**Backorder Prediction**

## 1. Introduction

In the modern world of E-commerce, a wide range of products are readily available at customers disposal. Consequently, the number of orders handled by the E-commerce business holders in a day has sharply increased. Nevertheless, despite all the cutting-edge business models, machine learning techniques it's hard to cope up with extensive uncertainty in the number of orders and to maintain inventory according to it.

A customer order that cannot be filled when presented, and for which the customer is prepared to wait for some time is called a backorder. The percentage of items backordered and the number of backordered days are important measures of the quality of a company’s customer service and the effectiveness of its inventory management. Backorders are both good and bad; strong demand can drive backorders, but so can suboptimal planning. Sales, customer service, supply chain and logistics, manufacturing… no matter which department it is backorder is a major problem. Just look at Apple[7]. Every iPhone they’ve ever released triggered a demand so great that it resulted in backorders, and people were willing to wait. But Apple has an amazing track record of getting those orders to their customers on-time. The nature of the backorder and the number of items on backorder will affect the amount of time it will take before the customer can eventually receive the ordered product. Higher the number of items backordered, higher the demand for the item.

Backorders are an important factor in inventory management analysis. If a company consistently sees items in backorder, then this could be taken as a signal that the company is running too lean and that it is losing out on business by not providing the products demanded by the customers. When an item is on backorder, a customer might be looking elsewhere for a substitute product, especially if the expected wait time until the product becomes available is too long. This can provide an opportunity for once loyal customers to try other companies products and potentially switch.

## 2. Problem Definition

Inventory management is a risky business. Too many products on hand increases carrying a cost. Too little increases chance of back order. Product backorder may be a result of strong sales performance. For example, the product is in such a high demand that production cannot keep up with sales. However, backorders can be upsetting to consumers and can lead to canceled orders and decrease in customer loyalty. Companies want to avoid backorders, but also want to avoid overstocking every product. A review of data done by the FedEx representatives suggests that a single backorder costs company around \$11 - \$15. For example, an average company processing 1 million orders per year with 20% backorder rate would experience 200,000 backorders during the year. At a cost of \$13.28 per backorder, the total increased cost due to backorders is approximately \$2,656,000. Thus, predicting a backorder is an essential task in inventory control planning. Predicting backorders can improve the profitability of a business. Implementing machine learning to complete the task is the best way to go about solving this issue. Using the previous data, we are trying to predict the backorders to prevent cost associated with it. One of the most important benefits of implementing machine learning prediction model is the effect of precision and recall on inventory strategy. By selecting optimum predicting method, we can maximize expected profits.

A predictive model can identify which products are most likely to experience backorders giving the organization information and time to adjust. Machine learning can identify patterns related to backorders before customer orders. Production can then be adjusted to minimize delays, while customer service can provide accurate dates to keep customers informed and happy. The predictive analytics approach enables the product to get into the hands of a customer at the lowest cost to the organization. The result is a win-win; organizations increase sales while the customers get to enjoy the products they demand.

## 3. Data Description

The data given to us is a 16,00,000 * 23 data frame. The data consists of Numerical and Categorical data. While further analyzing the data, we realize that the categorical data is mostly Binary flagged as 0 or 1. The sku has a unique value for each row, so it is the index column and should be dropped. The numerical features have different scales, which may be a problem for some machine learning algorithms. The features should be rescaled to have similar scale. Out of all the columns, the Lead time column has around 5.98% outliers. perf_6_month_avg has 7.67% outliers and perf_12_month_avg has 7.23% outliers as (-99). Also the last row is not a valid sample in both the training and testing datasets, and thus it is dropped. The column went_on_backorder is the column which contains the labels specifying YES / NO for SKU which went on a backorder. The dataset is highly imbalanced as the items which went on a backorder consisting of only 0.7% of the data.

