EMORY GOIZUETA BUSINESS SCHOOL | GOIZUETA BEYOND

Predicting Product Backorder Using Machine Learning

Machine Learning II Final Project

Team 404
Frank Fan, Carl Xi, Yaping Zhang, Jie Zhu

404

Hello!

We are Team 4(04)

Comprised of:

- Frank Fan
- Carl Xi
- Yaping Zhang
- Jie Zhu

404

Presentation Overview

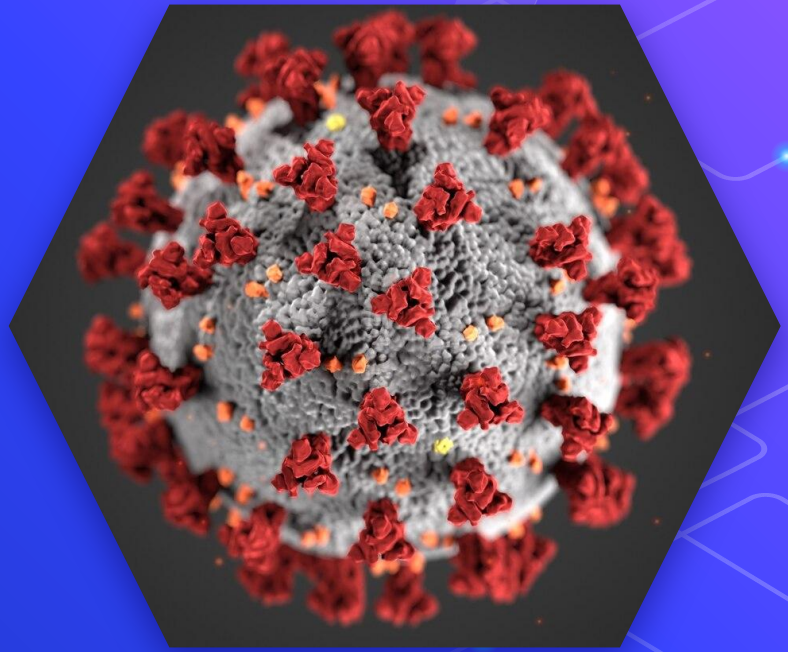
Carl	Yaping	Frank	Jie
Business Understanding Data Understanding	Data Preparation: • Data Cleaning • Dealing with Imbalance • Feature Engineering • Data Standardization	Modeling	Evaluation Deployment

1. Business Understanding

Motivation Behind Our Analysis

Project Inspiration

Recent COVID-19 outbreak has caused severe backorder problems for countless industries, from Nintendo Switches to masks.



In a nutshell:

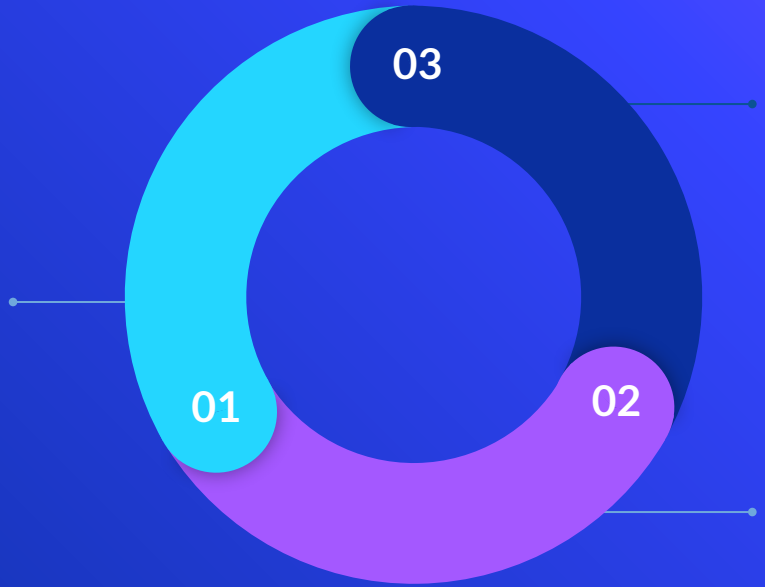
Can we build a cost-aware model that accurately predicts when backorders are most likely to happen?

> While Just-In-Time production methods and demand forecasting has significantly improved inventory overstocking issues, understocking and the backorder problems associated with it continues to erode business bottom line across industries.

