**Technical Report.**

**Problem: predict backorder problem for items.**

**Training dataset and data preprocessing:**

The training dataset consists more than one million of data records. However, there are only around 8900 records with “Yes” value in the field “went\_on\_backorder” (or Yes class), or less than 1% of the whole dataset.

This might lead to a problem when we train the machine learning models on the original training dataset, since it might cause bias against Yes class. Besides, the original dataset can also cause the Overfitting phenomenon to the trained models, which could negatively impact the predictive capability of these models.

To overcome such situation, we could do data preprocessing by reproduce a smaller training dataset. This could be achieved by selecting all 8900 records of Yes class, and withdraw another 8900 records of No class (data records with “No” in “went\_on\_backorder” field). The record are selected randomly from original training dataset and the numbers of both classes are kept equal so our model could be free from bias.

**Loading, transforming, and cleasing data:**

After preparing train data, we load it to our python program using Pandas. To fully optimize our training, we remove unnecessary columns. This data transforming and cleansing step help to reduce the training dataset’s size and remove redundancy in our data (some columns are dependent with other ones, for example: forecast\_3\_month, forecast\_6\_month, and forecast\_9\_month…) . The remaining columns are: national\_inv, lead\_time, forecast\_3\_month, sales\_6\_month, min\_bank, perf\_6\_month\_avg, went\_on\_backorder.

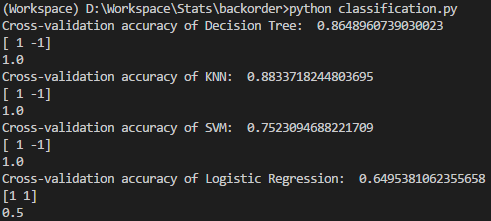
**Predictive Models:**

Some records have missing values in the field “lead\_time”, which is intepreted as NA, so we replace those missing values with numeric 0. After preprocessing and cleansing data, we split the training dataset into 2 parts: train data and train target. Because our goal is to classify all the test data into 2 classes: class 1 for backorder records and class -1 for non-backorder records. So, we replace the associated values of column “went\_on\_backorder” from “Yes” and “No” to “1” and “-1” respectively.

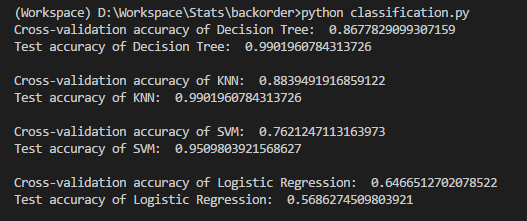
We consider 4 machine learning classifiers for this problem. They are Decision Tree, k-Nearest Neighbours, Support Vector Machine, and Logistics Regression.

**Performance and result:**

Predict the test\_set\_sample.csv (the predicted values are in square brackets)



Test with another 100 records withdrawn randomly from original dataset:



We could see that the Cross-Validation accuracy and real test scores for both Decision Tree and KNN are very high, while the accuracy of SVM in Cross-Validation stage is a bit lower. However, the performance of SVM in real test is also very high. On the other hand, the Logistic Regression perform kind of poorly in this problem.

**Conclusion:**

In this problem, the simple classifiers such as KNN or Decision Tree perform more efficiently than other algorithms such as SVM or Logistic Regression. The SVM model does not achieve a high Cross-Validation score but it still works very well on the real test data. The Logistic Regression does not provide a very convincing performance for this classification problem.

Hai Tran, M.Sc.