Product backorder prediction using different machine learning predictive models to minimize revenue loss in supply chain

Samiul Islam [1], Saman Hassanzadeh Amin [2, *]

Department of Mechanical and Industrial Engineering, Ryerson University, ON, Canada

[1] samiul.islam@ryerson.ca

[2, *] saman.amin@ryerson.ca (Corresponding author)

Product backorder prediction using different machine learning predictive models to minimize revenue loss in supply chain

Abstract

Prediction of backorders of products boosts up companies' revenues in many ways. In this work, the backorder of products is predicted using two machine learning models named Distributed Random Forest (DRF) and Gradient Boosting Machine (GBM) in H2O platform, and the performances of them are compared. We have observed that GBM successfully identified approximately 94 products out of every 100 products that go to backorder. We have noticed that the current stock level and the lead time of products act as the deciding factors of products' backorder in approximately 45% of the cases. We have shown how this model can be used to predict the probable backorder products before actual backorder can happen and how to visualize the impacts on inventory management. Based on the results of this paper, the decision threshold below 0.3 for high probable backorder products, and the threshold between 0.2 to 0.8 for low probable backorder products maximize organizational profit. The mentioned methods in this paper can be utilized in other supply chain cases to forecast backorders.

Keywords: Inventory Management; Product Backorder; Machine Learning; Gradient Boosted Machine, Supply Chain Management

1. Introduction

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 it will have a high impact on the respective company's revenue, share market price, customers' trusts and may end up with 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 or production costs and associated shipment costs (Carter and Rogers, 2008). Moreover, the reach on global market made customers' demands and expectations stochastic which made traditional supply chain management system less effective in many ways (Simchi-Levi et al., 2008), such as inaccurate demand forecasting or misclassifying of backordered products. Now a day, many companies are trying to predict the backorders of per unit product by applying a machine learning prediction process (Mitra, 2016) to overcome the associated tangible and intangible costs of backorders.

Backorder aging prediction (Rodger, 2014) can be feasible for the market with non-volatile demand where the lead time, price per unit, quantity of placed order and product stock level are the main drivers. But a sudden change in demand may raise other risk flags associated with the supply chain and may lead towards a loss (De Brito et al., 2008). To cope with the challenges of stochastic demand, a multi-objective inventory model has been introduced (Srivastav and Agrawal, 2016) and it has been mathematically proven that hybrid backorder (i.e, fixed and time-weighted backorder) inventory model is more efficient than fixed backorder inventory model in the market with volatile demand. To subside the stochastic demand problem, forecasting of partial backorders based on the periodic count on the current stock level (Xu et al., 2017) seems profitable but this process may exhaust local inventory system.

In this work, we have tried to provide a flexible inventory solution by predicting high probable backorder product and low probable backorder products by

employing machine learning models. First, we have investigated our dataset with a series of questions those are listed below, by constructing a relevant hypothesis and have performed hypothesis testing including the chi-square test.

Is the product out of stock resulting backorders?

Are the most sold items in the past month were backordered?

Are the operational limiting factors producing backorders?

Are the potential issues of products resulting in backorders?

The outcomes of our hypothesis tests have aided us to choose the appropriate machine learning model for prediction. Distributed Random Forest (DRF) and Gradient Boosting Machine (GBM) techniques (Ridgeway, 2006) have been chosen initially on H2O platform (Torgo, 2016). Based on their performances, we have selected a leader model for the prediction of backorder products. To resolve the imbalanced class problem, we have used synthetic minority oversampling technique (SMOTE) (Chawla et al., 2002) on our target class. The business impact of this prediction and how it can be beneficial for better inventory control is also discussed.

2. Literature Review

Machine learning (ML) techniques enables to forecast accurately multiple aspects related to supply chain management such as demand, sell, revenue, production, backorder and so on. Machine learning approaches were used to predict manufacturers' garbled demands where the researchers applied a representative set of traditional and ML-based forecasting techniques to the demand data to compare the accuracy of those used methods (Carbonneau et al., 2007). The researchers also found that the average performance of the ML techniques did not outperform the traditional approaches, but when a support vector machine (SVM) (Hearst et al.,

1998) trained on multiple demand series, it produced the most accurate forecasts. The same researchers extended their research work using Support Vector Machines (SVM) and Neural Networks (NN) (Funahashi 1989). The researchers found that the techniques of applying machine learning models provided noticeable improvements over the traditional model (Carbonneau et al., 2008).

An analysis of Supply Chain's demand prediction was carried out by applying the Support Vector Regression (SVR) method (Guanghui, 2012). The outcome of that investigation indicated that the prediction performance of SVR is superior to Radial Basis Function (RBF) (Chen et al., 1991), as SVR produced smaller results of the relative mean square error along with higher forecast accuracy of the supply chain. However, several factors were not taken in to account in that research work such as imbalance class problem, application of advanced machine learning techniques like neural network and ensemble methods due to the limitation of the computational resources.

To minimize the supply chain and inventory control costs, a risk based dynamic backorder replenishment planning framework was proposed (Shin et al., 2012) using Bayesian Belief Network. Similar framework was also prescribed (Acar and Gardner, 2012) using optimization and simulation technique.

A risk triggering model using fuzzy feasibility Bayesian probabilistic estimation of supply chain backorder aging was presented (Rodger, 2014) by employing stochastic simulation with Markov blankets to reduce inventory management costs.

To deal with the imbalanced class problem efficiently, machine learning classifiers were examined to identify a suitable forecasting model (de Santis et al., 2017). To carry out this task, investigators applied different measures along with the ensemble learning. The result of that investigation showed that the ensemble learning method provided feasible performance when considering precision-recall

curves and also minimizes the computational costs. The researcher also suggested applying different machine learning algorithms such as SVM and NN for the verification of potential performance improvements.

The prediction uncertainty in inventory model was addressed (Prak and Teunter, 2018) and a framework was proposed to estimate unknown demand parameters for better inventory decision.

The competition among different ML techniques also produces higher rate of accuracy of forecasts which improvised the necessitous decisions to increase revenue. An order policy-based inventory system model was proposed (Petropoulos et al., 2018) to observe the performance using ARIMA models (Zhang 2003), theta method and multiple temporal aggregations. Performances of ML models (Yu et al., 2018) with and without google trends were also measured to identify the trend of oil consumptions. Comparisons among different error measures such as mean square error (MSE), root mean square error (RMSE), mean absolute error (MAE), median absolute error (MdAE), mean absolute scaled error (MASE) and mean absolute percentage error (MAPE) were shown (Hyndman and Koehler, 2006) to indicate the models' performance. A new metric of measuring the performance known as mean arctangent absolute percentage error (MAAPE) (Kim and Kim, 2016) was also introduced and compared with other machine learning methods' error calculation. Performance of ML models such as Lasso, Extreme learning machine and Gradient tree boosting were evaluated to forecast future purchase trends (Martínez et al., 2018). The efficiency and impact of different types of forecasting methods were also measured for promotional products in the business (De Baets et al., 2018).

As per the recommendations of the previous works, we have tried to focus on implementing advanced machine learning techniques to predict backorder and have analyzed their performances to spot the efficient model. Table 1 shows a

comparative overview of our contribution relating with the previous research works.

Table 1
Review of some related papers

Authors	Prediction domain	ML models	Performance metrics	Flexible inventory control	Cost minimization
Carbonneau et al., (2008)	Manufacturers' garbled demands	SVM, NN	✓		
Guanghui, (2012)	Supply chain's demand	SVR, RBF	✓		
Shin et al., (2012)	Backorder replenishment planning			✓	✓
de Santis et al., (2017)	Material backorder in supply chain	LOGIST, CART, Ensemble	✓		
Prak and Teunter, (2018)	Prediction uncertainty			✓	✓
Proposed work	Product backorder in supply chain	DRF, GBM	✓	✓	✓

3. Exploratory analysis

The training dataset which is used for this analysis has 1,687,861 observations of 23 variables and the testing dataset has 242,076 observations of 23 variables. Both data sets contain a mix of features with floating point, integer and string values. The first column of each dataset is SKU which is known as the stock keeping unit which has 1,687,861 and 242,076 unique values respectively. That means sku has

unique values for each row of data. As this attribute is used for indexing purpose, we can ignore this column in our proposed model.

