**Predicting Backorders using Machine Learning**

*Lu Yang*

1. ***Abstract***

Backorder is a common supply chain problem which can cause many bad effects. Multiple suppliers and limited time that customer are willing to wait make this problem even more difficult to solve. In this paper, two machine learning models are used to build a predictive model for monitoring backorders. They are decision tree and random forest. Sampling and dummy variables are employed in these two models.

1. ***Introduction***

1. Question: Can you predict product backorders?

In other words, can you predict backorder risk for products based on historical data?

2. Data Source:

* Can you predict product backorders?

https://www.kaggle.com/tiredgeek/predict-bo-trial

This dataset includes training data and test data.

Training dataset contains the descriptive data for the 8 weeks prior to the week we are trying to predict. These data was recorded weekly.

* Columns:

1. Predictor variables:

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

1. Target variable:

went\_on\_backorder - Product actually went on backorder

3. Backgrounds:

1) What is a backorder?

Backorders are products that are temporarily out of stock, but will ship to customers when it is available.

In other word, a backorder generally indicates that customer demand for a product or service exceeds a company's capacity to supply it.

1. Backorder is a big problem.

* Huge number of sales orders and different suppliers where the out of stock items from increase complexity to the workload.
* Customers may not have the patience to wait for items. It may lead to lost sales and low customer satisfaction.

1. Way to handle it:

In order to monitor backorder, we can use machine learning to identify products at risk of backorders. Sales person can prepare for these products in advance.

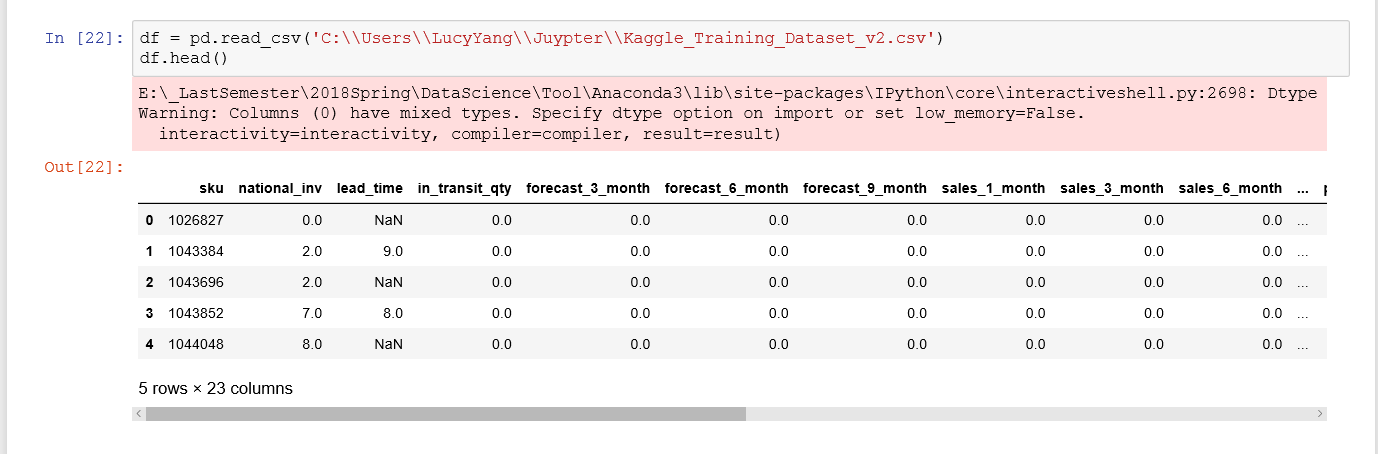
Machine learning algorithm: Naive Bayes, Support vector machine, Decision tree, Random forest, K nearest neighbors

1. ***Code with Documentation***

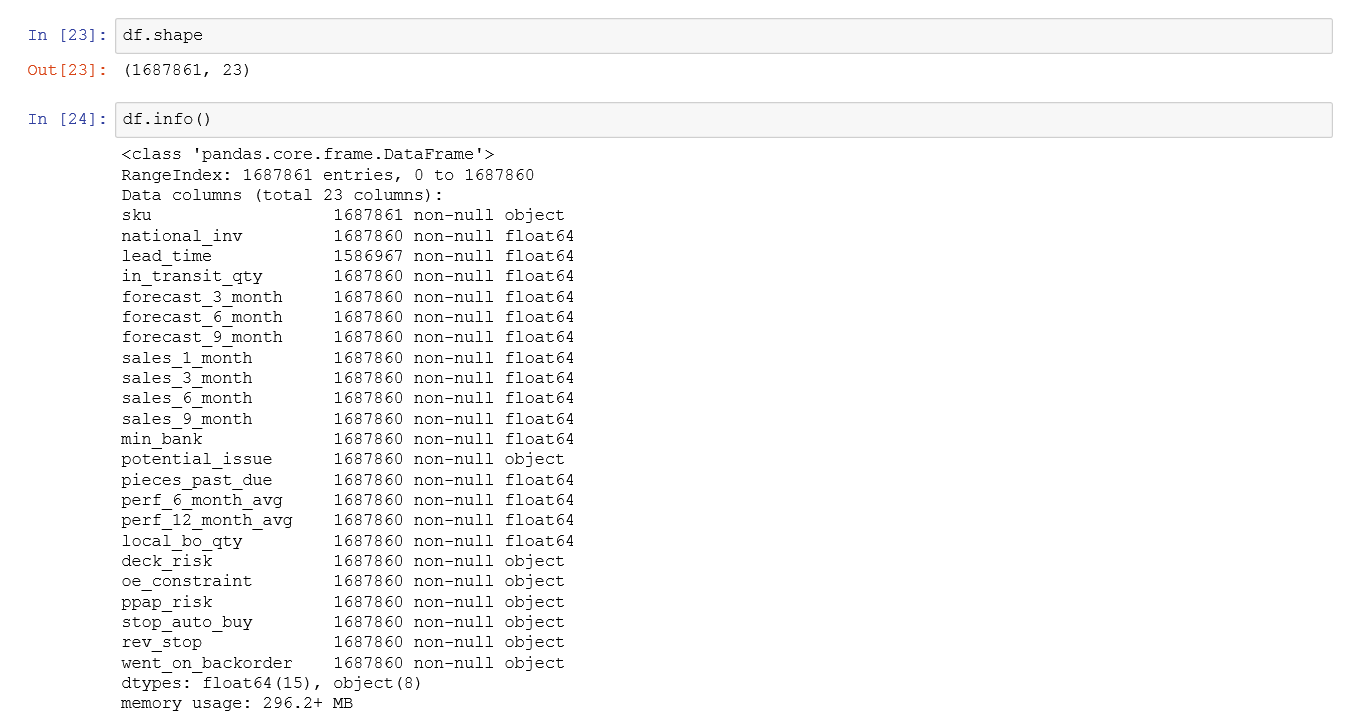
EDA:

1. Load and View my data

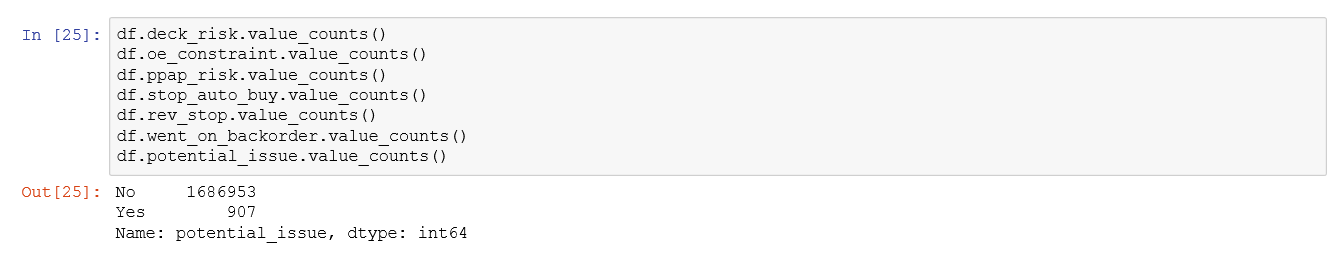
Load my data:



View basic info:



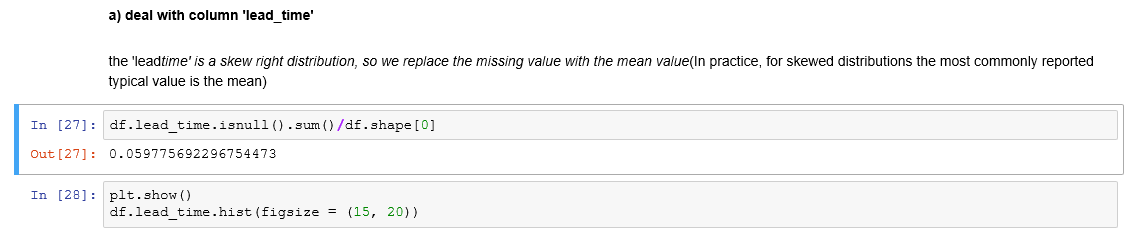
View categorical variables:

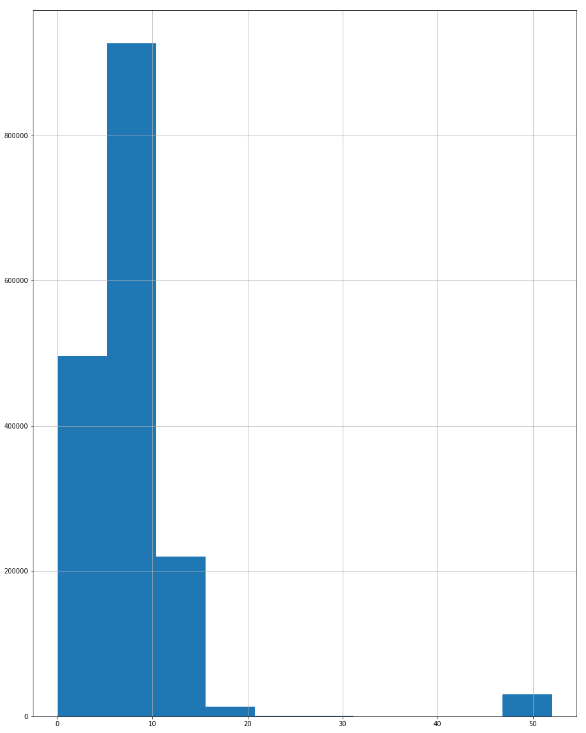


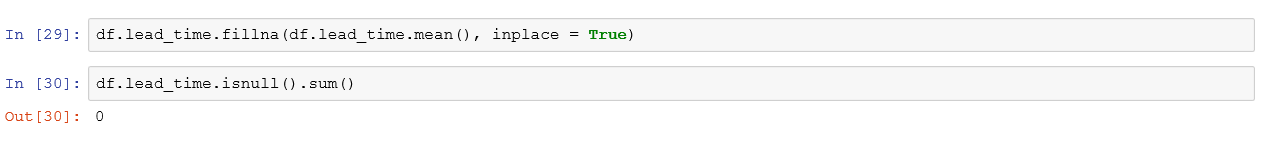
1. Data cleaning

Check missing value:

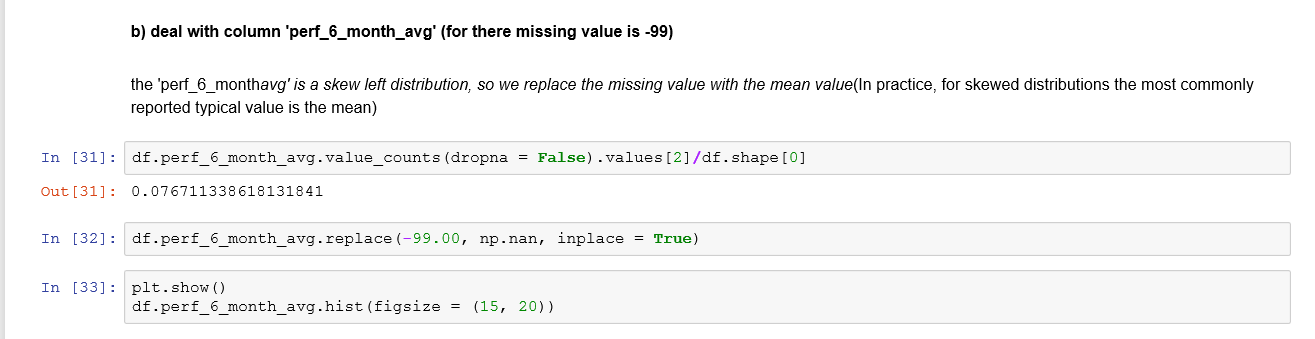
1. Deal with column ‘lead\_time’

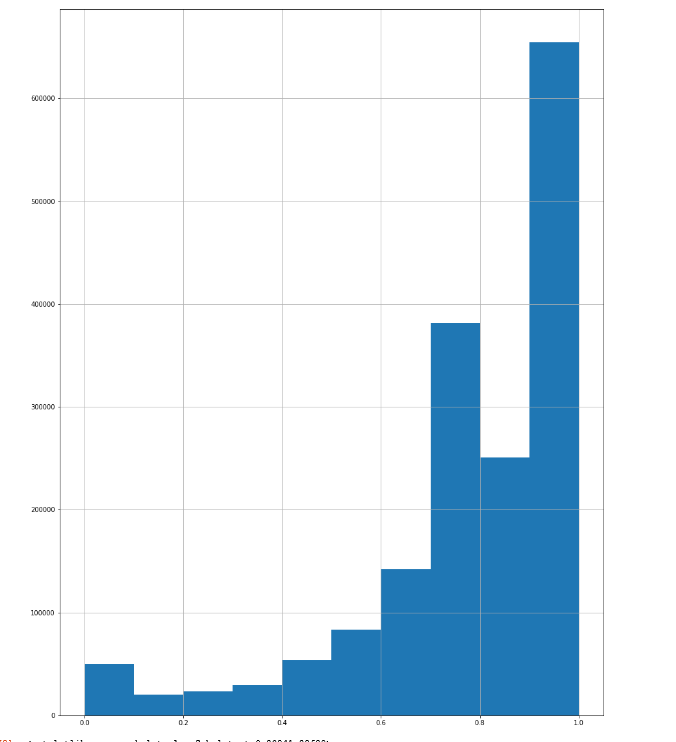


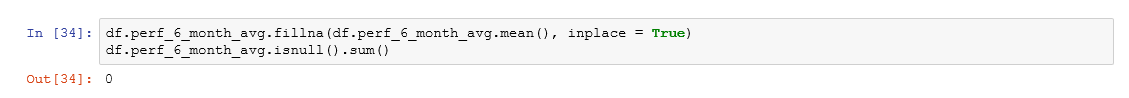




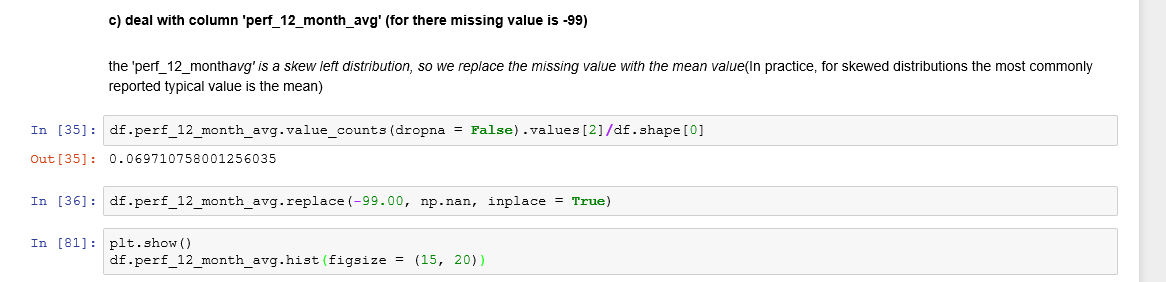
1. deal with column 'perf\_6\_month\_avg' (for there missing value is -99)

