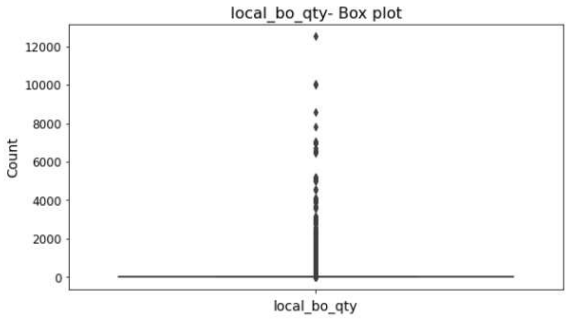



```
Skewness = -3.3021812484797537
Kurtosis = 8.905503219179664
```

- IQR of the 'perf_12_month_avg' is large as compared to the range of its values.
- Feature left skewed as it has low negative skewedness value.
- Kurtosis value is low implying that there are a some values located in the tail part of the distribution.

Numeric Feature: 'local_bo_qty



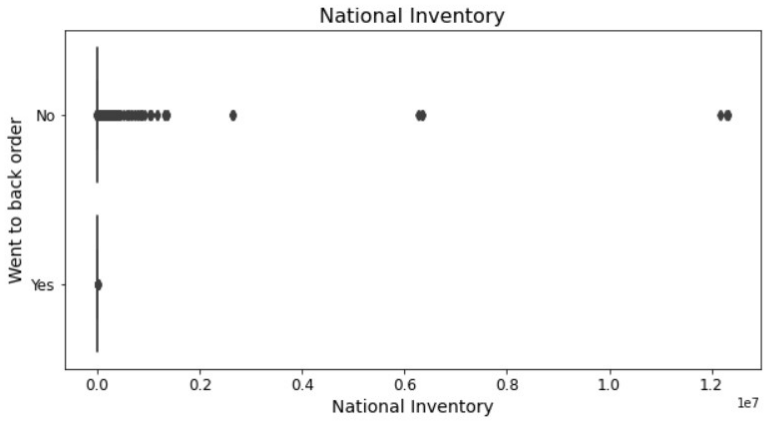



```
Skewness = 165.19054793748316
Kurtosis = 38154.955457397235
```

- IQR of the 'local_bo_qty' is very small as compared to the range of its values.
- Feature heavily right skewed as it has high positive skewedness value.
- Kurtosis value is very high implying that there are a lot of values located in the tail part of the distribution.

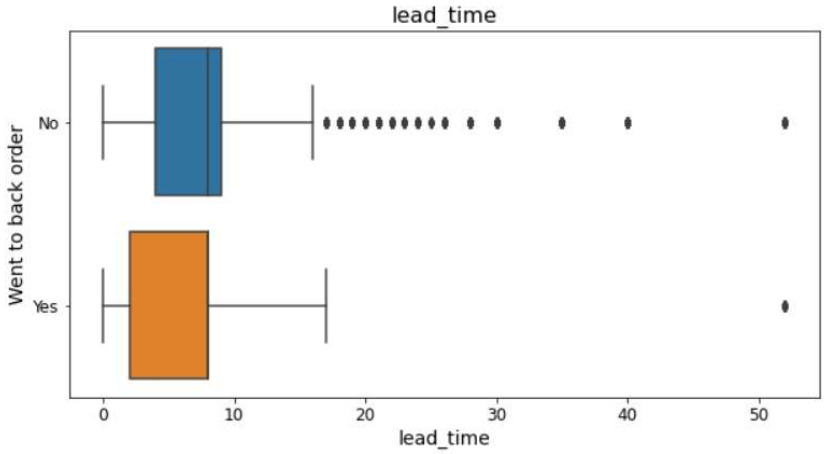
6.2 Relation between variables - Bi-variate Analysis for numerical features

Numerical Feature : 'National Inventory'



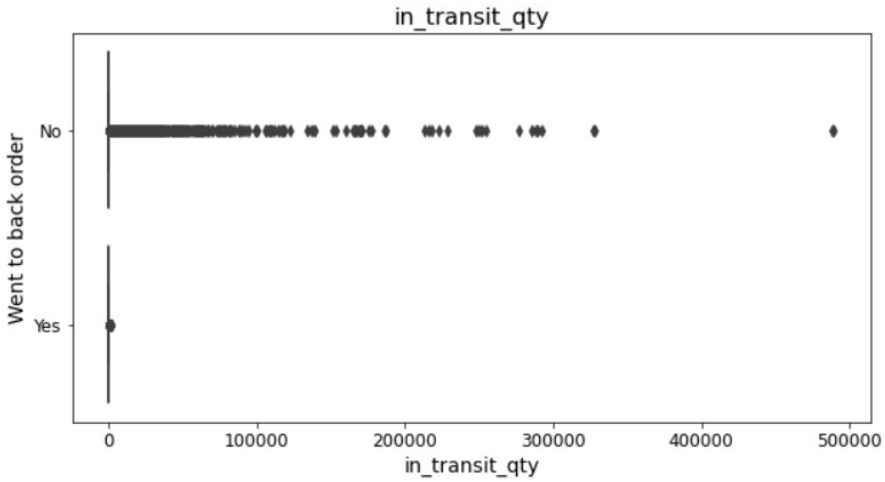
- When we split the plots based on the target variable, we can see that the extent of outliers are more for the No category as compared to Yes. We can see that the data for the yes category has the highest points near 0.
- Here we can see that when the product goes to back order, the national inventory is 0. That is there is high chance for the product to go to back order when the national inventory is 0.

Numerical Feature : 'lead_time'



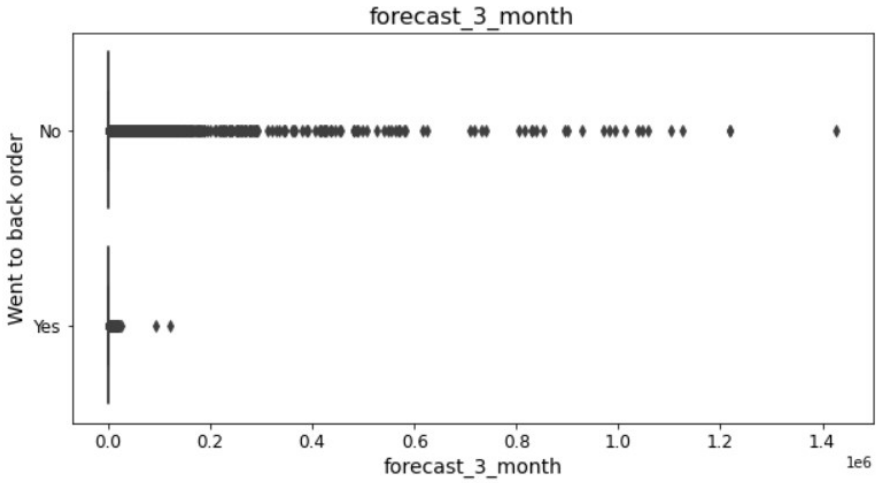
- The data split based on target variable show a similar pattern.
- For both the classes, IQR is overlapping. Median values are overlapping .
- Both the data have outliers with similar values.