3.1. Null values and missing values

In both datasets, we have missing values for the 'lead_time' feature. In the training dataset, there are 100,894 missing values in lead_time which is 5.98% of the training data. Whereas in the testing dataset we have 14,725 missing values which is 6.08% of the testing dataset. We have assumed that the missing values are put as 'NA' in these datasets. The ‘lead time’ feature indicates the elapsed time between the placement of products’ orders and delivery of those products to the customers. In our training dataset, we have the maximum lead time of 52 hours, minimum lead time of 0 hours and the mean of lead time is 7.87 hours. We have replaced the NA values with the mean in the training dataset and we leave the testing data set as it is purposely. We have 0.00006 percentage of missing values in the other 8 features which is insignificant in amount to have an effect on the analysis part.

3.2. Class imbalance

The column ‘went_on_backorder’ is our target class which has 2 factors: ‘Yes’ and ‘No’. The ‘Yes’ class denotes that the product actually went on backorder. Unfortunately, we have only 0.669% data from 'Yes' class and 99.33% data from 'No' class. From this we can say that our data set is highly imbalanced. And, if we train our model with this imbalanced dataset, there is high possibility to have low model efficiency. Sometimes, it also may mislead by producing high accuracy of the classifier. As an example, if the classifier wrongly marked all predictions as “No” class, it will still produce 99.33% accuracy rate. Figure 1 shows the distributions of the classes for our target variable.

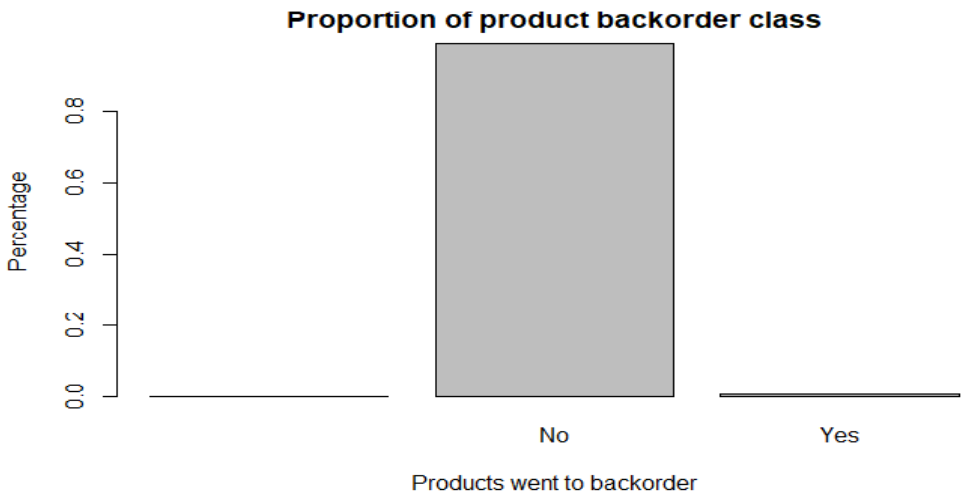

Figure 1: Proportion of class distribution of target variable

3.3. Principal component and important features

To find out the important feature, we have first looked at the training dataset samples and have tried to figure out how the data are distributed among different features.

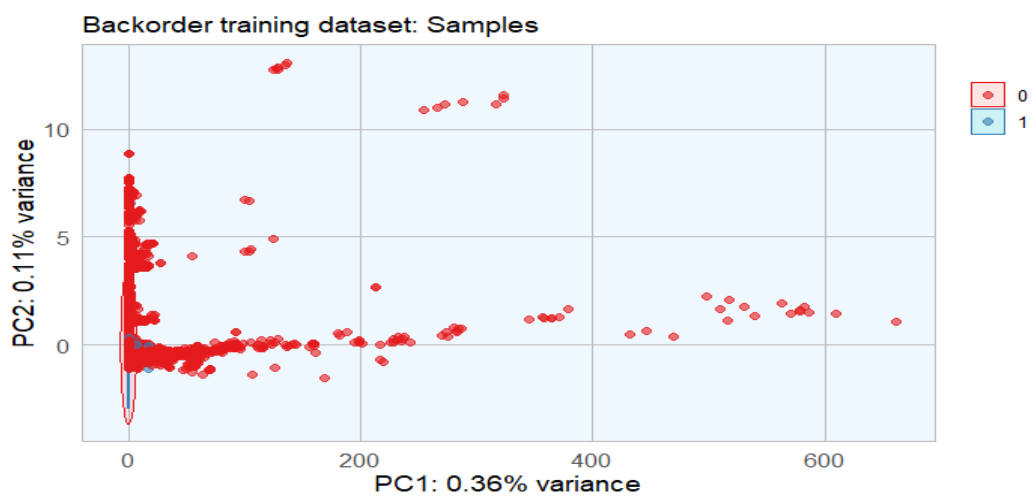

Figure 2: Distribution of samples among features

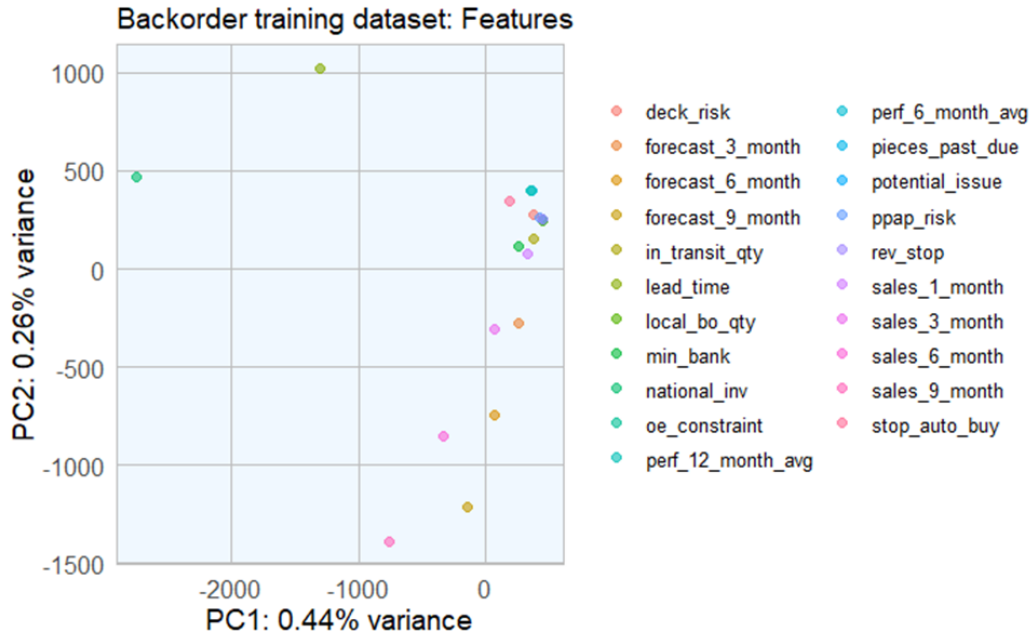

Figure 3: Magnitude of different features

From Figure 2, we can see that the data points of different features are close to their mean values as the variance is very small. The majority portion of data points is also identical and close to each other. We have also observed that the variance among the features as in Figure 3 are also small which has prevented us to draw a conclusion about the features those can be eliminated.

3.4. Correspondence and dimensions

We have total 7 categorical features composed of 2 factors – 'Yes' and 'No'. One of them is our target variable- 'went_on_backorder' and other 6 are 'potential_issue', 'deck_risk', 'oe_constraint', 'ppap_risk', 'stop_auto_buy' and 'rev_stop'. If any issue has been identified for a product, the column potential_issue' is updated with 'Yes' for that corresponding product and 'No' for otherwise.