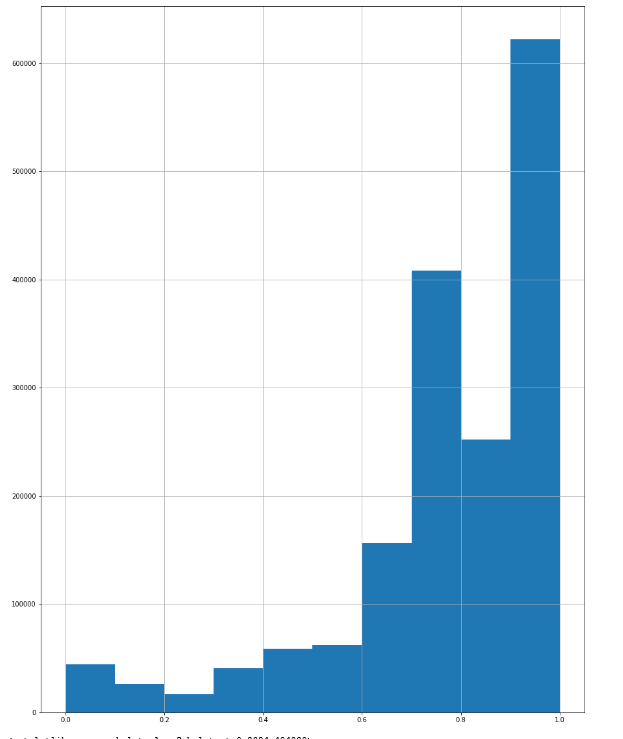


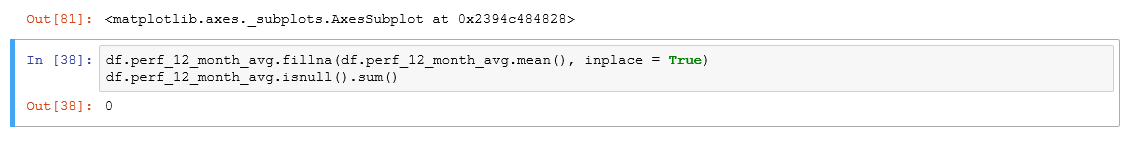




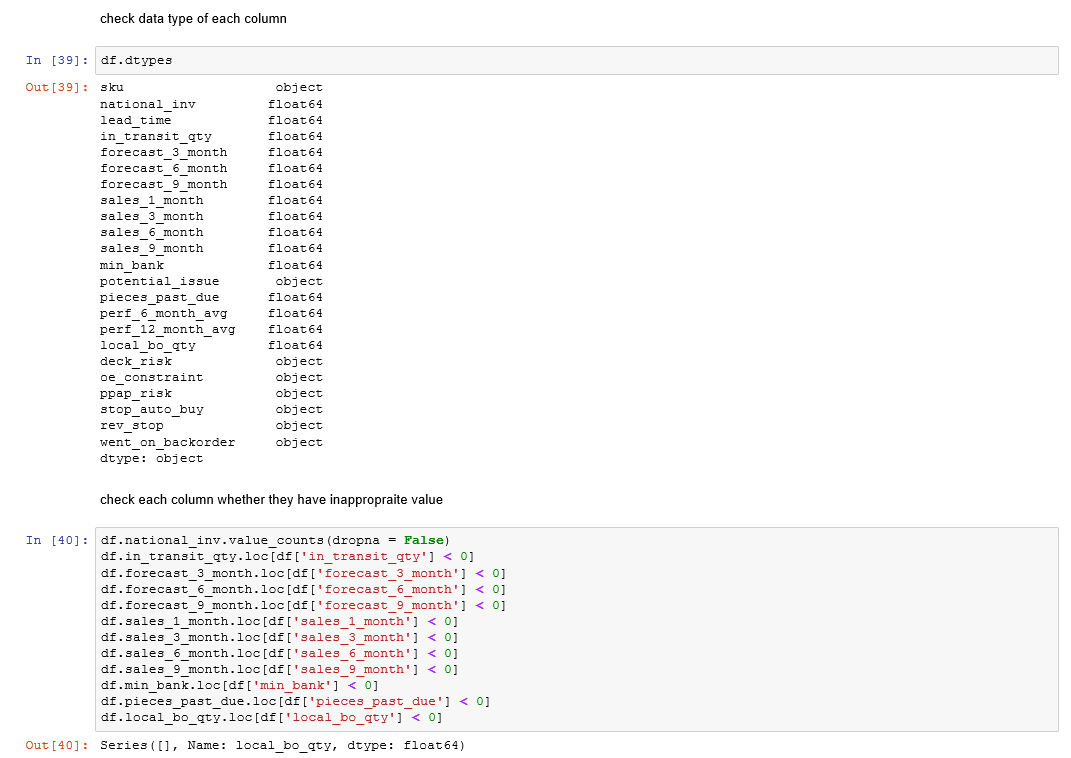
1. deal with column 'perf\_12\_month\_avg' (for there missing value is -99)