Backordering is a Lose³ Scenario

Businesses Lose

Businesses lose potential profit and has to put in backorder in hopes of future revenue, but consumers have the right to churn and chose another retailer



Suppliers Lose

Suppliers are strained to play “catch-up” in order to fulfill the above-expected demand. Production lines are pushed into overtime, inducing cost along the entire production and supply line

Consumers Lose

Consumers do not have their needs fulfilled at the time it arises (e.g. a birthday present), forcing them to find alternatives or even bear the consequences (e.g. shortage of face masks.)

2. Data Understanding

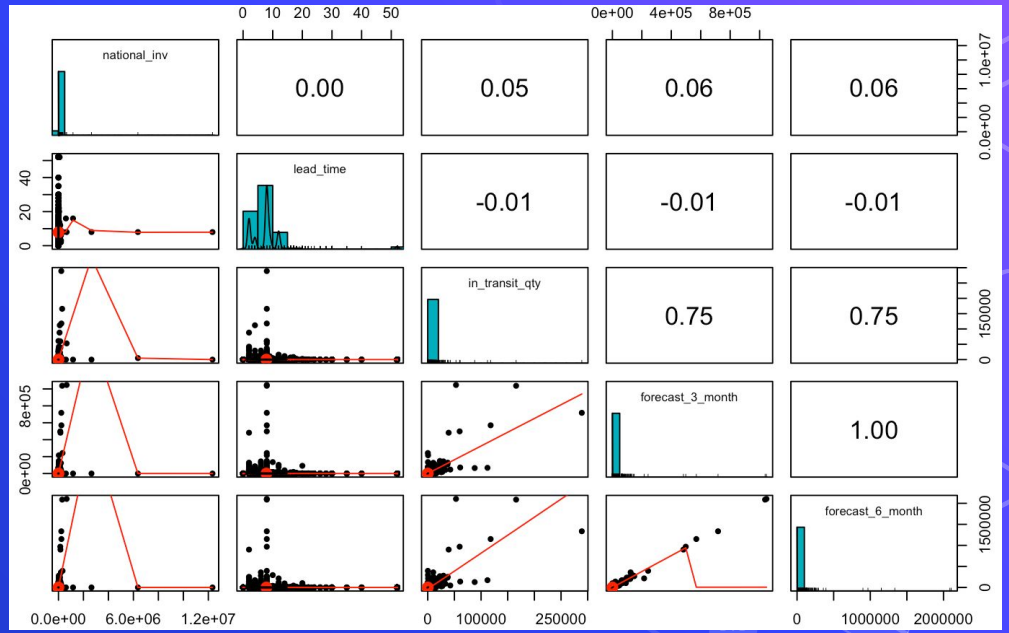
Describing the Data

Data Understanding

- Source: datahub
- Title: Global Garment Supply Chain Data
  - Walmart Data Subset
- ~1.6 million entries * 22 attributes
- 6 binary, 15 numeric and 1 notation variables
- Significantly Imbalanced Data
- Summary Statistics Included in Appendix

Data Understanding

- We also looked at a pairplot of our variables (to the right is an example for legibility, not our final graph) to look at the distribution of our variables and for any potential multicollinearity
- Insights:
  - Some variables heavily correlated (forecast 3&6 months
  - Many Variables heavily skewed (lead_time)
- Result: Data Transformation is necessary (e.g. normalization & standardization). Details covered in Data Preparation



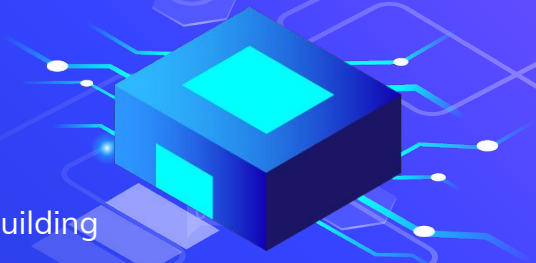
3.
Data Preparation

Preparing The Data For Modeling

1. Data Cleaning
2. Dealing with Imbalance
3. Feature Engineering
4. Data Standardization

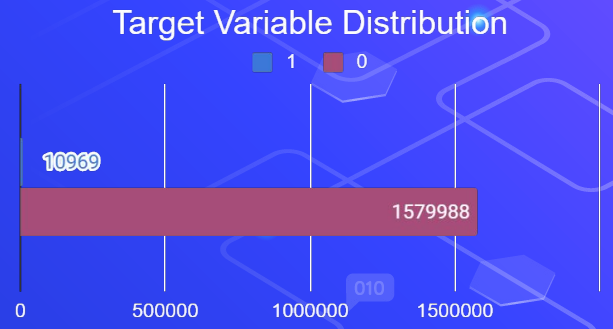
Data Cleaning

- 23 attributes: 22 features and 1 target variable
  - The features are a representative of different product suppliers' historical logistic performance, and the target variable shows whether or not a product supplier went on backorder.
- Data Issues:
  - Some of the values are not valid with a -99 load or marked with NA.
  - One notation column
- Data Cleaning Process
  - For the obvious missing data, we will delete the rows.
  - For the notation column, since it will not be useful in the model building part, we will also delete it.