Numerical Feature : 'In Transit quantity'



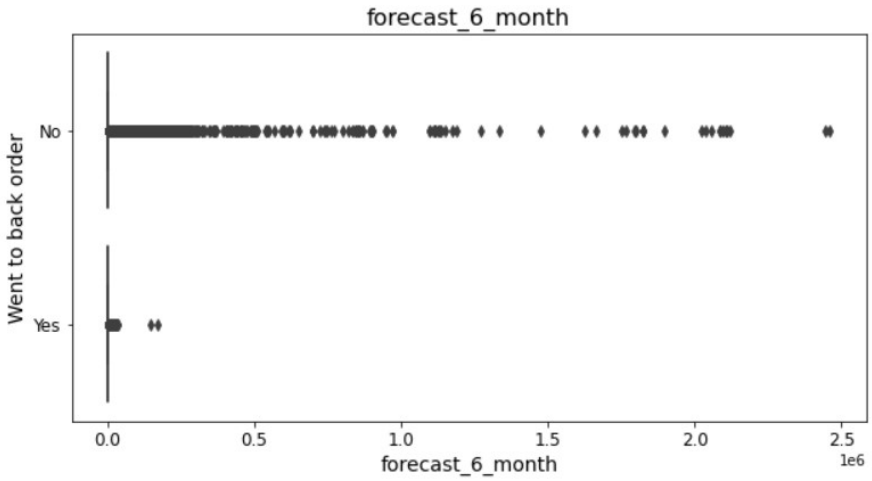
- The in transit quantity for went to back order is most focused at zero. This is because, the feature in transit quantity sincerely refers to the products which can be already available, if it is available then probabilities of the product going to returned order could be very less.

Numerical Feature : 'forecast_3_month'



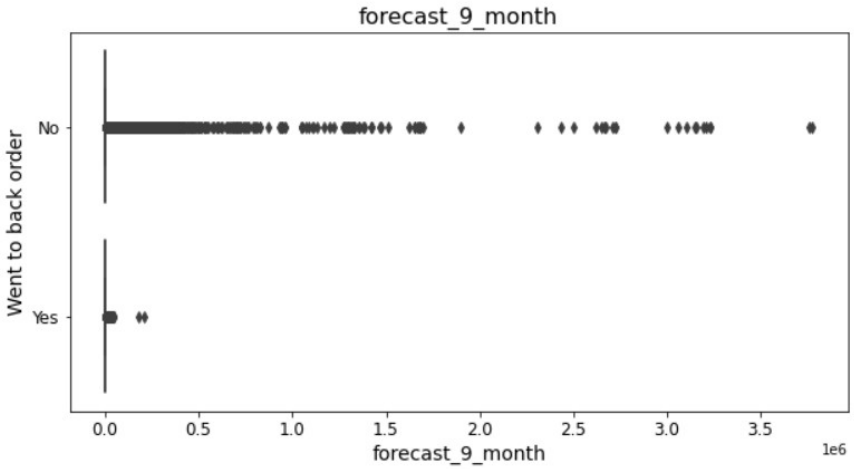
When split based on the target variable, the no category has large value of outliers present as compared to the yes category.

Numerical Feature : 'forecast_6_month'



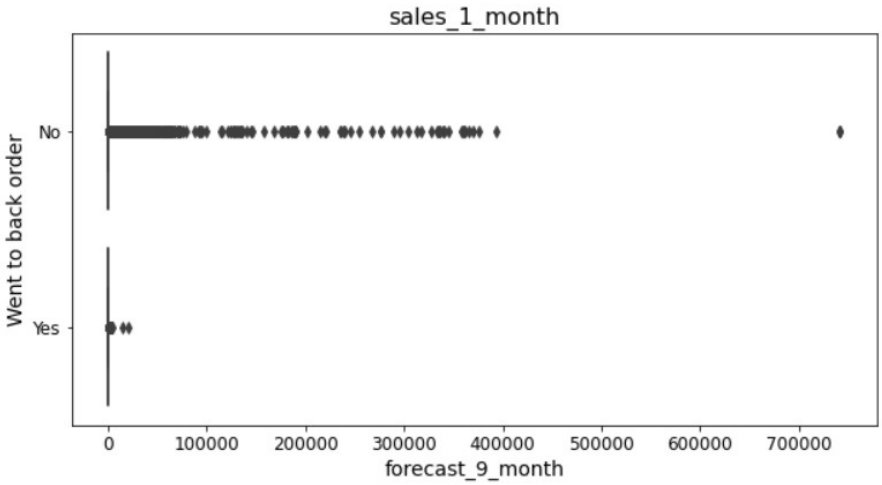
- When split based on the target variable, the no category has large value of outliers present as compared to the yes category.

Numerical Feature : 'forecast_9_month'



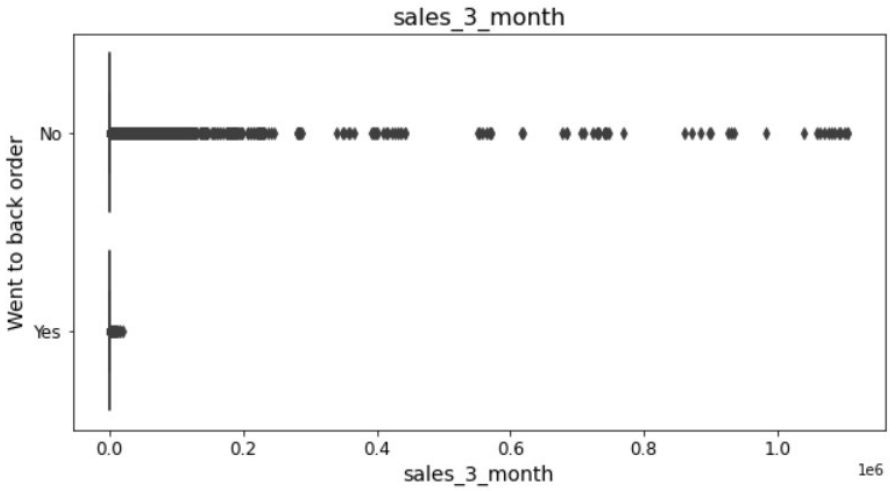
- When split based on the target variable, the no category has large value of outliers present as compared to the yes category

Numerical Feature : 'sales_1_month'



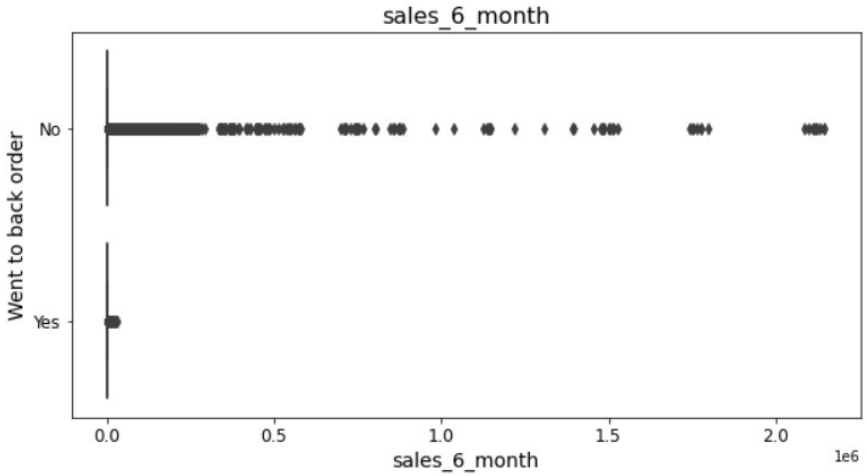
- On the target variable split ,We can clearly see the large number of observation near zero for both the categories
- The yes category has less outlier and where as the no category outliers are widely spread,
- The Yes category has more number of records at 0 value as compared to No category.

Numerical Feature : 'sales_3_month'



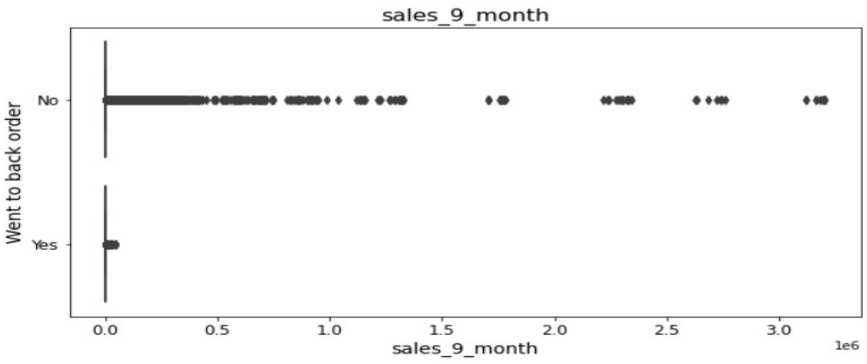
- On the target variable split ,We can clearly see the large number of observation near zero for both the categories
- The yes category has less outlier and where as the no category outliers are widely spread,
- The Yes category has more number of records at 0 value as compared to No category.