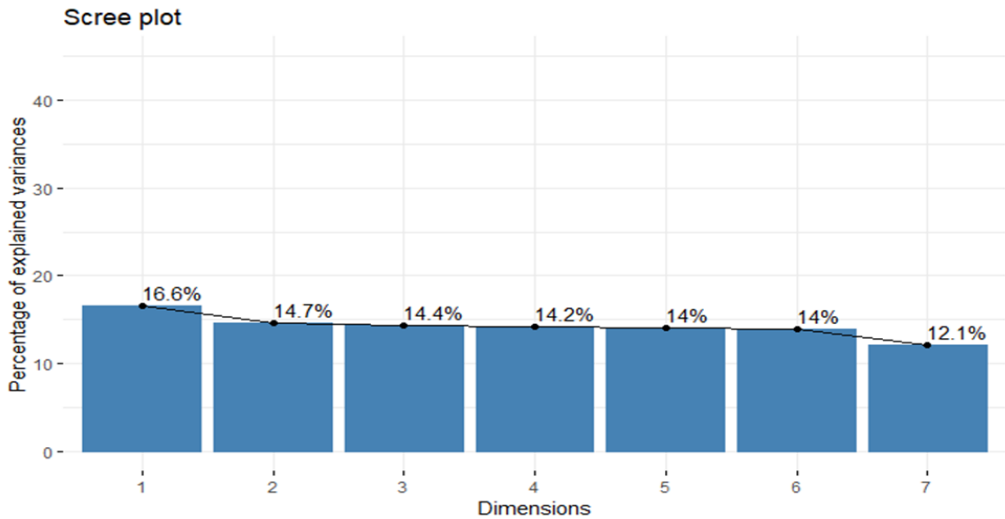

Figure 4: Correspondence among categorical features

The feature 'deck_risk' identifies the products those might remain in the shop/dock/stock and the 'oe_constraint' distinguishes those products which have some operational limiting factors such as bottle neck. The term 'ppap' stands for package/production part arrival process, hence 'ppap_risk' column refers to the risks associated with that process. The feature 'stop_auto_buy' indicates whether the automatic product selling process has been stopped or not and the 'rev_stop' indicates the revenue status for specific products. We have tried to examine the relationship of other categorical variables with our target variable. To do so, we have used multiple correspondence analysis and have found that there are small variances among those variables which are depicted in Figure 4. It also reflects the percentage of inertia in different dimensions.

Next, we would like to examine which variables are mostly correlated in each dimension. We have plotted those factor variables in 2 dimensions used as a coordinate as shown in Figure 5 and based on the squared correlation among those variables.

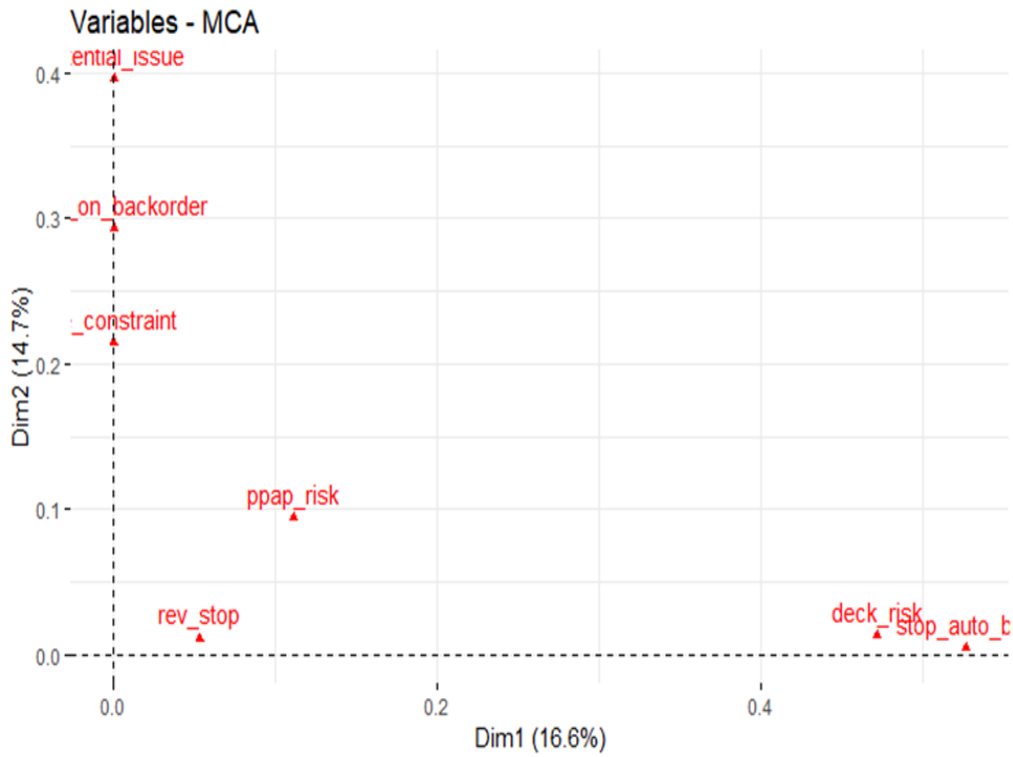

Figure 5: Correlations among categorical features

We have noticed that the features ‘rev_stop’, ‘deck_risk’ and ‘stop_auto_buy’ are mostly correlated with dimension 1 whereas ‘potential_issue’, ‘oe_constraint’ are strongly correlated with our target variable ‘went_on_backorder’ in the dimension 2.

3.5. Hypothesis testing

We would like to examine the relationship of the products those went on backorder with some features of our training dataset. First, we want to see whether the current level of products’ stock has an effect on the decision of backorder or not. We have assumed a null hypothesis that if the stock level of a product reaches to zero, it results in backorder. We have selected the significance level Alpha to

0.05, which mean that there is 5% chance of rejecting our null hypothesis when the hypothesis is actually true. We have observed that the p-value for this null hypothesis is far below than the significance level hence we cannot accept this hypothesis. However, the accuracy is high because of the high number of TN, probably due to the effect of imbalanced class. Next, we assumed that the most sold items in past 1 month went on backorder. For this, we have come up with our second null hypothesis which states if the quantity of a sold product in last 1 month is higher than or equal to the average number of sold products during that time span, that will go for backorder.

Table 2
Hypothesis testing summary

Question	Hypothesis	Result analysis	Decision
Is the product's out of stock resulting backorders?	H_0: $n = 0$; where n = Quantity of product in the stock. Ha: $n \geq 1$	TP: 3,197, FP: 5,196 FN: 83,582, TN: 1,253,544 p-value < 2.2e-16, for α=0.05	We cannot accept the null hypothesis as the p-value is insignificant.
Are the most sold items in the past month were backordered?	H_0: $n \geq \mu$; where n = Number of sold item of a product in last 1 month, μ = Average number of products sold in last 1 month. Ha: $n \leq \mu$	TP: 696, FP: 8,413 FN: 102,155, TN: 1,239,024 p-value = 0.9472, for α=0.05 Precision: 0.07640795 Recall: 0.006767071 F1 Score: 0.01243301	As our p-value is much higher than the significance level of 0.05, hence we may accept the null hypothesis H_0.
Are the operational limiting factors producing backorders?	H_0: $Backorder_{Yes} = OpConst_{Yes}$; Ha: $Backorder_{No} = OpConst_{No}$	TP: 6, FP: 9,103 FN: 177, TN: 1,341,002 p-value = 0.000117; for α=0.05	We cannot accept the null hypothesis as the p-value < 0.05.
Are the potential issues of products resulting backorders?	H_0: $Backorder_{Yes} = Issue_{No}$; Ha: $Backorder_{No} = Issue_{Yes}$	TP:40, FP: 9,069 FN:671, TN: 1,340,508 p-value < 2.2e-16; for α=0.05	We cannot accept the null hypothesis as the p-value is insignificant.