Table 1: Data Definitions

| Feature | Description | Type |
|---|---|---|
| sku | Random ID for the product | Numerical |
| national_inv | Current inventory level for the part | Numerical |
| lead_time | Transit time for product (if available) | Numerical |
| in_transit_qty | Amount of product in transit from source | Numerical |
| forecast_3_month | Forecast sales for the next 3 months | Numerical |
| forecast_6_month | Forecast sales for the next 6 months | Numerical |
| forecast_9_month | Forecast sales for the next 9 months | Numerical |
| sales_1_month | Sales quantity for the prior 1 month time period | Numerical |
| sales_3_month | Sales quantity for the prior 3 month time period | Numerical |
| sales_6_month | Sales quantity for the prior 6 month time period | Numerical |
| sales_9_month | Sales quantity for the prior 9 month time period | Numerical |
| min_bank | Minimum recommend amount to stock | Numerical |
| potential_issue | Source issue for part identified | Categorical |
| pieces_past_due | Parts overdue from source | Numerical |
| perf_6_month_avg | Source performance for prior 6 month period | Numerical |
| perf_12_month_avg | Source performance for prior 12 month period | Numerical |
| local_bo_qty | Amount of stock orders overdue | Numerical |
| deck_risk | Part risk flag | Categorical |
| oe_constraint | Part risk flag | Categorical |
| ppap_risk | Part risk flag | Categorical |
| stop_auto_buy | Part risk flag | Categorical |
| rev_stop | Part risk flag | Categorical |
| went_on_backorder | Product actually went on backorder. This is the target value. | Categorical |

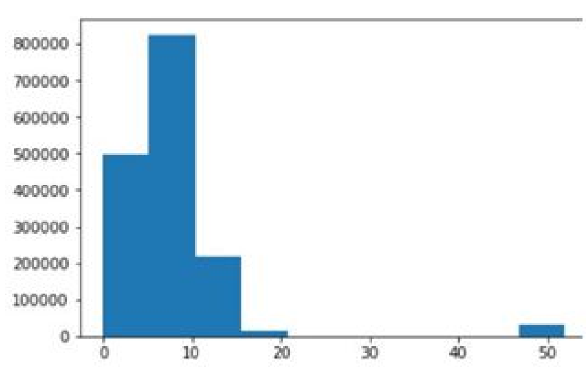
Table 3: Categorical data summary

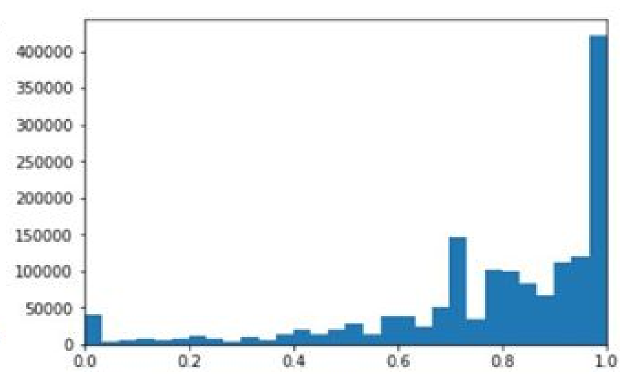
|  | count | unique | top | freq |
|---|---|---|---|---|
| sku | 1687860 | 1687860 | 3271681 | 1 |
| potential_issue | 1687860 | 2 | No | 1686953 |
| deck_risk | 1687860 | 2 | No | 1300377 |
| oe_constraint | 1687860 | 2 | No | 1687615 |
| ppap_risk | 1687860 | 2 | No | 1484026 |
| stop_auto_buy | 1687860 | 2 | Yes | 1626774 |
| rev_stop | 1687860 | 2 | No | 1687129 |
| went_on_backorder | 1687860 | 2 | No | 1676567 |