Numerical Feature : 'sales_6_month'



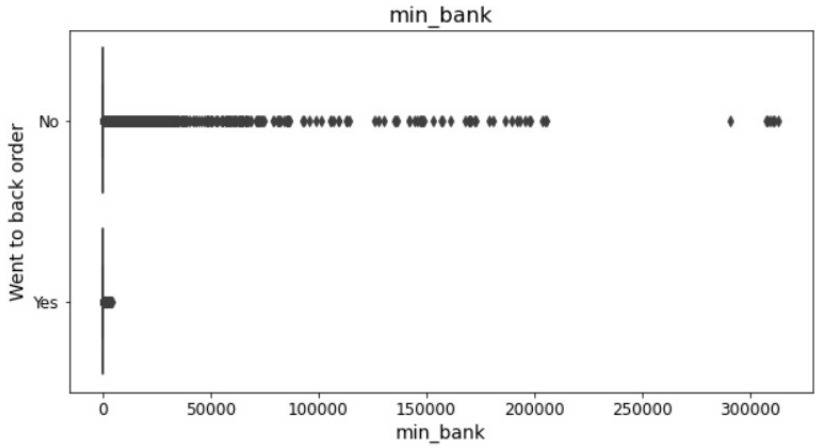
- On the target variable split ,We can clearly see the large number of observation near zero for both the categories
- The yes category has less outlier and where as the no category outliers are widely spread,
- The Yes category has more number of records at 0 value as compared to No category.

Numerical Feature : 'sales_9_month'



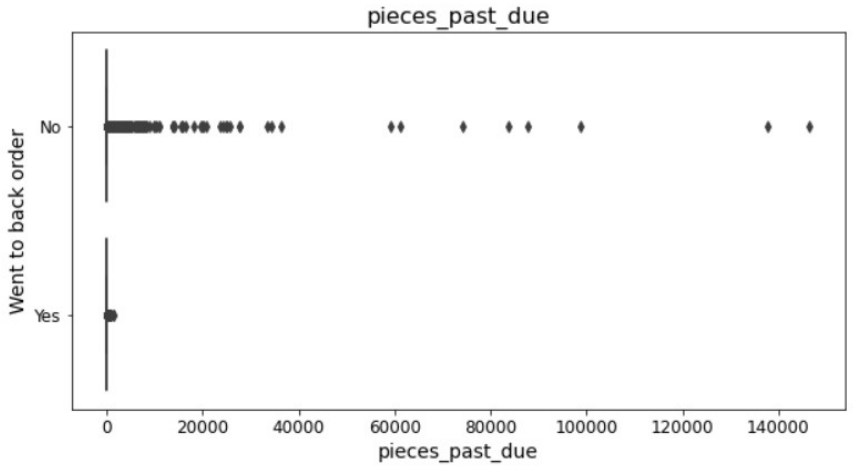
- On the target variable split ,We can clearly see the large number of observation near zero for both the categories
- The yes category has less outlier and where as the no category outliers are widely spread,

Numerical Feature : 'min_bank'



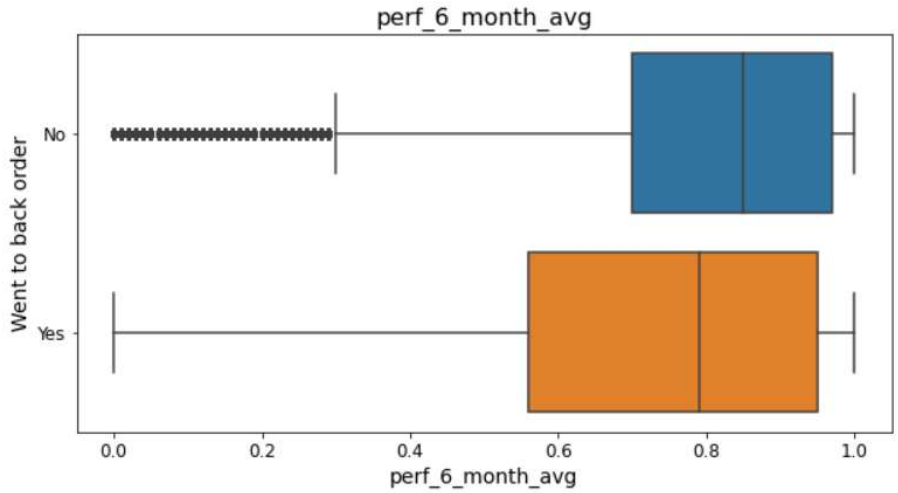
- On the target variable split ,We can clearly see the large number of observation near zero for both the categories
- The yes category has less outlier and where as the no category outliers are widely spread,
- The Yes category has more number of records at 0 value as compared to No category.
- We are able to see that the goods that went to back orders had maximum of the observations with zero as minimal recommended stock.

Numerical Feature : 'pieces_past_due'



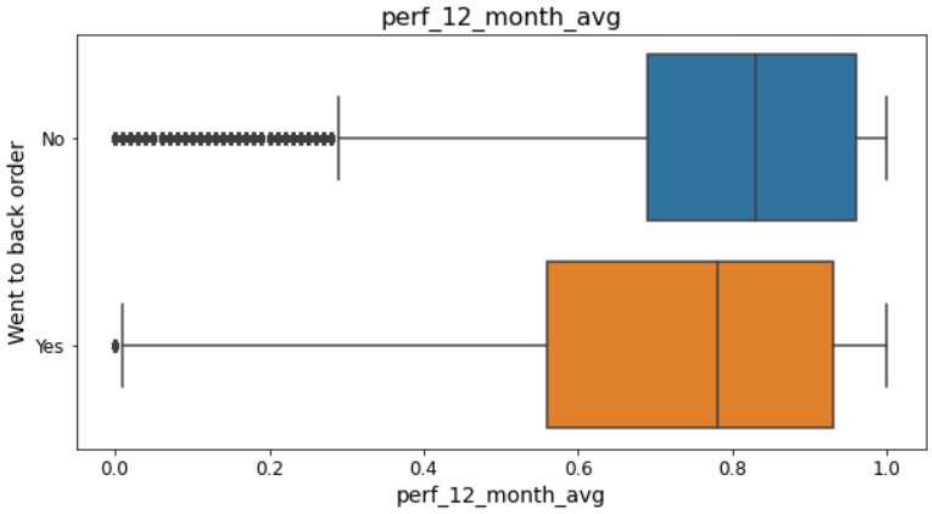
- On the target variable split ,We can clearly see the large number of observation near zero for both the categories
- The yes category has less outlier and where as the no category outliers are widely spread,
- The Yes category has more number of records at 0 value as compared to No category.