We have observed that the p-value is higher greater than the significance level, which allows us to accept this hypothesis. By accepting this hypothesis, we can tell that approximately 8 out of 100 product those went for backorder has a higher or equal number of selling quantity than the average number of sold products in last one month.

We have also examined whether the operational limiting factors such as bottle neck in different parts of supply chain are causing backorders or not. For this, we have assumed that if there exist some operational constraints for a product, that product would probably go for backorder. We have found our p-value is less than the significance level, thus we cannot accept this null hypothesis. Moreover, we assumed that if a product distinguished with potential issue, that product probably would not go for the backorder and set our fourth null hypothesis accordingly. As the p-value is much lower than the alpha value, we cannot accept this hypothesis. In both cases, we have observed very high accuracy which is approximately 99%, probably as because the imbalance class distributions where less than 1% of the products went on backorder. Table 2 shows the summary of Hypothesis testing results and decision.

4. Methodology

From the findings of our exploratory analysis, we have divided our work into different challenge sectors and applied different methods to overcome those challenges. The method implied in different parts are described below.

4.1. Preparing validation dataset

Though we have separated training and testing datasets, we would like to create another validation dataset from the training dataset. The idea of this validation set is to tune the parameters of our classifier/model before actual exposure to the test set so that the model can perform the test with minimum error and work efficiently. We have divided our training dataset into 80 and 20 percent randomly. Hence total observation of our training dataset becomes 1,350,288 and we have now a new validation dataset with 337,573 observations.

4.2. Handling missing values

As we have seen earlier, one of our dataset features named 'lead_time' contains 5.98% and 6.08% missing values in training and testing datasets respectively. As this is the time taken between the placement of order and delivery of the products, we have replaced the missing value in our training dataset with the mean value of the lead time. We could have removed the corresponding rows of the missing values but the idea behind of not removing the corresponding rows is that we want to keep all possible rows in our dataset for the efficient training of the model.

4.3. Dealing with the imbalance class

As we have explored that our target variable contains a highly imbalanced class with the percentage of 99.33% and 0.663% that would influence our model's accuracy. To cope with the imbalanced dataset, we have considered two techniques. First, we have used Synthetic Minority Oversampling Technique which under-samples the majority class and oversamples the minority class. While doing the under-sampling of the majority class, it also generates synthetic samples of minority class to prevent overfitting. We have manually set the oversampling of

minority class to 150 and under-sampling of majority class to 200. This tuning of under-sampling of majority class means that for every 200 of majority class, 1 minority class is generated, thus it avoids the overfitting scenario. And, by this tuning, we have achieved the balanced class of 50%-50% in our training and validation dataset. We have also used K nearest neighbour or KNN approach and in our case, we have chosen the nearest neighbour number as 7 when we have generated new artificial observations. The derived training set has lower observation numbers (36,436) in compare to the actual training set observations (1,350,288), which is almost 97% of reduction. Hence, we have decided to examine another data balancing method for the experimental purpose. In this case, we have used Random Over-Sampling Examples technique. This technique has provided us almost perfect balancing of classes by keeping the number of observations same.

4.4. Learning classifiers

Classifiers are algorithms those have been designed to examine the input data and able to divide those data in the different group of classes. In our experiment, we have used supervised learning classifier technique first and observe the outcomes. To do so, we have adopted the h2o scalable platform. This platform is composed of many supervised and unsupervised machine learning (ML) classifiers those can be run simultaneously in parallel.

5. Experiment

5.1. Experimental environment setup

We have initialized the h2o cluster to run multiple ML algorithms in parallel. Our data sets are in data frame format and h2o require the data in the h2o frame format. So, we have converted the datasets into h2o frame.

5.2. Model construction

We have constructed the h2o models for both synthetic minority oversampling technique and random oversampling example technique. We manually set the run time on the h2o cluster to 95 seconds. So that, among many learning classifiers, those classifiers who can make the run within 95 seconds can be grouped together. From that group, the top performed classifiers are added on the leader-board. At first, we would like to see the leader-board of smote data.

5.3. Model evaluation

Table 3 shows the H2O Model leader on smote balanced data. In our work, the model leader is the Gradient Boosting Machine (GBM).

Table 3
Performance metrics of GBM

Reported on training data	Reported on validation data	Reported on 5-fold cross-validation on training data
MSE: 0.04055084	MSE: 0.06294006	MSE: 0.06100714
RMSE: 0.2013724	RMSE: 0.2508786	RMSE: 0.2469962
LogLoss: 0.1439889	LogLoss: 0.2078432	LogLoss: 0.2075514

Mean Per-Class Error: 0.05458887	Mean Per-Class Error: 0.3207321	Mean Per-Class Error: 0.08239104
AUC: 0.9889262	AUC: 0.9539909	AUC: 0.9743189
Gini: 0.9778524	Gini: 0.9079818	Gini: 0.9486377

Finally, we have saved these leader models as a model object in our local disk for future use on new datasets.

5.4. Making predictions

We have used our saved models on the testing dataset and observed the performances. We have measured the performances by calculating classification accuracy, precision, recall, specificity, sensitivity, f1 score, misclassification error, and by visualizing Receiver Operating Characteristics (ROC) Curve along with the Area Under the Curve (AUC).

5.5. Assessment of business impact

We have assessed the prediction impact of our chosen model on business by visualizing inventory strategy effects. To investigate the expected profit of item with low probability of backorder, first we have derived the 'Expected Rate' from our models generated thresholds, TPR, FPR, TNR and FNR. Then we have assigned a positive value for the TP and a negative weight for FP for an item which can be chosen randomly. The TN and FN for that item can be assumed as 0. We have run the expected profit function by considering p1, cb_tp and cb_fp of that item, where p1 represents the set of predictions of the model, cb_tp represents the profit for correctly identified backorder and cb_fp denotes the expenses for incorrect prediction. Someone may confuse about the positive weight of cb_tp and

negative weight of cb_fp. For simple understanding let us assume a product price which correctly identified as backorder is $100. The product total cost including inventory costs is $60. If the product is currently placed as backorder, there will be always a chance of selling the product and gain the profit of $40. Similarly, the cb_fp may occur with expenses and can be weighted negatively. For the calculation of expected profit, we have adopted the standard expected value framework method (Provost and Fawcett, 2013) which is shown in the equation (1).

$$expected_{profit} = p_1 * (tpr * cb_{tp}) + (1 - p_1) * (fpr * cb\_fp) \quad (1)$$

By the prediction, we have set a threshold point that divides all products into 2 categories- probable backorder products and non-backorder products. That means any item over the threshold is identified as the backorder product. Products those are close to the threshold point can also be identified as the items with the low probability of backorder. Similarly, if a product resides far above the threshold, it would be identified as the items with the high probability of backorder.

6. Results

We have divided our result section into 2 parts. In the first part, we have shown how our model performed when it has been exposed to the testing dataset. In the second part, we decided to measure the business impact based on our model's output.

6.1. Model performance

When we have exposed our selected models on the testing dataset, we have observed that the performance of Distributed Random Forest drastically falls by producing a low accuracy rate, hence we have rejected that model. For simplicity,

we have not shown all the performance calculation of that DFR model, rather we have shown the ROC-AUC visualization of that rejected model. Apart from that curve, from this point, we have carried out with our GBM model.