Table 2: Numerical data summary

|  | 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 | 1558382.0 | 0.782381 | 0.237014 | 0.0 | 0.70 | 0.85 | 0.97 | 1.0 |
| perf_12_month_avg | 1565810.0 | 0.776976 | 0.230490 | 0.0 | 0.69 | 0.83 | 0.96 | 1.0 |
| local_bo_qty | 1687860.0 | 0.626451 | 33.722242 | 0.0 | 0.00 | 0.00 | 0.00 | 12530.0 |

Fig 1: histogram for perf_6_month_avg & histogram for lead_time



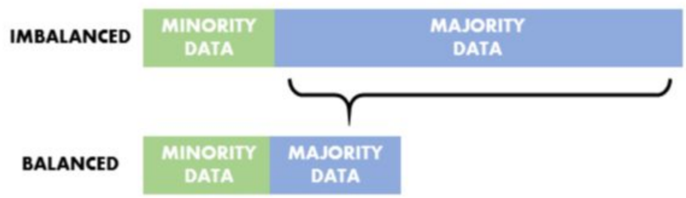

## 4. Proposed Methods

### 4.1 Dealing with Missing values and outliers

As mentioned in the data description above we have a number of missing values in the lead time variable which accounts to around 6% of the total lead time and outliers in perf_6_month_avg and perf_12_month_avg variables which are around 7% of the respective data. As shown in the Fig1. the lead time and the pref_6_months_avg are skewed so we imputed the missing data and outliers with medians of respective variable.

### 4.2 Imbalanced Models

The main problem with the data used is majority class significantly outweighs the minority class. To account for the imbalanced outcome.




**Approach to handling Imbalanced Data**

**Under-sampling:** Under sampling balances the class by randomly eliminating the majority class. This is done until the minority class and the majority class instances are balanced out. This method help improve run time and the storage problems by reducing the sample size. As this method removes the data randomly there can be loss of potentially useful information which could be important for building the classifiers.

**Over-sampling:** This method is opposite to Under sampling. In this method the minority class is randomly replicated. This method leads to no information loss but increases the number of instances thus increases the overall size of the data. This method increases the likelihood of overfitting since it replicates the minority class events.

**Synthetic Minority Over-sampling Technique (SMOTE):** This method is used to avoid overfitting which can occur when replicating the minority class. A subset of data is taken from the minority class as an example and then new synthetic similar instances are created. These synthetic instances are then added to the original dataset. The major disadvantage of this is while generating synthetic examples SMOTE does not take into consideration neighboring examples from other classes. This can result in increase in overlapping of classes and can introduce additional noise. Also, SMOTE is found not effective for high dimensional data.[3]

## 4.3 Implemented Models

**Random forest** is an ensemble classifier that consists of many decision trees and outputs the class that is the mode of the class's output by individual trees. We use Random Forest estimator because they perform well with large imbalanced datasets and also they are robust to outliers and skewed predictors. Random Forest also works better with high number of predictive variables and high sample size. This is because they captures the variance of several predictive variables at the same time and enables high number of observations to participate in a prediction.

## 4.4 Validation Method

**K-fold validation**

We use 2 Fold cross-validation to tune the model parameters as well as compare the model performance. Cross-validation is a method to evaluate predictive models by partitioning the original data into a training set to train the model, and a test set to evaluate it. In k-fold cross-validation, the original data is randomly partitioned into k equal sized subsamples. Of the k subsamples, a single subsample is used as the validation data for testing the model, and the remaining k-1 subsamples are used as the training data. The cross-validation process is then repeated k times (or the number of folds), with each of the k subsamples used once as the validation data. The k results from the folds then can be averaged to produce a single estimation. The advantage of this technique is that all observations are used for both training and validation, and each observation is used for validation exactly once.

**AUC (Area under the Curve)**

We use AUC ( Area Under the ROC Curve) as a validation metric instead of Accuracy because accuracy is a false metric to judge the outcome. Since, it is imbalanced to great extent. AUC measures how true positive rate (recall) and false positive rate trade off, so in that sense it is already measuring something else. More importantly, AUC is not a function of threshold. It is an evaluation of the classifier as threshold varies over all possible values. It is in a sense a broader metric, testing the quality of the internal value that the classifier generates and then compares to a threshold. It is not testing the quality of a particular choice of threshold.