Numerical Feature : 'perf_6_month_avg'



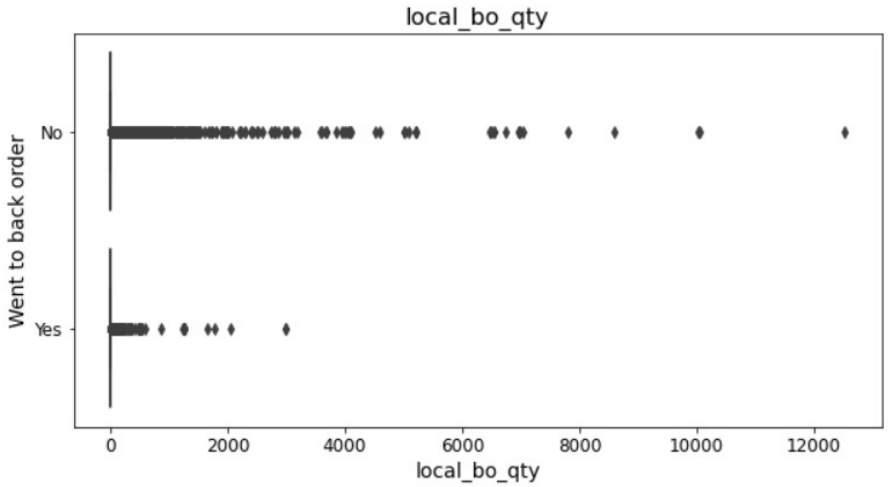
- We can see that No category has all the outliers and yes category has no outliers at all

Numerical Feature : 'perf_12_month_avg'



- We can see that No category has all the outliers and yes category has no outliers at all

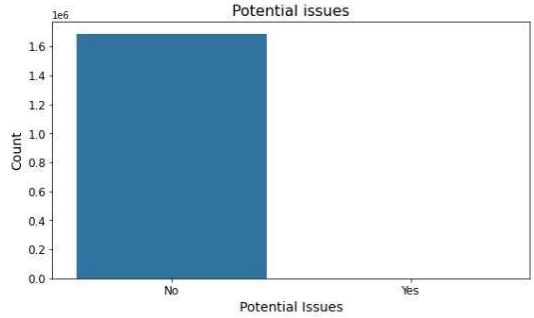
Numerical Feature : 'local_bo_qty'

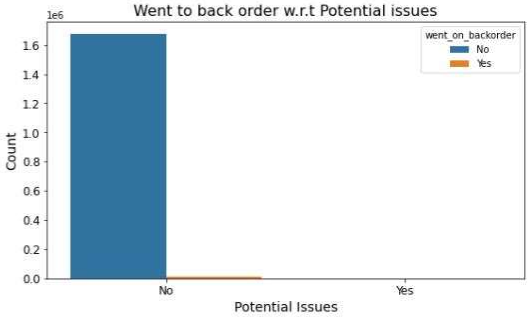


- There are outliers present in both the classes. The outliers for the No class of the target variable are more dispersed.
- We can observe that in both the categories most records are located near 0.

6.2 Relation between variables - Analysis for Categorical features

Feature : Potential Issues

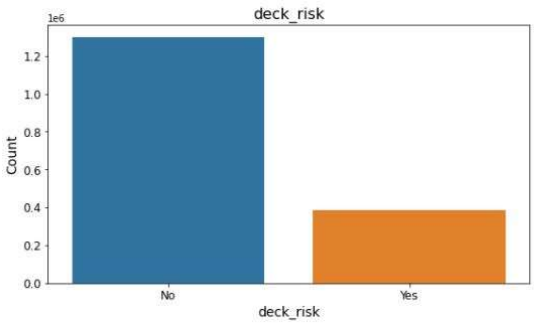


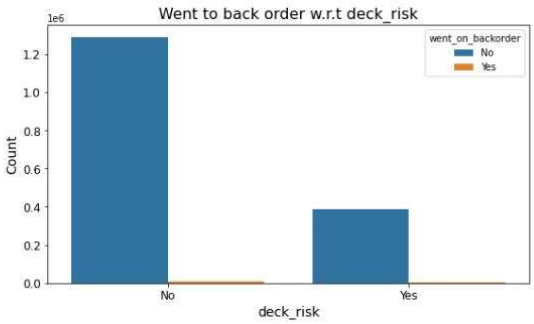


went_on_backorder	No	Yes
potential_issue		
No	99.3	0.7
Yes	94.4	5.6

- When 'potential_issue' is 'No', 0.7% products went to backorder.
- When 'potential_issue' is 'Yes', 5.6% products went to backorder.
- When 'potential_issue' is 'Yes', there 8 times more probability of product going to backorder.

Feature : Deck risk

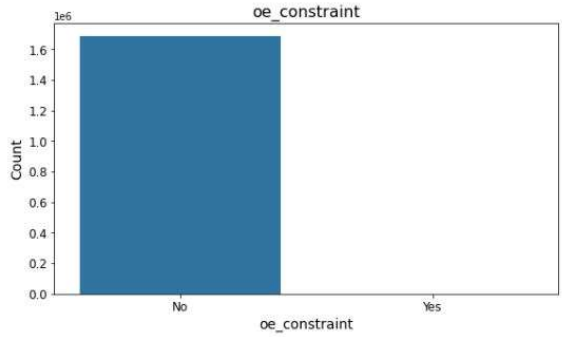


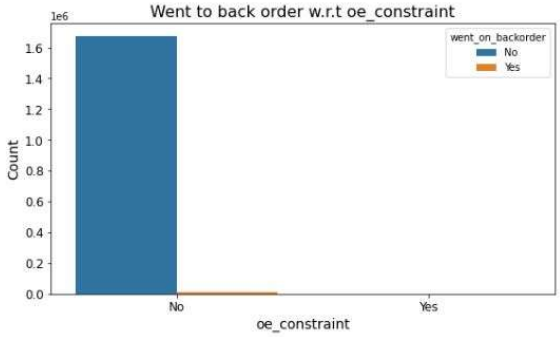


went_on_backorder	No	Yes
deck_risk		
No	99.3	0.7
Yes	99.5	0.5

- When 'deck_risk' is 'No', 0.7% products went to backorder.
- When 'deck_risk' is 'Yes', 0.5% products went to backorder.
- Since distribution of points, when product went to backorder is almost similar in this case, this feature doesn't seem helpful in determining whether or not product went to backorder.

Feature :OE Constraint

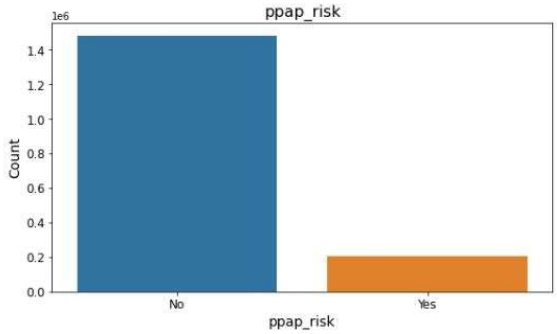


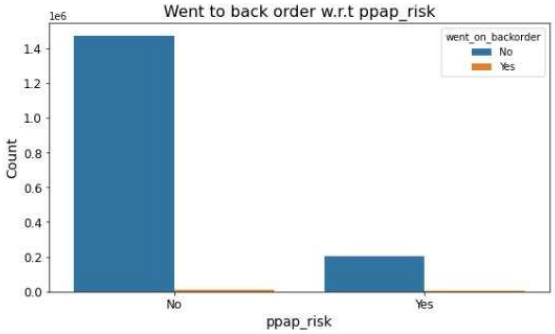


went_on_backorder	No	Yes
oe_constraint		
No	99.3	0.7
Yes	96.7	3.3

- When 'oe_constraint' is 'No', 0.7% products went to backorder.
- When 'oe_constraint' is 'Yes', 3.3% products went to backorder.
- So, it means it 'oe_constraint' value is 'Yes' there are approx 5 times more probability of product going to backorder. Hence, this feature is important.