Dealing With Imbalance

- Data issue:
  - Target variable is imbalanced : 1579988 ‘0’ values and 10969 ‘1’ values, meaning we have about 1% of the products that went on backorder
- Imbalance data consequence:
  - Further predictive models can get high overall accuracy but with a very low recall for the minority class
  - Models can’t generate any good insights
  - Problem Solving Methods:
    - Synthetic Minority Oversampling Technique (SMOTE)
    - Down-sampling



Dealing With Imbalance

- Synthetic Minority Oversampling Technique (SMOTE)
  - A new up-sampling approach
  - It solves the imbalance issue by resampling from the minority class while slightly perturbing feature values, thereby creating "new" samples

> “SMOTE first selects a minority class instance at random and finds its k nearest minority class neighbors. The synthetic instance is then created by choosing one of the k nearest neighbors b at random and connecting a and b to form a line segment in the feature space. The synthetic instances are generated as a convex combination of the two chosen instances a and b.”
>
> ----‘Imbalanced Learning: Foundations, Algorithms, and Applications, 2013’

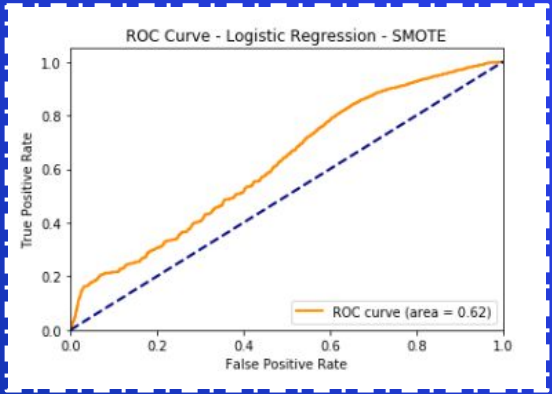
Dealing With Imbalance

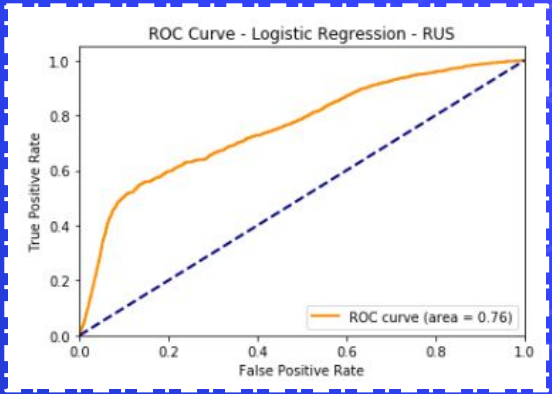
- **Down-sampling**
  - Down-sampling is the process of randomly removing observations from the majority class to prevent its signal from dominating the learning algorithm
  - The specific steps need to take are:
    1. First, we'll separate observations from each class into different DataFrames
    2. Next, we'll resample the majority class without replacement, setting the number of samples to match that of the minority class
    3. Finally, we'll combine the down-sampled majority class DataFrame with the original minority class DataFrame

Dealing With Imbalance

- **Compare Performance:**

  - After trying out both resampling approaches using **logistic regression** base model, the results show that the under-sampling technique works the best for our data (as shown in the above ROC graphs)





  - Thus, we decided to use undersampling to transform our dataset

Feature Engineering

- Methods:
  - Mathematical Based: We will raise the power of all continuous variables (15) and create interactive terms for all binary features (6)
  - Business Understanding Based: Based on domain knowledge, we created two features named 'forecast_mean' and 'sales_mean', indicating monthly mean value of the Forecast for the next 3,6 and 9 months and Sales quantity for the prior 1,3,6,9 months
- Results:
  - The dataset has 1586967 rows and 37 columns after adding interaction terms and 1586967 rows and 52 columns after adding powered features
  - The final date set has 1586967 rows and 54 columns