## 4.5 Feature selection

For better results we need to examine the data properly. One the major thing to look for in any data is its features to avoid problems caused by curse of dimensionality. Features can be selected by few method such as practical considerations, expert opinion, correlation analysis. Here in our case we performed some analysis with domain knowledge we shortlisted some of the features to test how they contribute towards finding our goal. [3]

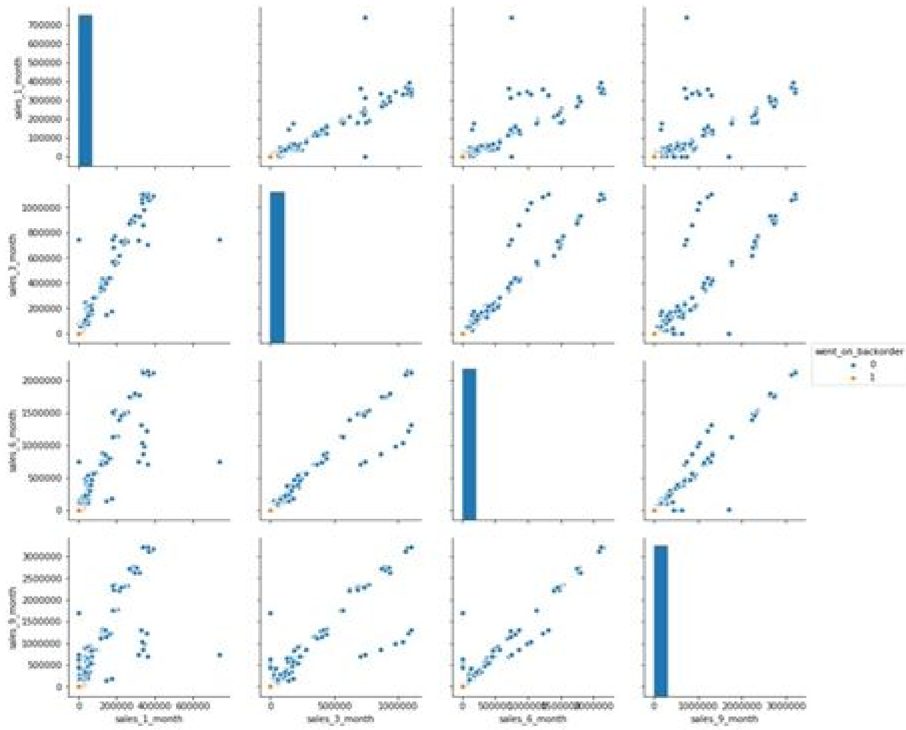



Fig 2:Sales Scatter Plot

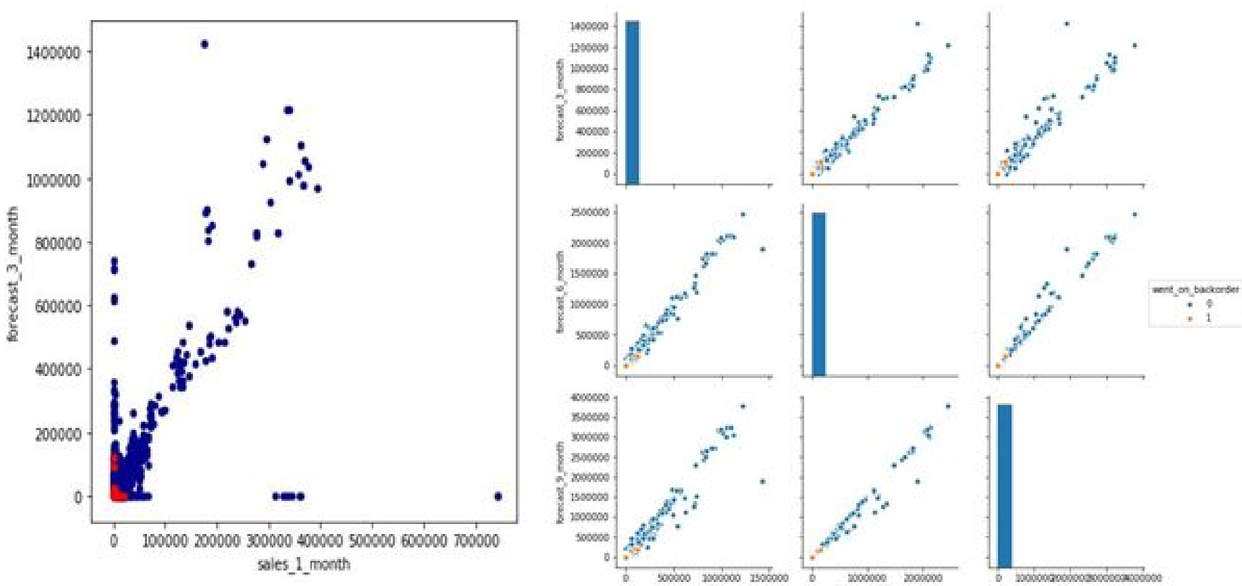



Fig 3: Scatter plot of forecast with sales (Left) Forecast scatter plot (Right)

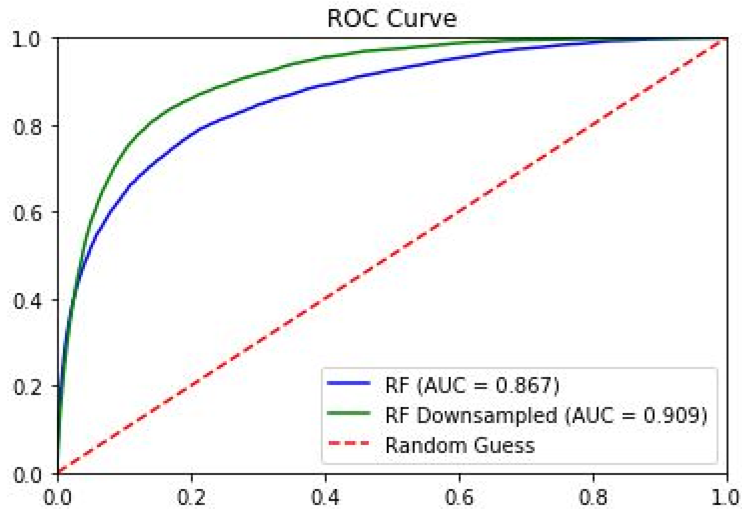
A regression analysis was performed on sales and forecast data from that we can see that the relationship between the variables are linear and Also, we observe that the backorders happen only when the value of sales and forecast is very low. We also observed that that the relation between the forecast and the sales follow a linear pattern and have a relatively high correlation between them.

The scatter plots show okay linear relationships between forecast, sales, in transit and recommended stock level. All the features range from 0 to over 300,000. Backorders only occur when the features are at low values.

Due to the good correlations and sufficiently linear relationships between these features, they can all be represented by a single feature in the machine learning models. The feature chosen is sales_1_month. This is because past sales is measured, whereas the quantity in transit, recommended minimum stock and forecasts are likely derived from past sales.