Feature : ppap_risk



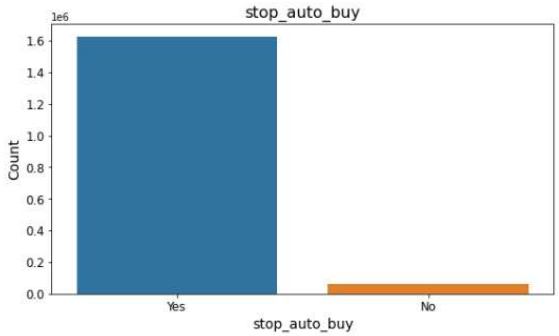


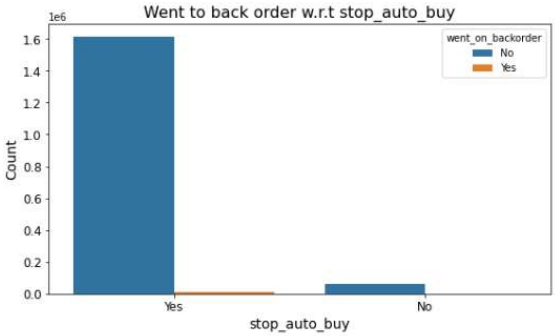
went_on_backorder	No	Yes
ppap_risk		
No	99.4	0.6
Yes	99.1	0.9

- When 'ppap_risk' is 'No', 0.6% of product goes to backorder.
- When 'ppap_risk' is 'Yes', 0.9% of product goes to backorder.

This feature also doesn't seem to be helpful as it is not providing much distinction between product went to backorder or not.

Feature : stop_auto_buy

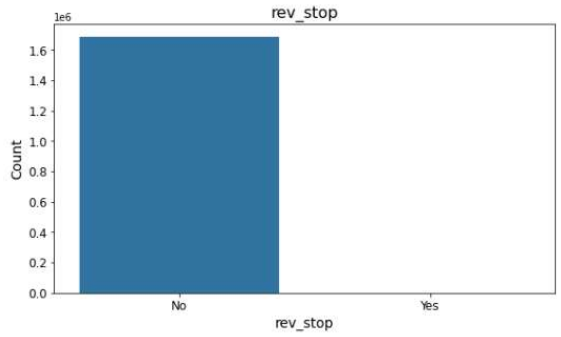


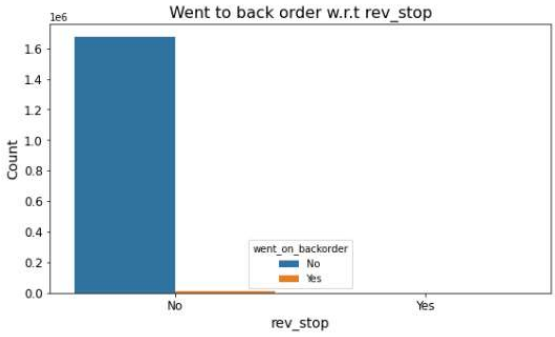


went_on_backorder	No	Yes
stop_auto_buy		
No	99.2	0.8
Yes	99.3	0.7

- When 'stop_auto_buy' was 'Yes', about 0.7% of the products went to backorder.
- When 'stop_auto_buy' was 'No', about 0.8% of the products went to backorder.
- This feature is not helping much in classification, as it have almost equal distribution of poi

Feature : Rev Stop





went_on_backorder	No	Yes
rev_stop		
No	99.3	0.7
Yes	100.0	0.0

- When 'rev_stop' is 'Yes', no product went ot backorder.
- When 'rev_stop' is 'No', about 0.7% of products went to backorder.

- This feature could be helpful in classification, as products goes to backorder when it set to 'No' only.

7. Statistical significance of variable:

The numerical features were checked for normal distribution using Jarque Bera test.
The features which were normally distributed were tested using ttest_ind.
The features which were not normally distributed were tested using manwhitneyu test.
For categorical features, since only 2 categories are present in each feature, proportions_ztest was performed. The results are as follows:

Significant numerical Features:

national_inv,	forecast_9_month,
lead_time,	forecast_9_month,
in_transit_qty,	forecast_9_month,
forecast_3_month,	sales_3_month,
forecast_6_month,	sales_6_month,
pieces_past_due,	sales_9_month,
perf_6_month_avg,	perf_12_month_avg,

Significant categorical features:

potential_issue	deck_risk
oe_constraint	ppap_risk
stop_auto_buy	

Significant Categorical Variable

There are 2 variables which are insignificant based on the statistical tests. As statistical test are stringent, they are not being dropped at this point of time and will be further analysed before modelling.

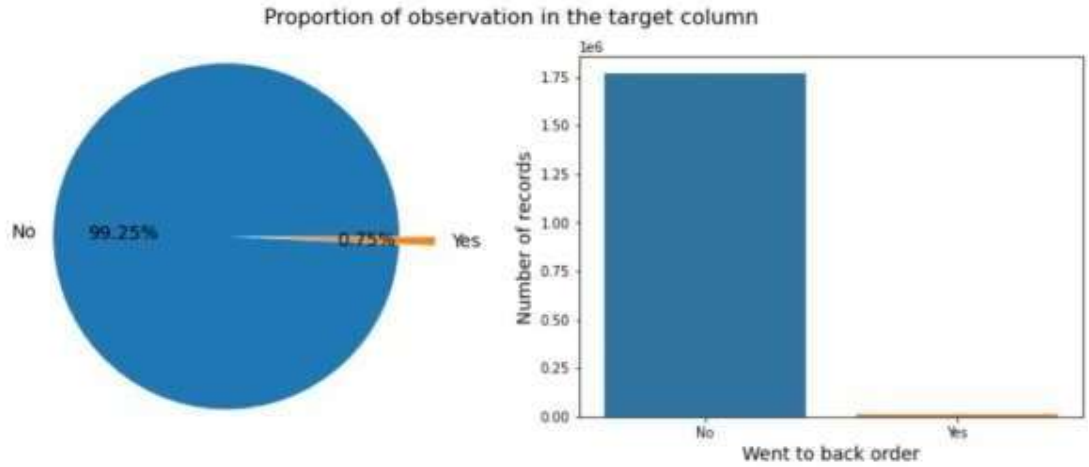
	Features	pvalue
0	national_inv	0.000000e+00
1	lead_time	3.796171e-155
2	in_transit_qty	9.875747e-128
3	forecast_3_month	0.000000e+00
4	forecast_6_month	0.000000e+00
5	forecast_9_month	0.000000e+00
6	sales_1_month	0.000000e+00
7	sales_3_month	0.000000e+00
8	sales_6_month	0.000000e+00
9	sales_9_month	0.000000e+00
10	pieces_past_due	0.000000e+00
11	perf_6_month_avg	2.519334e-158
12	perf_12_month_avg	6.169752e-169
13	local_bo_qty	0.000000e+00

	Features	P_value
0	sku	4.996234e-01
1	potential_issue	2.751637e-72
2	deck_risk	1.786458e-18
3	oe_constraint	8.663448e-06
4	ppap_risk	1.760307e-34
5	stop_auto_buy	3.945919e-05
6	rev_stop	1.947259e-01

- Here almost every Numerical features seems significant, where as in categorica ‘rev_stop’ and ‘sku’ seems insignificant.
- The numerical features were checked for normal distribution using Jarque Bera test.
- The features which were normally distributed were tested using t_test_ind.
- The features which were not normally distributed were tested using manwhitneyu test.
- The categorical features were tested using Chi-square contingency test.

8. Class imbalance and its treatment:

As we can see that the majority class is having 99.33 % and minority is having 0.75%, our data is highly imbalanced. This has to be treated. Class imbalance can be handled in two ways. Balancing the dataset using different Oversampling or Under sampling techniques .



From the data proportions of the target variable, we can see that there is a high imbalance in the dataset.

9. Feature Engineering:

- Transformation techniques like square root, cube root, log1p, power transformation will be applied and results verified before finalising.
- Feature selection – using KBest, Recursive Feature Elimination or Sequential Feature Selection.
- Dimensionality reduction using PCA can be considered.
- SMOTE or oversampling of train data to deal with imbalance.

Sampling:

As the dataset is enormous, only a 5% sample from the total data is considered and statistical tests were conducted to confirm that the sample data was representing the original data. Hence, going forward, only this sample data will be considered for base model and final modelling.