6.1.1 ROC-AUC curve

In this part, we would like to show our models performance by visualizing Receiver Operating Characteristics (ROC) Curve along with the Area Under the Curve (AUC). The ROC curve tells us how our model performed throughout the prediction phase for all possible threshold values whereas the AUC represents the performance summary in a single value. So, the higher the AUC value, the most accurate the predictive model. The ROC curve is plotted considering True Positive Rates (TPR) in the y-axis and False Positive Rates (FPR) in x-axis in a scale from 0 to 1. The TPR and FPR are calculated for each threshold points of the classification process. The threshold points are the probability values that have been used to determine the class. In our model, we have achieved a maximum F1 score for the threshold value 0.416672. Which indicates that the maximum number of products those went on backorder have the probability values higher than the threshold value of 0.42. However, our model reaches maximum accuracy and recall at the threshold point of 0.576883 and 0.007325 respectively. Figure 6 depicts that the performance of our model is closer to 1 in the y-axis which contains TPR and that indicates the model's high performance. The diagonal red dotted lines represent the random guessing states for each threshold. The colored portion of the figure denoted as the area under the curve. The more the AUC value, the better the model is. We have achieved AUC value of 0.926489.

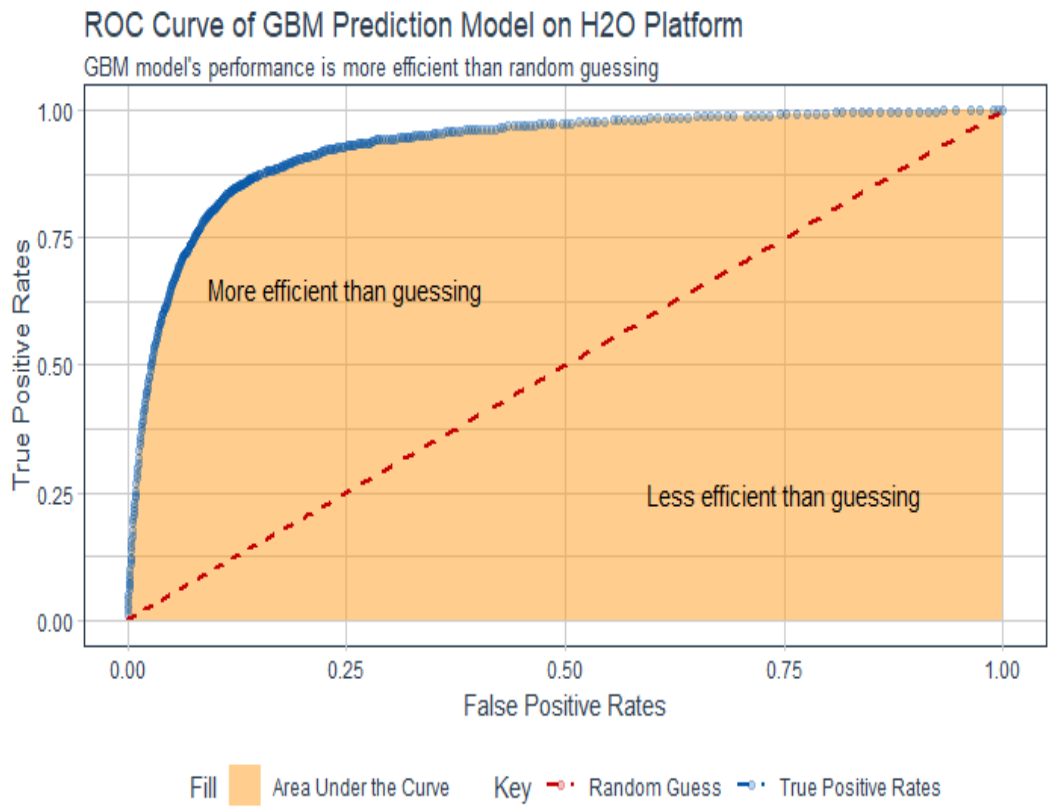

Figure 6: Area Under the Curve of GBM model

As we exposed the different Figure 7 shows the AUC of the DRF model and as the performance is much lower than GBM, we have rejected that Model. The main reason that H2O performed differently is relied on how the imbalance class problem is resolved.

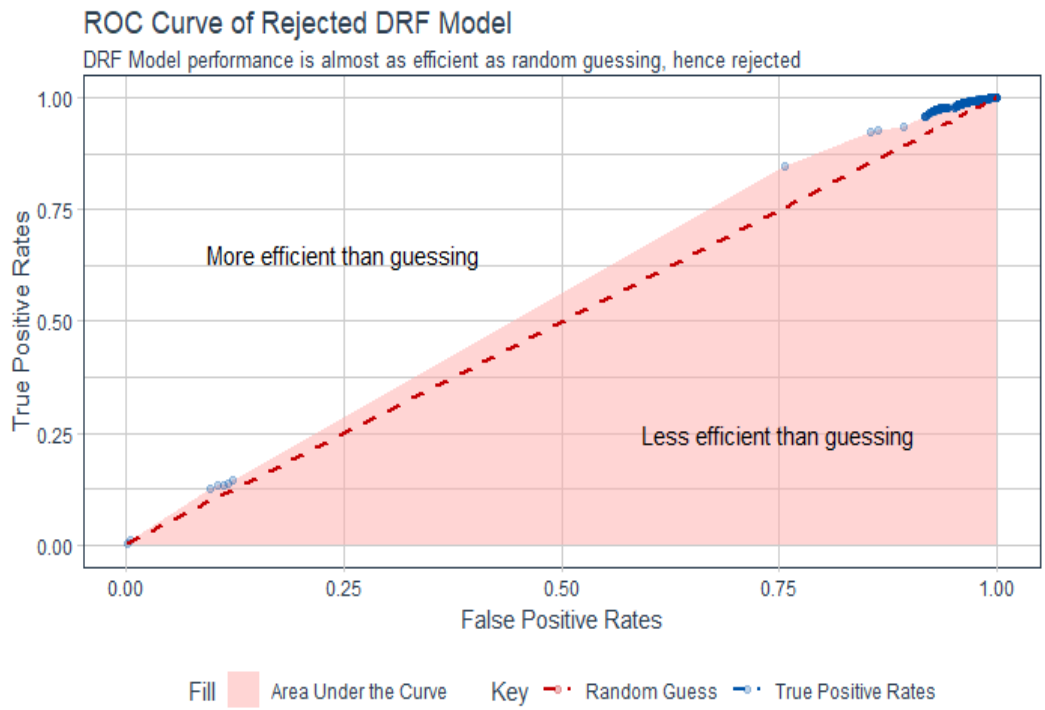

Figure 7: Area under the curve of DRF model

The reported AUC for this model is 0.55072 which is 40.5% less than the GBM. So, we can conclude that the impact of different imbalance class handling technique has reduced the AUC by 37.5%.

6.1.2. Variables considered

Figure 8 depicts the variables considered for the predictive modelling by GBM according to their order of importance.

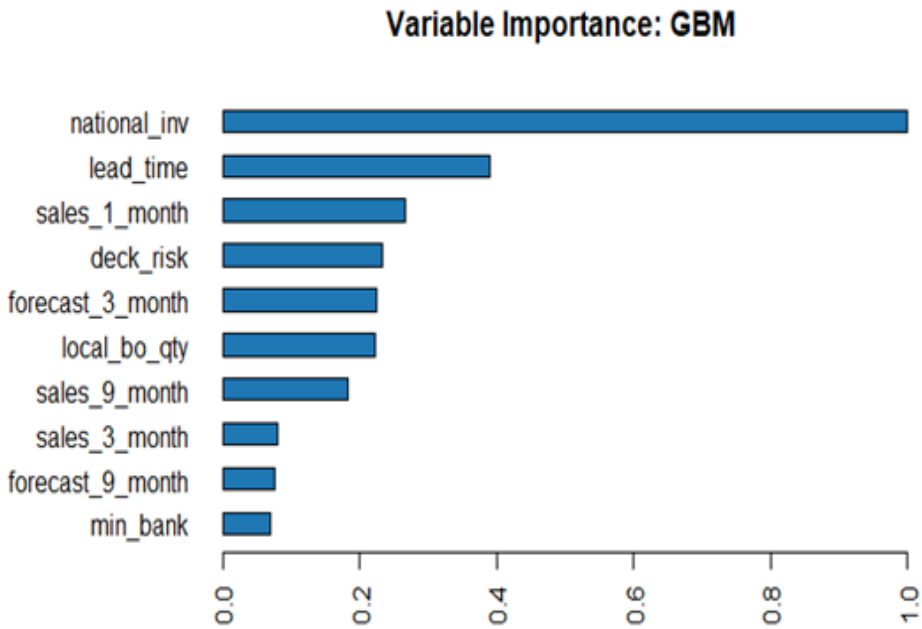

Figure 8: Important variables used for prediction

From Table 4, we can see that current stock level plays as a deciding factor in approximately 33% cases. Combine with the lead time, the approximation becomes almost 45% cases.