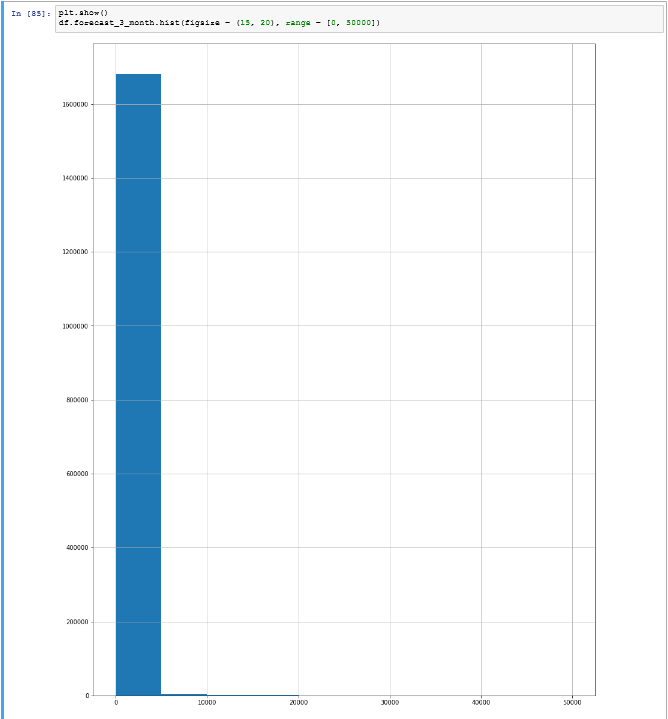


Check inappropriate value

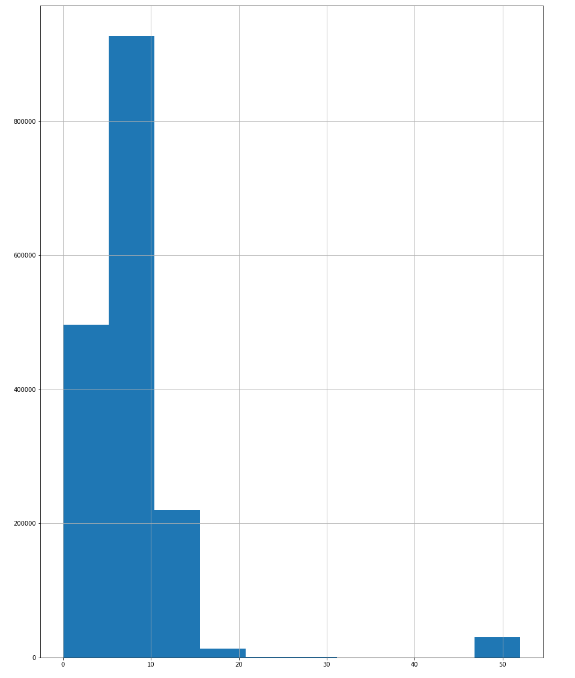


1. Data distribution
2. Each column distribution

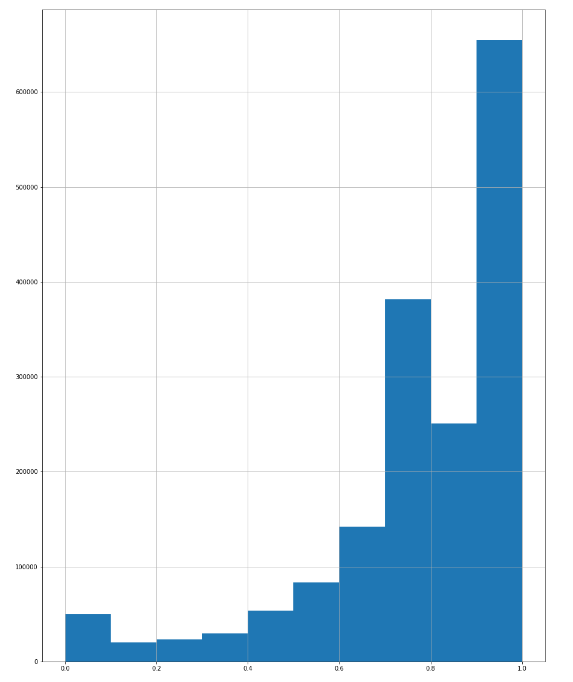
**forecast\_3\_month is a skew right distribution. Most values are gathered in [0, 20000]** (the mean is greater than the median.)



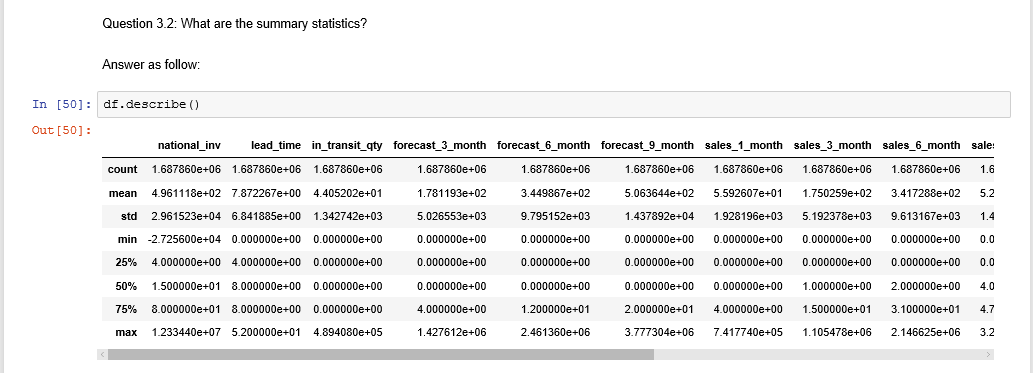
**lead\_time is a skew right distribution. Most values are gathered in [0, 60]** (the mean is greater than the median.)



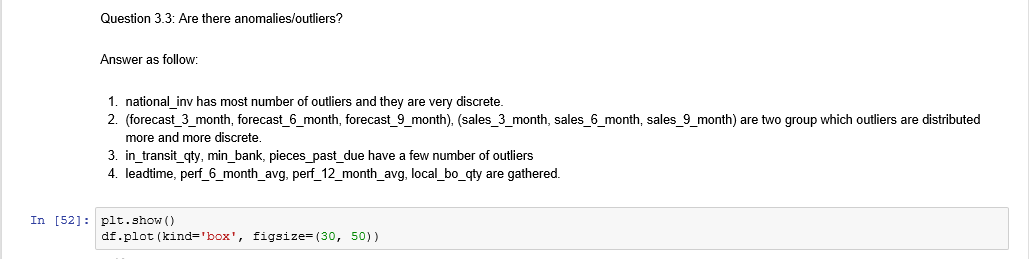
**perf\_6\_month\_avg is a skew left distribution. The range is [0, 1]** (the mean is less than the median.)