Number of rows in 5% sample data = 89068

Number of columns in 5% sample data = 23

10.Base Model with raw data:

Base model was done on the raw data without any treatment or feature engineering. Two base models have been considered:

a. Logistic Regression
b. Random Forest Classifier

Due to the imbalance in the data, our metric for evaluation will be **F1 Score**
Metrics Considered: F1 Score, R

1.Logistic Regression Performance:

Train F1_score = 0.00101	Test_F1_score = 0.0
Train ROC AUC score = 0.7263	Test ROC AUC score = 0.6644
Train Accuracy = 0.99194	Test Accuracy = 0.99197

: Performance of Logistic regression as base model

The accuracy score of the base model is very high. But we can see the f1_score is absolutely horrendous. The sheer imbalance of the data, the model has completely classified all the went to back order - yes class as no. And since the number of yes class is so few the accuracy score is not affected by that. But f1 score being the harmonic mean of the precision and recall, it clearly shows that model is just acting like a null model and giving the output as class the higher probability. Hence, we conclude that we have to perform data balancing techniques

2.Random Forest Classifier performance:

Train F1_score = 0.986	Test_F1_score = 0.04
Train ROC AUC score = 0.999	Test ROC AUC score = 0.888
Train Accuracy = 0.999	Test Accuracy = 0.991

Performance of Random Forest as base model.

From the results we can see that random forest has overfitted the train data. The test data performance is poor. Like the Logistic Regression the imbalance has affected the model really bad, though it performed slightly better than the Logistic regression model. The test F1 Score is very low hence we need to proceed with the data treatment.

11.Data Treatment

11.Treatment of Class Imbalance:

As discussed before, the data is highly imbalanced because of which our base model f1 score is very low, which is our prime metric for evaluation and hence SMOTE technique to balance the data.

SMOTE:

SMOTE is an oversampling technique to synthesize training samples in the feature space. It is based on nearest neighbours judged by Euclidean Distance between data points in feature space. The major problem with an imbalanced dataset is that it creates a bias towards the majority class and result in poor overall performance of the model. Therefore, an imbalanced data must be balanced before model building and SMOTE is one of the techniques to make the dataset balanced by synthesizing training samples near the minority class samples in the feature space.

11.2Outliers Treatment:

As data is highly skewed, it has lots of outliers present in it. Removing outliers through IQR method doesn't reduce the skewness and also leads to more imbalance in the data. Hence power transformation is applied on the data which has reduced the skewness mostly and also t reated the outliers.

Skewness after Outliers treatment is as follows:

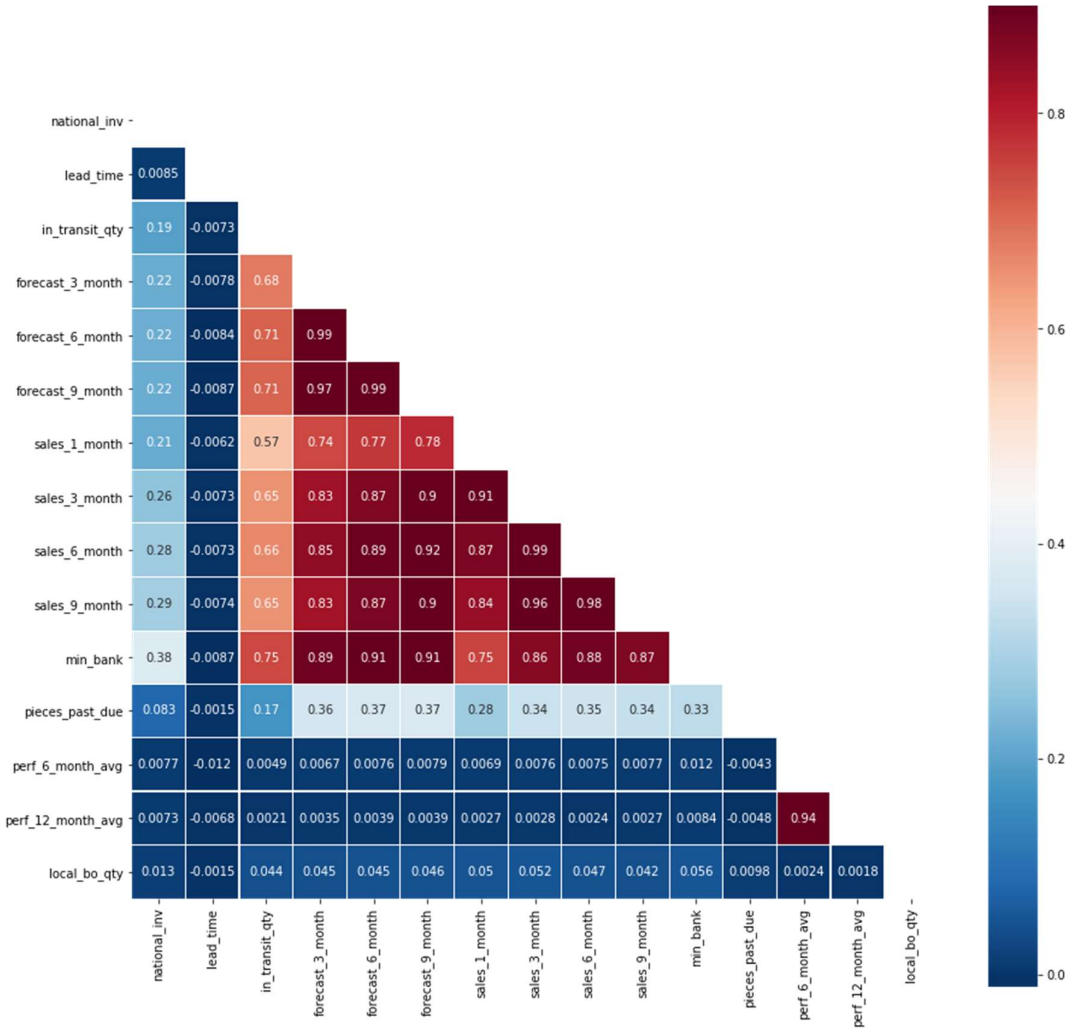
	Features	skewness
0	national_inv	4.920324
1	lead_time	0.016160
2	in_transit_qty	1.684185
3	forecast_3_month	0.249037
4	forecast_6_month	0.177607
5	forecast_9_month	0.147291
6	sales_1_month	0.328823
7	sales_3_month	0.170744
8	sales_6_month	0.118766
9	sales_9_month	0.095086
10	min_bank	0.592197
11	pieces_past_due	3.653505
12	perf_6_month_avg	-0.325724
13	perf_12_month_avg	-0.300686
14	local_bo_qty	3.005006
15	potential_issue_Yes	51.135444
16	deck_risk_Yes	2.225176
17	oe_constraint_Yes	118.883557
18	ppap_risk_Yes	3.136505
19	stop_auto_buy_Yes	-5.165571
20	rev_stop_Yes	84.054447

Skewness after treatment of outliers

Multi-collinearity Treatment:

Multi-collinearity is a situation when there is a significant dependency or association between the independent variables. This can lead to redundancy in the data hence need to be treated before final modelling.

Multi-collinearity can be depicted from the heat map of the correlation matrix shown in fig



Heat Map for numerical variables

- We can see that the Performance of source in past 6 months and 12 months are highly correlated.
- Similarly, there is high correlation between forecast sales of 3, 6 and 9 months respectively.
- High correlation is also observed between sales quantity for 1, 3, 6 and 9 months time period respectively.
- Minimum recommended stock is also correlated with all the forecast and sales features.
- In transit quantity and National inventory is also showing some level of positive correlation with forecast and sales features.

For the treatment of multi collinearity we devised two approaches using Principal Component Analysis (PCA): Principal component analysis, or PCA, is a statistical procedure that allows you to summarize the information content in large data tables by means of a smaller set of "summary indices" that can be more easily visualized and analysed. The underlying data can be measurements describing properties of production samples, chemical compounds or reactions, process time points of a continuous process, batches from a batch process, biological individuals or trials of a DOE-protocol, for example.