Table 4
Aattributes' influences

Features	Relative importance	Scaled importance	Percentages
national_inv	9,909.242188	1.000000	0.328441
lead_time	3,861.459229	0.389683	0.127988
sales_1_month	2,639.374268	0.266355	0.087482
deck_risk	2,307.009766	0.232814	0.076466
forecast_3_month	2,226.631104	0.224702	0.073801

As we are trying to predict whether a product will go on backorder or not, our prediction scenario falls under the binary classification. Hence, we have considered focusing on several binary classification metrics to investigate our model from different angles.

6.1.3. Confusion matrix

The first thing we would like to focus on our model's Confusion matrix as it is one of the easiest ways to get the glimpse of the correctness of the model. Moreover, most of the performance measures depend on the different term values of Confusion matrix. The confusion matrix consists of 4 terms, namely True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN). For the prediction run of our model on the testing dataset, our classifier perfectly classified 13,654 products went on backorder those were actually marked as went on backorder on the dataset, so our TP instances are 13,654. The predictive model also classified correctly 13,252 negative instances those products actually did not go for the backorder, so we get 13,252 TN instances. The classifier marked 1,267 products as did not go on backorder which actually did go on backorder in the dataset. Also, 917 products were wrongly classified as went on backorder. Hence,

we get 1,267 FN instances and 917 FP instances.

We have observed a high number of TN instances, probably as because most of the product in our dataset did not go on backorder. Moreover, someone may argue on the FN value of our classifier, but the high number of correctly predicted instances should overcome that concern. The distribution of TP, FP, TN, FN is shown as a Confusion matrix table in Table 5.

Table 5
Confusion matrix

		Actual values in the dataset	
		Backorder	Non-backorder
Models' prediction	Backorder	TP 13,654	FP 917
	Non-backorder	FN 1,267	TN 13,252

6.1.4. Classification accuracy

The classification accuracy can be interpreted as the number of correctly classified instances, divided by, the total number of predicted instances. We can put this idea in the equation (2) to get the classification accuracy of our model.

$$Model's\ classification\ accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (2)$$

$$= \frac{26906}{29090} \approx 0.9249$$

The classification accuracy of our model reflects that it predicted approximately 93 products correctly out of every 100 products whether those go on backorder or not.

6.1.5. Precision

By the precision measure, we would like to find out what proportion of products that we predicted as going to the backorder, actually went for the backorder. To do so, we have considered predicted positives which are TP and FP, along with the actual positives TP which represents the products those are classified as backorder also went on backorder. From equation (3), we can calculate the precision of our model which is almost 94%. That means this model can predict correctly 94% of the products those went on backorder.

$$Model's\ Precision = \frac{TP}{TP+FP} \quad (3)$$

$$= \frac{13654}{14571} \approx 0.937$$

6.1.6. Recall/Sensitivity

In this part, we would like to examine the true positive rate, also known as recall or sensitivity of our model. Sensitivity corresponds to the proportion of correctly predicted products those went on backorder, with respect to all backorder products in the dataset. From equation (4), we can see that the predictive model accurately classified 91.5% of products those went on back order.

$$Model's\ Sensitivity \quad = \frac{TP}{TP+FN} \qquad (4)$$

$$= \frac{13654}{14921} \approx 0.915$$

6.1.7. Specificity

Next, we would like to consider the Specificity of our model which is also known as the false positive rate. Specificity tells us what proportion of products those did not go for back order, also predicted by the model as non-backordered products. From equation (5), we can see that the model predicted accurately 93.5% of the non-backordered products.

$$Model's\ Specificity \quad = \frac{TN}{FP+TN} \qquad (5)$$

$$= \frac{13252}{14169} \approx 0.935$$

6.1.8. F1 Score

The F1 Score is also known as the Harmonic Mean between Precision and Recall which gives us the test accuracy. By using F1 score we can verify the robustness of our model as it tells us how many times the model correctly predicted backorder products together with how many times it correctly predicted non-backordered products. The calculated F1 score in equation (6) tells us that our model correctly classified 95% of the product for both backorder and non-backordered class.

$$\text{F1 Score of the } Model \quad = 2 * \frac{1}{\frac{1}{Precision} + \frac{1}{Recall}} \qquad (6)$$

$$= 2 * \frac{1}{\frac{1}{0.937} + \frac{1}{0.915}} = 0.95$$

6.1.9. Misclassification error

Misclassification Error is also known as Error Rate. By investigating Misclassification Error, we would like to see how often our model has predicted incorrectly. The lower the misclassification error, the higher the model's performance. We can calculate the Error Rate using equation (7) that tells us that our model incorrectly classified approximately 7 products out of every 100 products.

$$\text{Misclassification Error of the } Model = \frac{FP+FN}{Total\ number\ of\ Instances} \qquad (7)$$

$$= \frac{2184}{29090} \approx 0.075$$

6.2. Business impact

To understand the impact of our model's output in the business, we would like to visualize the recall and precision vs threshold those produced by the model during the classification process. As it has been shown in Figure 9, the cut off point for a maximum F1 score of our model makes the inventory strategy flexible. The higher the cut-off value the more rigid the inventory strategy decision. As an example, the higher cut off value of more than 0.85, every product gets the decision of backorder which we have shown as a 'YES' threshold in Figure 9. The problem of this rigid part that is above the 'YES' threshold, the model would not consider the current inventory level of the stocks of those products those fall in that category and place a decision of product backorder, sometimes which might lead to the redundant stock of those particular products. In our model, the rigid portion covers a very small area which can be considered as a positive attitude of our model.

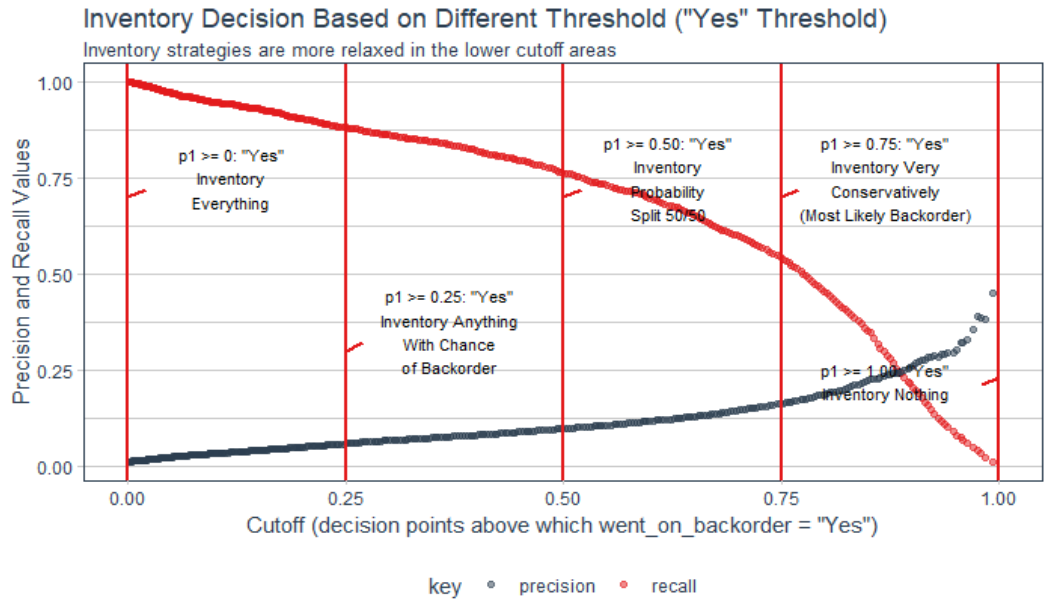

Figure 9: Inventory strategies at different cut-off

6.2.1. Investigate an expected profit of item with low & high probability of backorder

For the expected profit curve of low backorder probability, showed in Figure 10, we started with the p1 value as low as 0.1 and for the expected profit curve of high backorder probability, showed in Figure 11, we have started with the p1 value as high as 0.85, which is the maximum threshold for our F1. As depicted in Figures 10 & 11, the products with the low probability of backorder and the products with the high probability of backorder act reversely between 0 to 0.2 threshold values. As the threshold has increased, both type of products tends to generate zero profits or in other words, they are acting similarly at higher threshold values. The revenue generation curve of high backorder probability falls drastically with compare to the low probable backorder as the thresholds are increased. Organizations may take