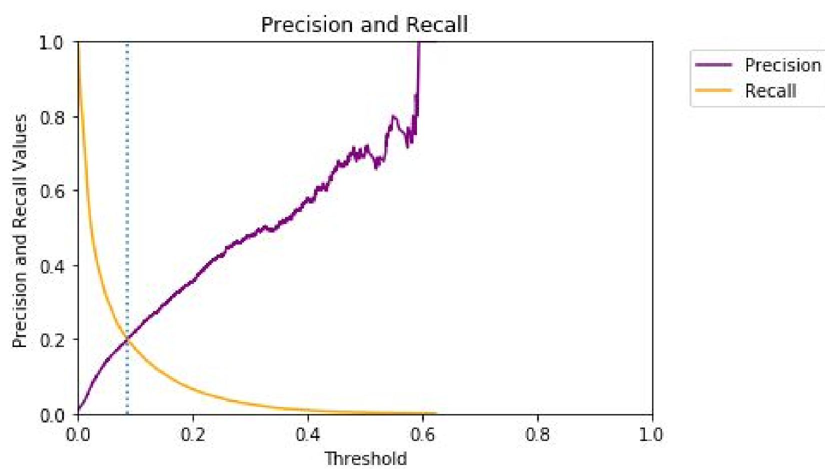
## 5. Experimental Results

By applying K-Fold cross validation we iterate our model by changing the min_sample_leafs generated and max_features used and then compare our results based on the iterations. We get the following ROC curve:




Looking at the curve, we see that the area under the curve for the downsampled data is 0.909 which is the best out of all the combinations. This curve was obtained for the setting max_trees : 50, max_features : 7, min_leaves : 5.

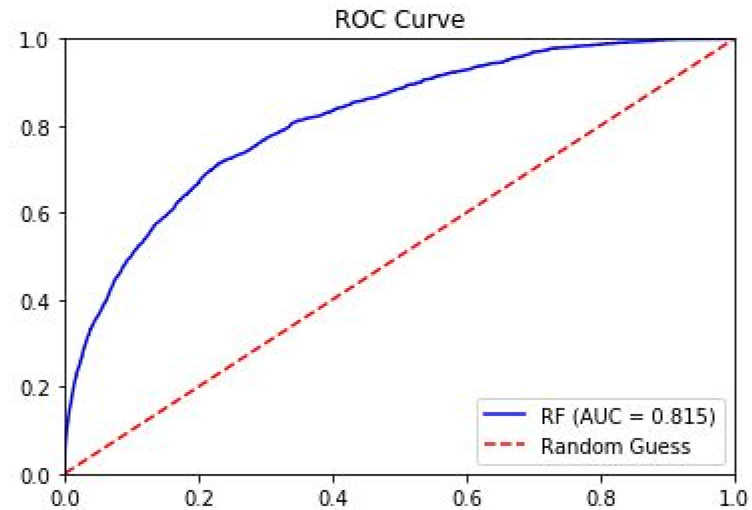
Further we also plot the Precision - Recall curves in order to define the optimal cutoff point. The graph looks like follows:



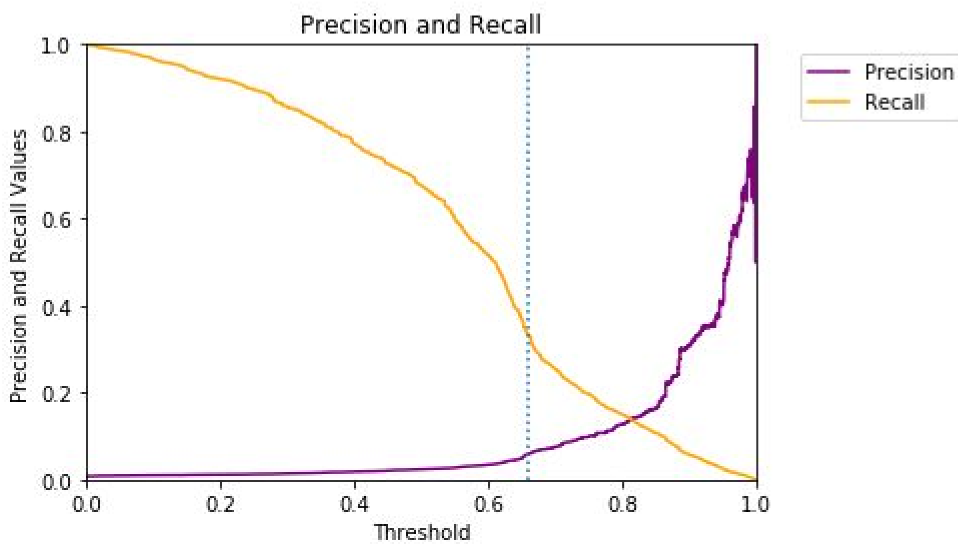
By looking at the graph we see that the optimal cutoff value is at 0.085. Thus, using this value as the cutoff value, we create the confusion matrix with an overall accuracy of 98.8%