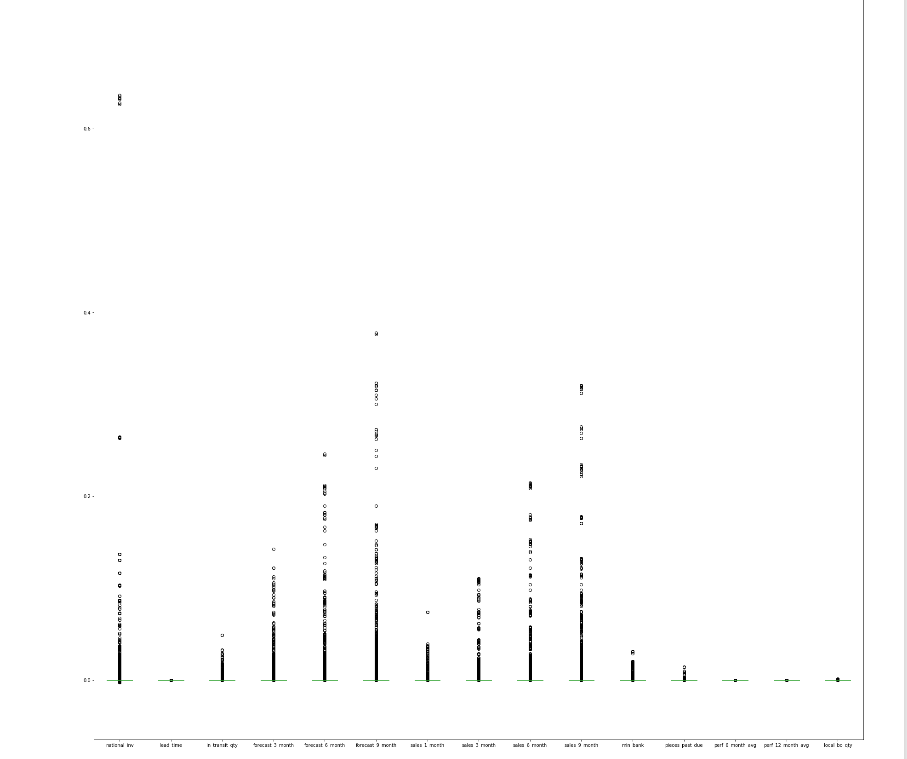


1. Summary statistics

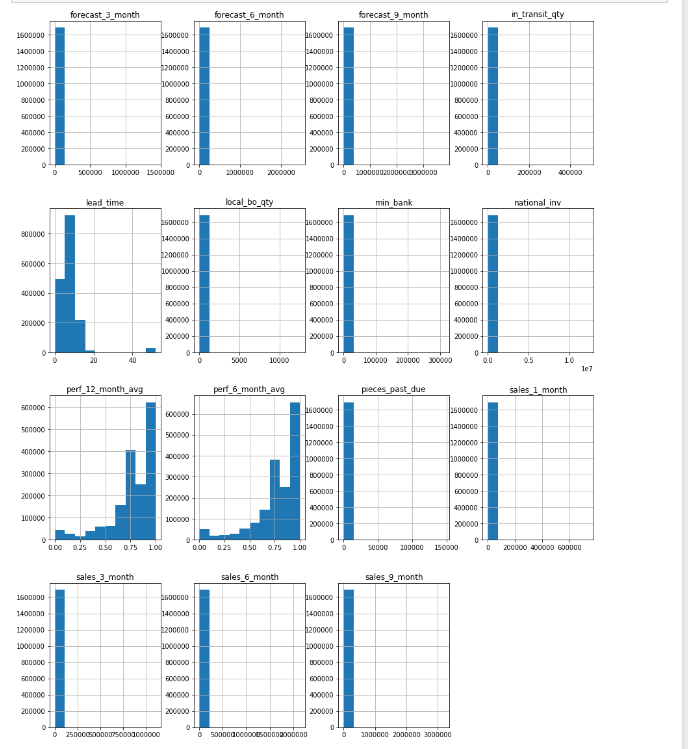


1. Outliers





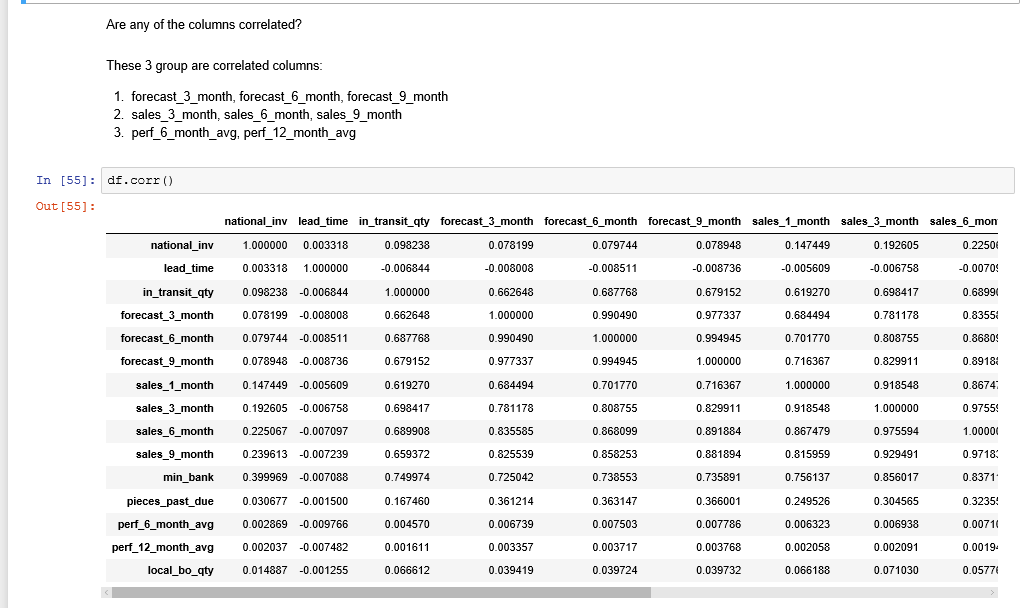
1. Plot each column



Summary

* Most columns have a very clustered and representative range.
* Lead time, perf\_6\_month\_avg, perf\_12\_month\_avg cannot find a typical value to represent these columns