1. **PCA on 3 columns:**

We have 3 main groups with high multicollinearity:

Sales columns: sales_1_month, sales_3_month, sales_6_month, sales_9_month

Forecast columns: forecast_3_month, forecast_6_month, forecast_9_month

Performance columns: perf_6_month_avg, perf_12_month_avg

We have extracted each of these groups separately and applied PCA on them and finally concatenated the derived principal components we chose. By doing this, we have reduced the number of columns from 9 to 3 columns and multi-collinearity is mostly reduced.

All the Sales Columns were grouped into Sales_Index.

All the Forecast Columns were grouped into Forecast_Index.

All the Performance Columns were grouped into Performance_Index.

By following this approach, the biggest advantage is the retention of the feature explainability.

Using SelectKBest method to find the features which are significant after PCA on 3-columns.

	Features	scores	pvalues
0	national_inv	1158.419	0.000
1	lead_time	3004.049	0.000
2	in_transit_qty	1206.823	0.000
3	min_bank	41.961	0.000
4	pieces_past_due	5589.201	0.000
5	local_bo_qty	9841.985	0.000
6	potential_issue_Yes	50.110	0.000
7	deck_risk_Yes	5894.334	0.000
9	ppap_risk_Yes	2870.422	0.000
10	stop_auto_buy_Yes	589.072	0.000
11	rev_stop_Yes	20.005	0.000
12	Sales_index	7378.459	0.000
13	forecast_index	32451.855	0.000
14	performance_index	5522.478	0.000
8	oe_constraint_Yes	6.401	0.011

SelectKBest pvalues for all columns

From above table, we see that all features are important, as the pvalue is less than 0.05(significance level) for all the features. Hence not dropping any features from the PCA data.

2. PCA on entire data:

In this method, we have applied PCA on the complete data and were left with 11 components explaining 98% variance of the data.

PCA Approach-1 is preferred as it gives a better explainability which is the most important factor for business interpretation.

12.Base Model with treated data:

	Models	Train f1 score	Test f1 scores	Train ROC	Test ROC	Test recall	Test precision	kapppa_train_score	kapppa_test_score
0	Base model - Logistic regression	0.010152	0.000000	0.726344	0.664434	0.000000	0.000000	0.009693	-0.000542
1	Base model - RandomForest	0.986784	0.040000	0.999988	0.888479	0.021739	0.250000	0.986678	0.038809
2	SMOTE Data - Logistic Regression	0.811905	0.031192	0.873330	0.773835	0.666667	0.015969	0.581636	0.016307
3	SMOTE Data - RandomForest	0.392536	0.040000	0.983348	0.888479	0.021739	0.250000	0.244185	0.038809
4	PowerTransformed Data - Logistic Regression	0.842373	0.069396	0.913154	0.842870	0.724638	0.036443	0.686503	0.055463
5	PowerTransformed Data - RandomForest	0.999901	0.188235	0.999999	0.896752	0.173913	0.205128	0.999802	0.182423
6	Approach 1 PCA - Logistic regression	0.839317	0.063616	0.909818	0.839558	0.710145	0.033299	0.678183	0.049550
7	Approach 1 PCA - RandomForest	0.999901	0.208469	0.999998	0.889220	0.231884	0.189349	0.999802	0.201660
8	Approach 2 PCA - Linear Regression	0.836749	0.066234	0.908219	0.848437	0.739130	0.034670	0.673698	0.052207
9	Approach 2 PCA - RandomForest	0.999901	0.100592	0.999995	0.842819	0.246377	0.063197	0.999802	0.089363

Performance of base models with and without treatment

As seen above, Logistic Regression and Random Forest were applied on the treated data in different stages and model performance was best when Approach 1 PCA was applied. Hence going forward we will be running the other models using our Approach 1 PCA data.

Hence the final features considered for further modelling:

national_inv
lead_time
in_transit_qty
min_bank
pieces_past_due
local_bo_qty
potential_issue_Yes
deck_risk_Yes
oe_constraint_Yes
ppap_risk_Yes
stop_auto_buy_Yes
rev_stop_Yes
Sales_index
forecast_index
performance_index

Final Features selected

13.Final Model

As our target variable is a binary classification variable with two-classes, the models that we considered for modelling are Random Forest Classifier, Gradient Boost Classifier, Light GBM Classifier, XG Boost Model, Stochastic Gradient Descent(SGD) Classifier, Bagging Classifier with base model as Logistic Regression. We have treated and cleaned the data, so that many models can be applied and check the performance of each model.

We run the before mentioned models along with few tuned models on the PCA Approach-1 data and hence evaluated the performance of each model which can be seen below clearly.

	Models	Train f1 score	Test f1 scores	Train ROC	Test ROC	Test Recall
0	RandomForest Classifier	0.955154	0.071649	0.999998	0.890467	0.840580
1	RandomForest Tuned Classifier	0.893821	0.060561	0.972958	0.903678	0.884058
2	GradientBoost Classifier	0.910360	0.067750	0.986418	0.916791	0.891304
3	GradientBoost Tuned Classifier	0.927133	0.081072	0.990110	0.912334	0.833333
4	SGD Classifier	0.834294	0.064410	0.904487	0.835140	0.710145
5	SGD Tuned Classifier	0.856846	0.050671	0.919663	0.864251	0.847826
6	XGB Classifier	0.942222	0.082618	0.999688	0.881689	0.731884
7	XGB Tuned Classifier	0.904748	0.058321	0.999450	0.888551	0.840580
8	LGBM Classifier	0.899048	0.057534	0.998642	0.904077	0.884058
9	LGBM Tuned Classifier	0.909214	0.061780	0.998863	0.901734	0.855072

Performance of different models

Even though the models were built by optimising the F1 Score and ROC Score, the main aim of our project is to reduce False Negatives and to increase True positives, so that it will be able to predict the backorders correctly with a minimum beta-error. We have optimised this by using the best value of Youden's index.

Youden's Index:

Youden's index is used to find the best-threshold suitable for that model. By default, model will consider 0.5 as the threshold value which might not give a good performance for the model. Hence we found the best-threshold for every model and evaluate the model based on that.

Instead of selecting one best model, we selected 3 best models based on the highest F1-score and Recall-score out of all the above models and passed them into **Voting Classifier**.

The final models were selected on the basis of maximum number of true positives. The **top 3** models performed as follows:

1. Gradient Boost Base model - 123True positives out of 138 Positives with (Recall =0.91).

2. Hyper tuned Light GBM Model - 122 True positives out of 138 Positives (Recall=0.88).

3. Hyper tuned RandomForest Model – 122 True positives out of 138 Positives (Recall=0.88)

Gradient Boost Classifier gave a good TPR rate after hyper parameter tuning as well, but considering the fact that the base model already gave a better performance, we decided not to consider it as a best model.

XGBoost Classifier after hyper parameter tuning also gave a good TPR, but the f1 score was lowest of the lot, and we can observe high level of overfitting from the train and test roc scores. Hence not being considered for the voting classifier.

Cross-validation on selected models:

Further the chosen 3 models in terms of best performance were cross validated on the basis of accuracy score. Kfold cross validation with 10 splits wereused for the same and observed that the cross validation accuracy score is between the train and test accuracy scores for all the models.