interest to set different threshold values for these 2 categories of products to maximize their revenue. The products with the low probability of backorder can generate maximum profit between the thresholds 0.2 and 0.8 whereas for the products with the high probability of backorder can maximize profit at lower thresholds. Understanding these threshold values are crucial in inventory control and supply chain management.

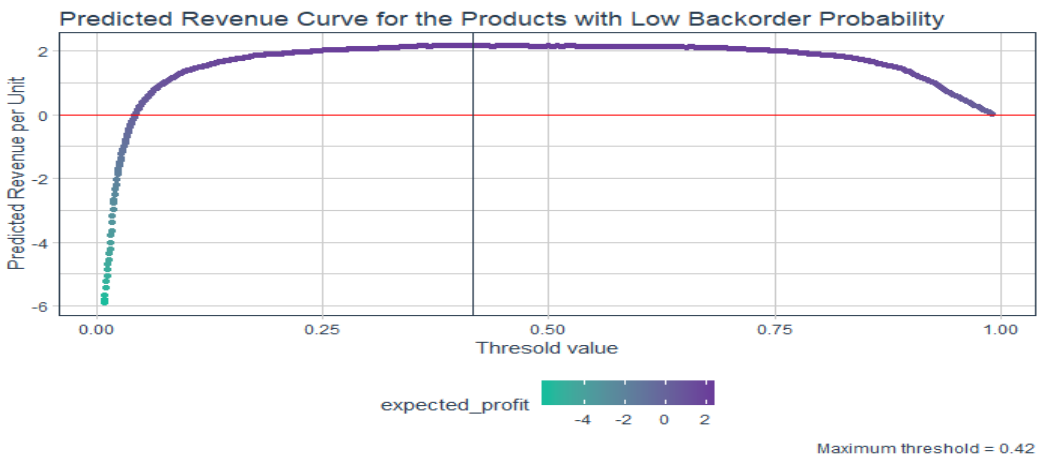

Figure 10: Predicted profit for the items of low backorder probability

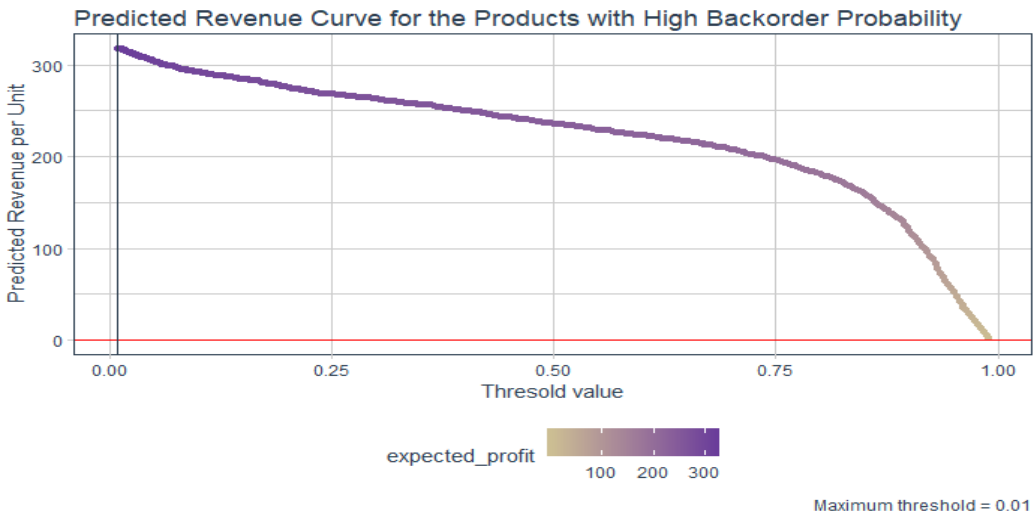

Figure 11: Predicted profit for the items of high backorder probability

6.2.2. Visualizing expected profit for the first seven items

To observe the outcome of our expected profit function, we have implemented it in the 7 hypothetical items with different inventory stock levels. We have then observed the characteristics of the curve at different threshold values which are shown in Figure 12. Though each item has been assigned different p1 values for starting and has been assigned different cb_tp and cb_fp values, we can identify the nature of backorders for the items over different threshold points.

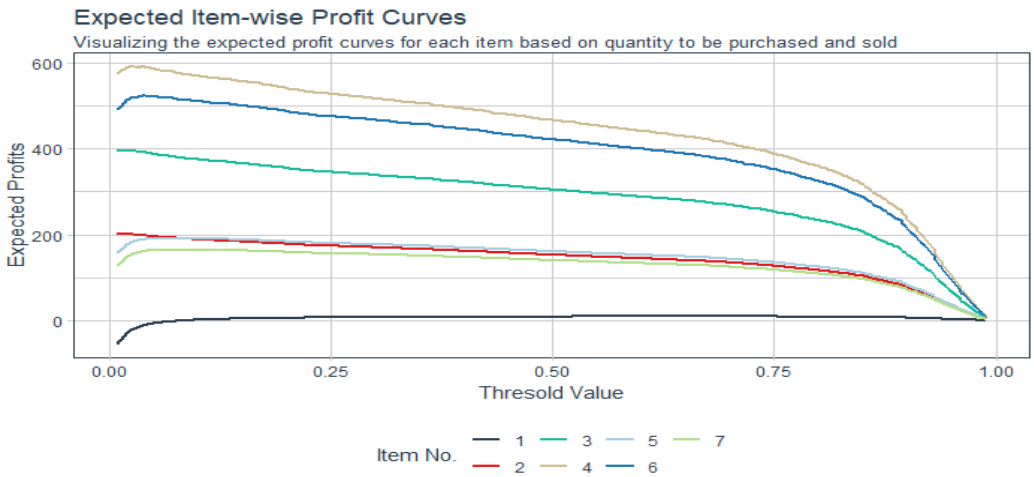

Figure 12: Expected profit for multiple items

6.2.3. Total extended expected profit based on threshold

Finally, we would like to determine the maximum profit regions in between different threshold values. We can achieve this by summing up the expected profits by thresholds to figure out the optimal strategy, which is shown in Figure 13.