1. Columns Correlating

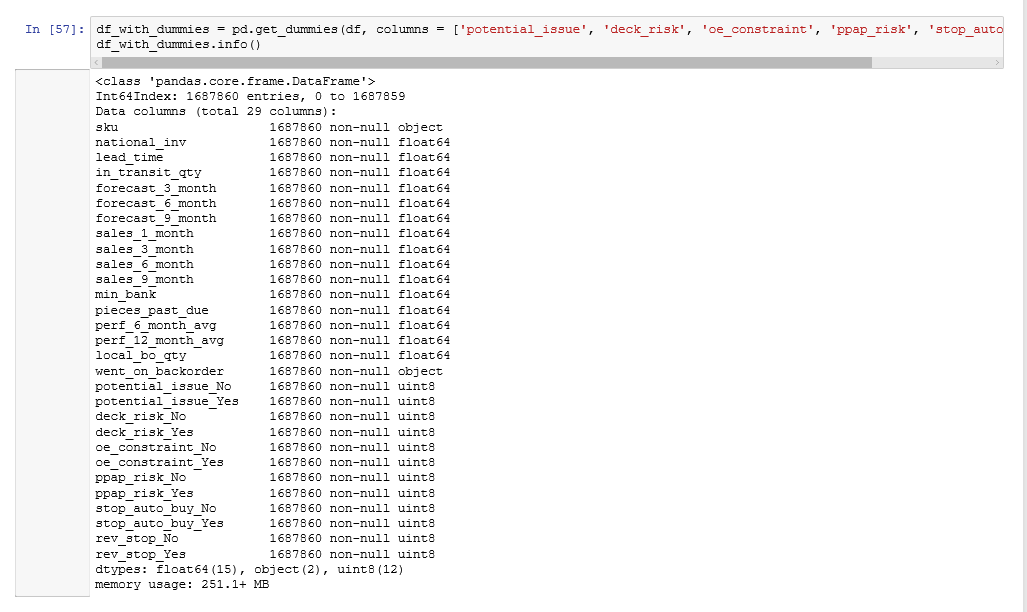




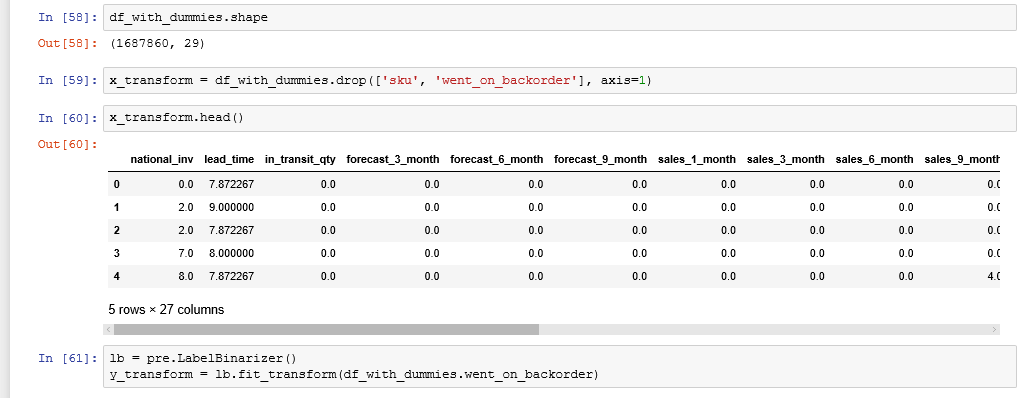
Analyze data:

1. Data transform

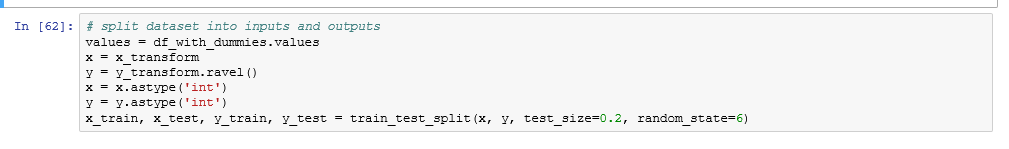
Converting categorical data into number:



Transform data for model:



Sampling into train data and test data



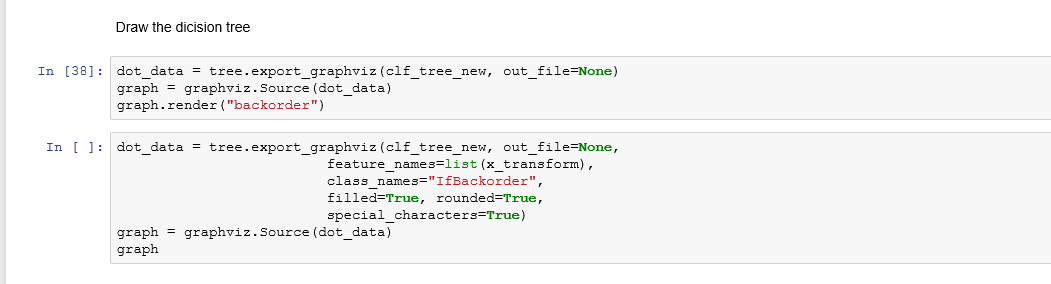
1. Default decision tree



1. Decision tree(Max Depth = 3)



Draw the decision tree:



1. Random forest



1. ***Results***

* Summary of EDA:

1. Overall:

1) This dataset is a very big dataset which have 1687861 rows and 23 columns. It means that I need build a large model for this dataset.