Data Standardization

- Since different models have different requirements for data, we will selectively perform data standardization
  - Logistic Regression (NO Data Standardization)
  - SVM (With Data Standardization)
  - Neural Network (With Data Standardization)
  - Gradient Boosting (NO Data Standardization)

4.
Modeling

Experimenting with Various Modeling Methods

Modelling Process

- Algorithms:
  - Logistic Regression;
  - SVM
  - Neural Network
  - Gradient Boosting
- Stratified 5-fold Cross Validation
- Evaluation Metric : F1
- Feature transformation
  - Standardized features
  - Unstandardized features
- Feature selection
  - Correlation
  - Automated Stepwise Forward Selection
- Hyperparameter tuning with nested cross validation

A tradeoff between accuracy and comprehensibility

Feature Selection

- First step: Select Features based on Correlation
  - Remove highly-correlated features
  - Filter features with absolute correlation larger than threshold

- Second step: Stepwise Forward Selection
  - Iterative approach
  - Adds features based on F1 score
  - Find the optimal subsets of features for each algorithm

Algorithm	Selected Features	F1-Score after feature selection
Support Vector Machine	'local_bo_qty', 'perf_12_month_avg', 'local_bo_qty^2', 'forecast_mean', 'national_inv'	0.72
Neural Network	'potential_issue=Yes', 'oe_constraint=Yes, stop_auto_buy=Yes', 'potential_issue=Yes, 'deck_risk=Yes', 'local_bo_qty^2', 'forecast_9_month', 'national_inv', 'sales_1_month', 'sales_3_month', 'sales_6_month', 'sales_9_month', 'in_transit_qty'	0.85
Gradient Boosting	'deck_risk=Yes, ppap_risk=Yes', 'perf_12_month_avg', 'forecast_mean', 'forecast_9_month', 'national_inv', 'sales_1_month', 'sales_3_month', 'stop_auto_buy=Yes', 'in_transit_qty', 'perf_12_month_avg^2', 'lead_time^2'	0.88

Table: Validation Results after Feature Selection

Hyperparameter Tuning

- Grid Search
  - All the possible combinations of parameter values in a grid are evaluated and the best combination is found based on F1.
- Nested Cross Validation
  - Inner Folds for Choosing the parameters
  - Outer Folds for Evaluating the generalization performance

Algorithm	Parameters	Meaning	Value
Gradient Boosting	n_estimator	The number of boosting stages to perform.	200
	max_depth	The maximum depth of the individual regression estimators.	12
	min_sample_leaf	The minimum number of samples required to be at a leaf node.	9
Support Vector Machine	kernel	The kernel type to be used in the algorithm. For example, rbf refers to a radial basis function kernel.	rbf
	C	Regularization parameter. The strength of the regularization is inversely proportional to C.	1
Neural Network	activation	Activation function for the hidden layer. For example, relu refers to the rectified linear unit function.	relu
	alpha	L2 penalty parameter.	0.0001

Table: Parameters Adopted for Classifiers

Summary of Performances

Models	Raw Model F1-Score	Correlated Features F1-Score	Automated Selected Features F1-Score	F1-Score after Hyperparameter Tuning
Logistic Regression (Base Model)	0.643			
Support Vector Machine	0.683	0.702	0.715	0.731
Neural Network	0.801	0.821	0.847	0.868
Gradient Boosting	0.834	0.857	0.879	0.903

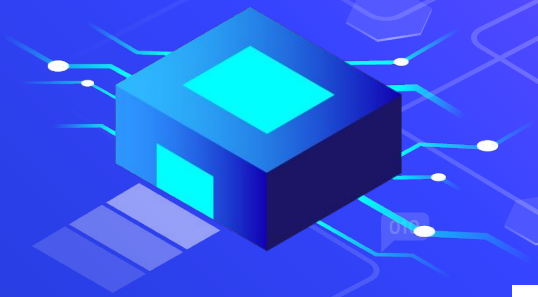
We have made improvements along the way.
The final model is a gradient boosting model with max_depth=12, max_sample_leaf=9, using 12 features.

5. Evaluation

Estimating the Dollar Amount Impact

Cost Analysis

- False Positives
  - Cause overstock
  - Generate extra warehousing costs
- False Negatives
  - Customer loss
    - Customers switch to other sources
    - Cause revenue loss in future
  - Potential profit loss
    - Supply cannot cover consumer demands
    - Lose profit from direct sales loss



Facts Used in Calculation

- Walmart has total revenues of $514,405M and cost of sales of $385,301M in 2019, having a gross margin of (514,405 – 385,301) / 514,405 = 25.10%.