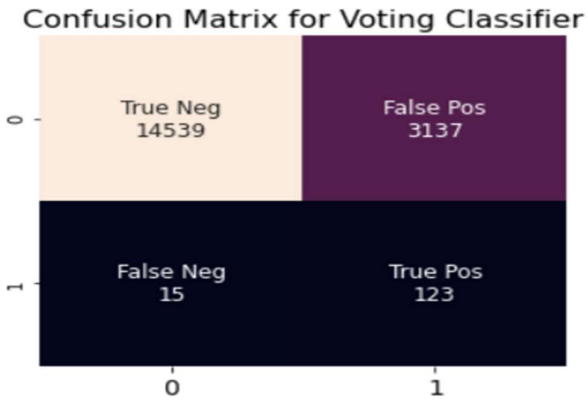
	Models	Train Accuracy	Test Accuracy	Cross Validated accuracy
0	RandomForest Tuned Classifier	0.910335	0.876109	0.908477
1	GradientBoosting Base Model	0.944263	0.927697	0.944934
2	LightGBM Base model	0.983100	0.970697	0.981027

Cross validation score on selected 3 best models

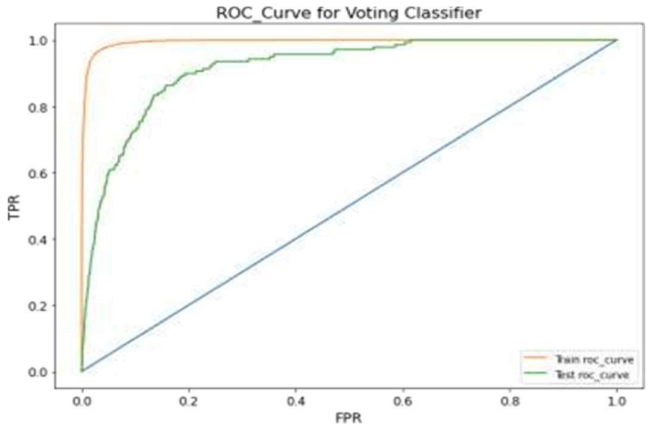
Voting Classifier:

A Voting Classifier is a machine learning model that trains on an ensemble of numerous models and predicts an output (class) based on their highest probability of chosen class as the output.

The selected 3-models were also passed into a voting classifier and by combining the power of the three models we managed to predict 123 True positives out of 138 Positives and F1_score of 0.07 which is so far the best.



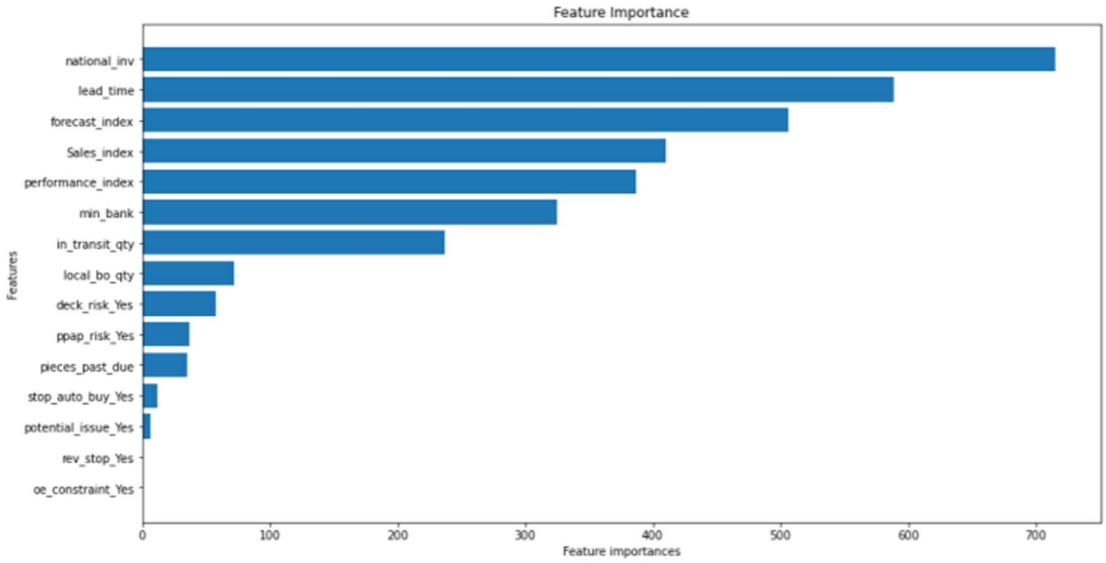
Confusion Matrix of Voting Classifier



ROC Curve for Voting Classifier

We can see that Test ROC Score = 91.7%

14. Feature Importance:



Feature Importances of final model

The top-5 important variables in predicting the backorder are:

1. forecast_index = Cummulative Forecast sales for the next 9 months
2. national_inv = Current inventory level of part.
3. Sales_index = Cummulative Sales quantity for the prior 9 month time period
4. lead_time = Transit time for product (if available)
5. in_transit_qty = Amount of product in transit from source

15.Conclusions:

From all the models that we have tried on our data, we got the apt results when we used the stacking technique on the 3 best performing models which are Gradient Boosting Base Model, Tunned Random Forest Model, Hyper parameter tuned Light GBM. The recall rate was 0.9 after using Voting Classifier on the above 3 models.

16.Business Interpretation:

As we can see that the most important features which is contributing almost **77%** are as follows:

1. **Forecast Index**: This is a resultant column of our PCA transformation of the original forecast columns. This feature has the highest importance of 31.16%. It says about the forecast of the sales in different intervals. Since it is very important to know the probable sales of every product well in advance so as to avoid any inventory shortage which in turn will lead to a back-order situation in case of a high demand for the product. The general idea is, the bigger the item value, the more delivery tolerance a business gets from customers. The high demand can be due to abnormal demand, flash sale, human errors and many other reasons. This can be avoided by staying up to date on current market trends and maintaining a minimum inventory levels for potential products that can go into high demand.
2. **National Inventory Level**: It has the second highest importance of 29.6% which means it has a huge impact on predicting back-orders. If the inventory level is in accordance with the demand of the product at that particular time there would be very little risk of that order going to back order. We can see from EDA, the parts that have low inventory levels tend to get into backorder frequently. A business selling a particular product without any stocks in inventory most likely to get into backorder when there is a surge in demand from customers.
3. **Sales Index**: This is a resultant column of our PCA transformation of the original Sales performance columns. It has the third highest importance of 11%. Whenever the sales performance for any product is high, the company maintains enough stock in inventory for that particular product to fulfil the surge in demand of that product. But when any particular products sales performance is poor, we can see from EDA that product getting into backorder when customer makes an order as the organisation did not anticipate any demand from customer. As seen in the heat map, there is a high level of collinearity in Forecast_Index and Sales_index.
4. **Lead Time**: It has a feature importance of 6.17%. It signifies how much time is required for a product to be delivered to the customer. As per its business logic lesser the lead time more will be the satisfaction level of the customer. If the lead time tends to increase the logistics will be hampered as there would be a pile up of delay in delivering

products. This kind of a scenario will definitely make the product go into back-order.

17.Implications:

As per our final inference, Inventory level, Sales Index, and Forecast index should be high and Lead time should be low so as to prevent a back-order scenario. After the deployment of our prediction model, the company would be able to predict most cases of back-orders well in advance. Through this, we are preventing any kind of financial loss that may occur due to the cancellation of back-order products. Also, it will prevent churning of customers and hence maintain a descent level of customer satisfaction.

18.Limitations:

Since, back-orders arise due to unprecedented situations which is generally hard to predict with full confidence, our model can show some anomaly in predicting a true case of back-order. This was quit evident by the large imbalance present in the data.

19.Reference and Bibliography:

Lokesh Malviya (2021), Backorder prediction in the supply chain using machine learning

James Duane Abbey (2008), Analysing impacts on backorders and ending inventory in MRP due to changes in lead-time, demand variability and safety stock levels

https://journalofbigdata.springeropen.com/articles/10.1186/s40537-020-00345-2

https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9046037

https://projekter.aau.dk/projekter/files/262657498/master_thesis.pdf

https://www.tandfonline.com/doi/abs/10.1080/07408170590961139

https://www.inderscienceonline.com/doi/pdf/10.1504/IJBDA.2020.108700

https://www.sciencedirect.com/science/article/pii/S2214785320392270#