2) This dataset has many categorical variables which I need to convert it to numerical variables.

3) The target variable is categorical. So I should choose classification mode to analyse it.

2. Model recommendation:

1) decision tree

2) random forest

3. Data cleaning:

1) This dataset has some missing value: lead\_time has NaN; perf\_6\_month\_avg and perf\_6\_month\_avg have -99 as missing value. I use mean to replace the missing value because they are skewed distribution which use mean as typical value.

2) This dataset doesn't have inapproprate values

4. Data distribution:

Most columns are skewd distribution but have a very clusterd and representative range.

1) national\_inv has te most number of outliers and are most discrete.

2) (forecast\_3\_month, forecast\_6\_month, forecast\_9\_month), (sales\_3\_month, sales\_6\_month, sales\_9\_month) are two group which are more and more discrete.

3) in\_transit\_qty, min\_bank, pieces\_past\_due have a few number of outliers and not very discrete.

4) leadtime, perf\_6\_month\_avg, perf\_12\_month\_avg, local\_bo\_qty are gathered.

5) Lead time, perf\_6\_month\_avg, perf\_12\_month\_avg cannot find a typical value to represent these columns

5. Column correlated:

(forecast\_3\_month, forecast\_6\_month, forecast\_9\_month), (sales\_3\_month, sales\_6\_month sales\_9\_month), (perf\_6\_month\_avg, perf\_12\_month\_avg) are column correlated.

* Model results:

1. Accuracy\_Score

|  |  |
| --- | --- |
| Model | Accuracy\_Score |
| Default Decision Tree | 0.99144775040583932 |
| Decision Tree (max\_depth = 3) | 0.99315701539227186 |
| Random Forest | 0.99426196485490503 |

2. Confusion Matrix

Default Decision Tree

|  |  |
| --- | --- |
| 333896 | 1366 |
| 1521 | 789 |

Decision Tree (max\_depth = 3)

|  |  |
| --- | --- |
| 335262 | 0 |
| 2310 | 0 |

Random Forest

|  |  |
| --- | --- |
| 335118 | 144 |
| 1793 | 517 |

1. ***Discussion***

* To build a predictive model for this classification problem, I try to use SVM model firstly. Support vector machine is a [supervised learning](https://en.wikipedia.org/wiki/Supervised_learning) model used for [classification](https://en.wikipedia.org/wiki/Statistical_classification). However, it runs too slow to build a model with 1687861 records.
* Next model I used is decision tree. Decision tree is a [decision support](https://en.wikipedia.org/wiki/Decision_support_system) tool that uses a tree-like [graph](https://en.wikipedia.org/wiki/Diagram) or [model](https://en.wikipedia.org/wiki/Causal_model) of decisions and their possible consequences, including [chance](https://en.wikipedia.org/wiki/Probability) event outcomes, resource costs, and [utility](https://en.wikipedia.org/wiki/Utility). I used default parameters this time and got a good result. The accuracy score of this prediction is 0.991.
* To avoid overfitting, I set max\_depth = 3 to build a new decision tree. This time I got a better accuracy score: 0.993.
* In order to get a better predictive model, I tried random forest at the end. Random forest is an [ensemble learning](https://en.wikipedia.org/wiki/Ensemble_learning) method for [classification](https://en.wikipedia.org/wiki/Statistical_classification), [regression](https://en.wikipedia.org/wiki/Regression_analysis) and other tasks, that operate by constructing a multitude of [decision trees](https://en.wikipedia.org/wiki/Decision_tree_learning) at training time and outputting the class that is the [mode](https://en.wikipedia.org/wiki/Mode_(statistics)) of the classes. I got the best result among these 3 modes: 0.994.
* **Conclusion**

1. One of the effective way to avoid overfitting is set max depth for the decision tree model.
2. Random forest is an optimized version of decision tree which can predict the result more accuracy than decision tree.
3. SVM is not very suitable for a very large dataset.
4. ***References***

*1. Matt, Dancho (2017). Sales Analytics: How to Use Machine Learning to Predict and Optimize Product Backorders. Retrieved from* [*http://www.business-science.io/business/2017/10/16/sales\_backorder\_prediction.html*](http://www.business-science.io/business/2017/10/16/sales_backorder_prediction.html)

*2. Yuqi, Li (2017). AALBORG UNIVERSITY Backorder Prediction Using Machine*

*Learning For Danish Craft Beer Breweries. Retrieved from* [*http://projekter.aau.dk/projekter/files/262657498/master\_thesis.pdf*](http://projekter.aau.dk/projekter/files/262657498/master_thesis.pdf)

*3. Rodrigo, Santis (2017). Predicting Material Backorders in Inventory*

*Management using Machine Learning. Retrieved from https://www.researchgate.net/publication/319553365\_Predicting\_Material\_Backorders\_in\_Inventory\_Management\_using\_Machine\_Learning*

*4.* *Jason, Brownlee (2017). How to Handle Missing Data with Python. Retrieved from https://machinelearningmastery.com/handle-missing-data-python/*

*5. FastML (2017). Converting categorical data into numbers with Pandas and Scikit-learn. Retrieved from http://fastml.com/converting-categorical-data-into-numbers-with-pandas-and-scikit-learn/*

*6. Mode (2017). Python Histograms, Box Plots, & Distributions. Retrieved from https://community.modeanalytics.com/python/tutorial/python-histograms-boxplots-and-distributions/*