- According to a report by McKinsey & Company, warehousing cost of retail industry is about 2.5% of total sales.

- According to a research by Statista, total sales from clothing department for Walmart is expected to be $22.1B in 2019 and $22.4B in 2020.

Assumptions Used in Calculation

- Walmart's clothing department has similar gross margin as the rest of the corporation.
- Walmart's clothing department has similar warehousing cost rate as the retail industry average.
- On average, each backorder causes 0.1% of the customer base to switch to other stores.
- The effect of customer base loss lasts for 2 years on average.
- The increase of warehousing cost due to overstock is 20% on average.
- There are 1586967 orders (total observations in the dataset) each year.

Cost Calculation

- True Negatives & True Positives: Cost = 0
- False Negatives:
  - Customer Loss Cost = (Clothing Sales 2019 + Clothing Sales 2020) * Sales Loss Rate / Total Orders = (22.1B + 22.4B) * 0.1% / 1586967 = $28.04
  - Potential Sales Loss Cost = Gross Margin * Clothing Sales 2019 / Total Orders = 25.10% * 22.1B / 1586967 = $3495.09
  - Total Cost = Customer Loss Cost + Potential Sales Loss Cost = $28.04 + $3495.09 = $3523.13
- False Positives: Cost = Warehousing Cost Rate * Clothing Sales 2019 * Increase Rate of Cost / Total Orders = 2.5% * 22.1B * 20% / 1586967 = $69.63

Expected Financial Impact

	Predicted N	Predicted Y
Actual N	0	69.62968
Actual Y	3523.135	0

Cost Matrix

	Predicted N	Predicted Y
Actual N	0	-69.62967
Actual Y	0	3523.1351

Profit Matrix

	Predicted N	Predicted Y
Actual N	77.99%	21.32%
Actual Y	0.06%	0.63%

Confusion Matrix

Expected dollar amount impact

= ∑(Condition Percentage * Total Orders * Profit Amount)

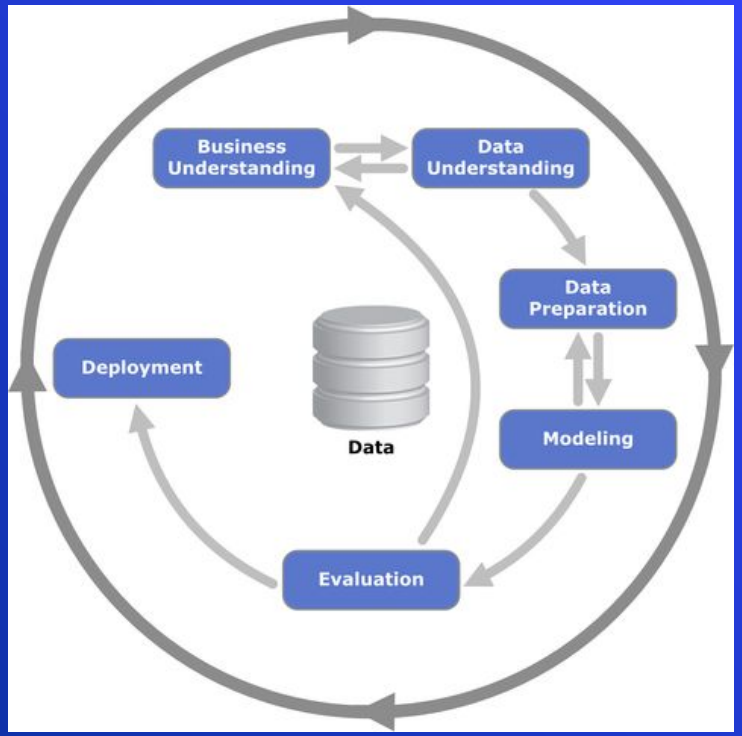
= 21.32% * 1586967 * -$69.63 + 0.63% * 1586967 * $3,523.14

= $11,494,262

6. Deployment

Apply and Improve

Overall Deployment Plan



- We recommend deploying our model with the Cross-industry standard process for data mining (**CRISP-DM**) framework as guidance, keeping the process iterative (e.g. trial run with limited product SKUs, re-evaluate business understanding and re-run entire CRISP-DM process)
- Costs of overstocking and understocking also differs business to business, **adjustments** must be made and the **cost** function must be re-evaluated
- Preventing backorder should not come at the cost of **ethics** reasons (e.g. overstocking on hand sanitizer in wake of Coronavirus Outbreak)
- Firms implementing our model should be aware of difference in industry and the **risks** associated with them (e.g. risks associated with the fashion industry are different from the ones for the consumer tech industry)

Risk Mitigation

- Apply this model to a portion of Walmart's clothing department
- Collect data for the whole clothing department to see the actual effect of application
- Include new data and re-run the model to improve the model performance
- Adjust other factors like customer loss rate to improve calculation accuracy of profit matrix
- Apply the new model to the whole clothing department
- Repeat step 2 to 4 and apply to more departments
- Repeat step 2 to 4 and apply to the whole market space

Thanks!