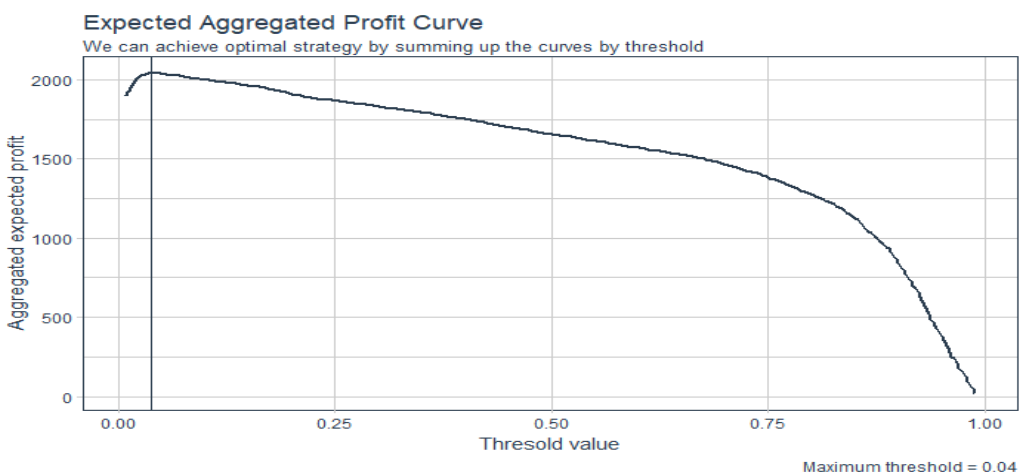

Figure 13: Aggregated profit curve for first seven items

7. Conclusions

As the market characteristics are varying rapidly based on customers' demand and expectations, flexibility in inventory control is required to maximize revenue. We have predicted the probable backorder items using machine learning techniques. We have then categorized those items into high probability and low probability of backorder items to assess business impact. We have shown that high probable backorder products can maximize profits at low thresholds. Correct identification of backorder probability and tune the threshold accordingly during inventory control play a critical part in boosting up revenues.

Although, by implementing our model we have achieve a higher cut-off range of approximately 0.85. From the cut-off value, inventory controller can have the overall idea of the product backorder trend, i.e., higher cut-off indicates lower possibility of backorders but sometimes make the system more rigid. Decision authority can manipulate this cut off point based on business demand, such as, to make the cut-off lower, we can intentionally increase the false positive rates such that the recall curve can fall sharply and produce new cut-off point. Playing with false positive rates is much safer in business area as the false positive prediction can only produce the production cost and that can be subsided if the product is sold.

Cost minimization can be done from multiple angles in an inventory control system and in this work, we have showed how backorder forecasting can allow a business to maximize its profits by employing flexible inventory control. However, the relation between sales forecast and predicted backorder of products is not examined in this work. The parameters we have used for backorder forecasting may have significant importance on the prediction of demand. As the uncertainty of demand plays a vital role to make the market volatile, the relationship between predicted demand and predicted backorder may also need attention. In our future

work we intend to come up with a model that will integrate these three perspectives and will provide more efficient and flexible inventory control system.

References

Acar, Yavuz, and Everette S. Gardner Jr. 2012. "Forecasting method selection in a global supply chain." *International Journal of Forecasting* 842-848.

Carbonneau, R., Laframboise, K., & Vahidov, R. 2008. "Application of machine learning techniques for supply chain demand forecasting. ." *European Journal of Operational Research,* 184(3), 1140-1154.

Carbonneau, R., Vahidov, R., & Laframboise, K. 2007. "Machine learning-Based Demand forecasting in supply chains. ." *International Journal of Intelligent Information Technologies (IJIIT),* 3(4), 40-57.

Carter, C. R., & Rogers, D. S. 2008. "A framework of sustainable supply chain management: moving toward new theory." *International journal of physical distribution & logistics management* 38(5), 360-387.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. 2002. " SMOTE: synthetic minority over-sampling technique." *Journal of artificial intelligence research* 16, 321-357.

Chen, Sheng, Colin FN Cowan, and Peter M. Grant. 1991. "Orthogonal least squares learning algorithm for radial basis function networks." *IEEE Transactions on neural networks 2* 302-309.

De Baets, Shari, and Nigel Harvey. 2018. "Forecasting from time series subject to sporadic perturbations: Effectiveness of different types of forecasting support." *International Journal of Forecasting* 163-180.

De Brito, M. P., Carbone, V., & Blanquart, C. M. 2008. "Towards a sustainable fashion retail supply chain in Europe: Organisation and performance." *International journal of production economics* 114(2), 534-553.

de Santis, R. B., de Aguiar, E. P., & Goliatt, L. 2017. "Predicting material backorders in inventory management using machine learning. ." *Computational Intelligence (LA-CCI), 2017 IEEE Latin American Conference.* IEEE. (pp. 1-6).

Funahashi, Ken-Ichi. 1989. "On the approximate realization of continuous mappings by neural

networks.” *Neural networks* 183-192.

Guanghui, W. A. N. G. 2012. “Demand forecasting of supply chain based on support vector regression method. .” *Procedia Engineering,* 29, 280-284.

Hearst, Marti A., Susan T. Dumais, Edgar Osuna, John Platt, and Bernhard Scholkopf. 1998. “Support vector machines.” *IEEE Intelligent Systems and their applications* 18-28.

Hyndman, Rob J., and Anne B. Koehler. 2006. “Another look at measures of forecast accuracy.” *International journal of forecasting* 22, no. 4 (2006): 679-688.

Kim, Sungil, and Heeyoung Kim. 2016. “A new metric of absolute percentage error for intermittent demand forecasts.” *International Journal of Forecasting* 669-679.

Martínez, Andrés, Claudia Schmuck, Sergiy Pereverzyev Jr, Clemens Pirker, and Markus Haltmeier. 2018. “A machine learning framework for customer purchase prediction in the non-contractual setting.” *European Journal of Operational Research.*

Mitra, A. 2016. “Fundamentals of quality control and improvement.” John Wiley & Sons.

Petropoulos, Fotios, Xun Wang, and Stephen M. Disney. 2018. “The inventory performance of forecasting methods: Evidence from the M3 competition data.” *International Journal of Forecasting.*

Prak, D., & Teunter, R. 2018. “A general method for addressing forecasting uncertainty in inventory models.” *International Journal of Forecasting.*

Provost, F. and Fawcett, T. 2013. *Data Science for Business: What you need to know about data mining and data-analytic thinking*. O'Reilly Media, Inc.

Ridgeway, G. 2006. “gbm: Generalized boosted regression models.” 1(3), 55. R package version.

Rodger, J. A. 2014. “ Application of a fuzzy feasibility Bayesian probabilistic estimation of supply chain backorder aging, unfilled backorders, and customer wait time using stochastic simulation with Markov blankets .” *Expert Systems with Applications* 41(16), 7005-7022.

Shin, K., Shin, Y., Kwon, J. H., & Kang, S. H. 2012. “Development of risk based dynamic backorder replenishment planning framework using Bayesian Belief Network. .” *Computers & Industrial Engineering,* 62(3), 716-725.

Simchi-Levi, D., Kaminsky, P., Simchi-Levi, E., & Shankar, R. 2008. “Designing and managing the supply chain: concepts, strategies and case studies.” Tata McGraw-Hill Education.

Srivastav, A., & Agrawal, S. 2016. “Multi-objective optimization of hybrid backorder inventory model.” *Expert systems with applications* 51, 76-84.

Torgo, L. 2016. *Data mining with R: learning with case studies.* Chapman and Hall/CRC.

Xu, Y., Bisi, A., & Dada, M. 2017. “A finite-horizon inventory system with partial backorders and inventory holdback.” *Operations Research Letters* 45(4), 315-322.

Yu, Lean, Yaqing Zhao, Ling Tang, and Zebin Yang. 2018. “Online big data-driven oil consumption forecasting with google trends.” *International Journal of Forecasting.*

Zhang, G. Peter. 2003. “Time series forecasting using a hybrid ARIMA and neural network model.” *Neurocomputing* 159-175.

Appendix A. Experimental Resources

Dataset link:

Training:
https://drive.google.com/file/d/1ZWy3e5KABVu02W6KrDcP_BKGkbTHP1FK/view?usp=sharing
Testing:
https://drive.google.com/file/d/1q6zwHxIuLG_sDKBWSdiO8RV-FrwzGdjn/view?usp=sharing

Source Code Link
https://drive.google.com/file/d/1Uzn_1W6CBGIbVGM0v1vbJ3gQJwVkSnXN/view?usp=sharing

Models' detail performance sheet:
https://drive.google.com/file/d/1OdVCTqanp7e7TtBAgPFP5nUlsUsChi77/view?usp=sharing