Q&A

Please Email your questions to
jie.zhu@emory.edu

Appendix I:

Feature Introduction:

- X1 = Current inventory level of a component: 'national_inv';
- X2 = Registered transit time for the product: 'lead_time';
- X3 = In transit quantity: 'in_transit_qty';
- X4,5,6 = Forecast for next 3,6 and 9 months: 'forecast_3_month', 'forecast_6_month', 'forecast_9_month';
- X7,8,9,10 = Sales quantity for the prior 1,3,6,9 months: 'sales_1_month', 'sales_3_month', 'sales_6_month', 'sales_9_month';
- X11 = Minimum recommended amount to stock: 'min_bank';
- X12 = Parts overdue from source: 'pieces_past_due';
- X13,14 = Source performance in last 6 and 12 months: 'perf_6_month_avg', 'perf_12_month_avg';
- X15 = Amount of stock orders overdue: 'local_bo_qty';
- X16-21 = General risk flags: 'potential_issue=Yes', 'deck_risk=Yes', 'oe_constraint=Yes', 'ppap_risk=Yes', 'stop_auto_buy=Yes', 'rev_stop=Yes';
- Y= Product went on backorder: 'went_on_backorder=Yes'.

Appendix II:

Descriptive Statistics:

	count	mean	std	min	25%	50%	75%	max
national_inv	1586967.0	489.509814	30461.681455	-27256.0	4.00	14.00	78.00	12334404.0
lead_time	1586967.0	7.872267	7.056024	0.0	4.00	8.00	9.00	52.0
in_transit_qty	1586967.0	45.474925	1309.357238	0.0	0.00	0.00	0.00	489408.0
forecast_3_month	1586967.0	188.743874	5182.992191	0.0	0.00	0.00	5.00	1427612.0
forecast_6_month	1586967.0	365.339017	10099.621249	0.0	0.00	0.00	15.00	2461360.0
forecast_9_month	1586967.0	536.280119	14825.764059	0.0	0.00	0.00	25.00	3777304.0
sales_1_month	1586967.0	56.911400	1854.774698	0.0	0.00	0.00	5.00	741774.0
sales_3_month	1586967.0	178.483536	4971.128633	0.0	0.00	1.00	16.00	1105478.0
sales_6_month	1586967.0	352.231864	9679.297427	0.0	0.00	3.00	33.00	2146625.0
sales_9_month	1586967.0	544.127176	15148.714501	0.0	0.00	4.00	50.00	3205172.0
min_bank	1586967.0	53.203800	1119.033891	0.0	0.00	0.00	3.00	205786.0
potential_issue=Yes	1586967.0	0.000568	0.023834	0.0	0.00	0.00	0.00	1.0
pieces_past_due	1586967.0	2.172666	243.402053	0.0	0.00	0.00	0.00	146496.0
perf_6_month_avg	1586967.0	-1.014934	13.272727	-99.0	0.69	0.84	0.97	1.0
perf_12_month_avg	1586967.0	-0.553222	11.445797	-99.0	0.69	0.82	0.96	1.0
local_bo_qty	1586967.0	0.633321	33.439327	0.0	0.00	0.00	0.00	12530.0
deck_risk=Yes	1586967.0	0.203254	0.402420	0.0	0.00	0.00	0.00	1.0
oe_constraint=Yes	1586967.0	0.000154	0.012424	0.0	0.00	0.00	0.00	1.0
ppap_risk=Yes	1586967.0	0.118809	0.323564	0.0	0.00	0.00	0.00	1.0
stop_auto_buy=Yes	1586967.0	0.975367	0.155003	0.0	1.00	1.00	1.00	1.0
rev_stop=Yes	1586967.0	0.000258	0.016071	0.0	0.00	0.00	0.00	1.0
went_on_backorder=Yes	1586967.0	0.006912	0.082850	0.0	0.00	0.00	0.00	1.0