Further we used the SMOTE method for sampling the unbalanced data and performed the same tasks. This resulted in a much lower accuracy and auc values for the same settings.

The ROC curve for the balanced dataset using SMOTE is:



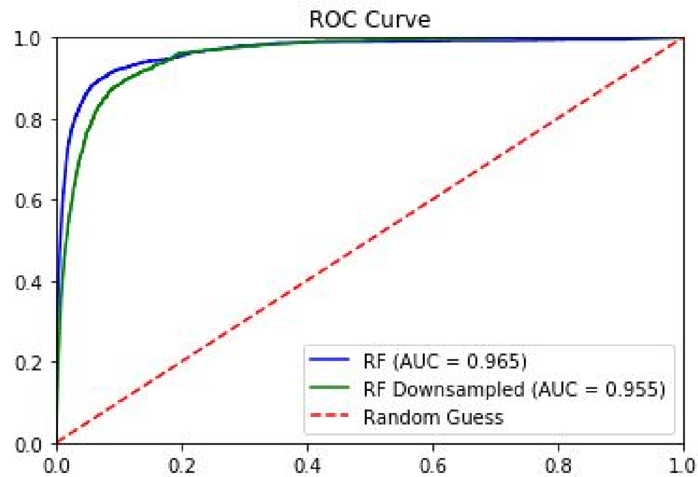

And the Precision - Recall graph is:



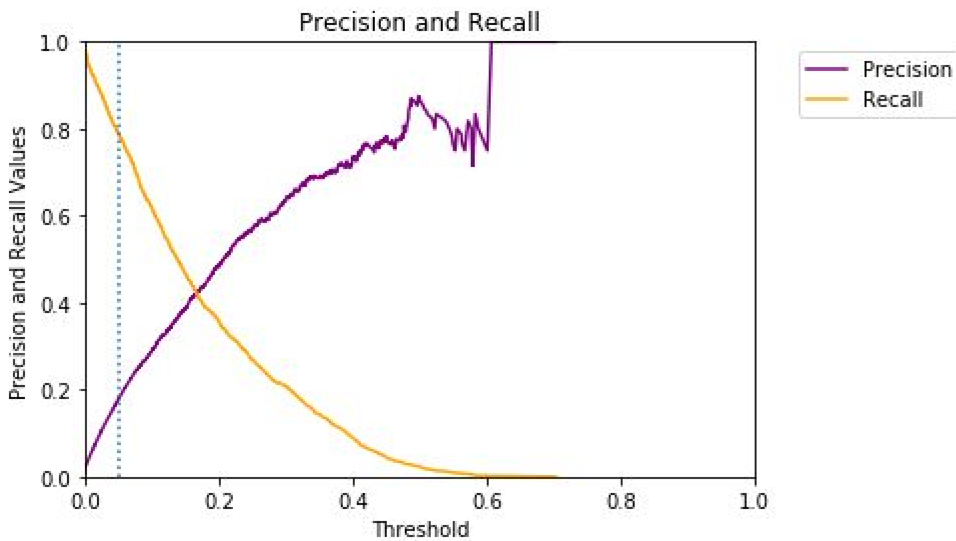

Thus, looking at the curves and the overall accuracy for the model, i.e. 95.66% which is lower that the accuracy of undersampling, we tend to select undersampling as a better option as compared to the oversampling.

We further run the model on the entire featureset to test our results in comparison to the results otherwise.

The ROC curve for that is:



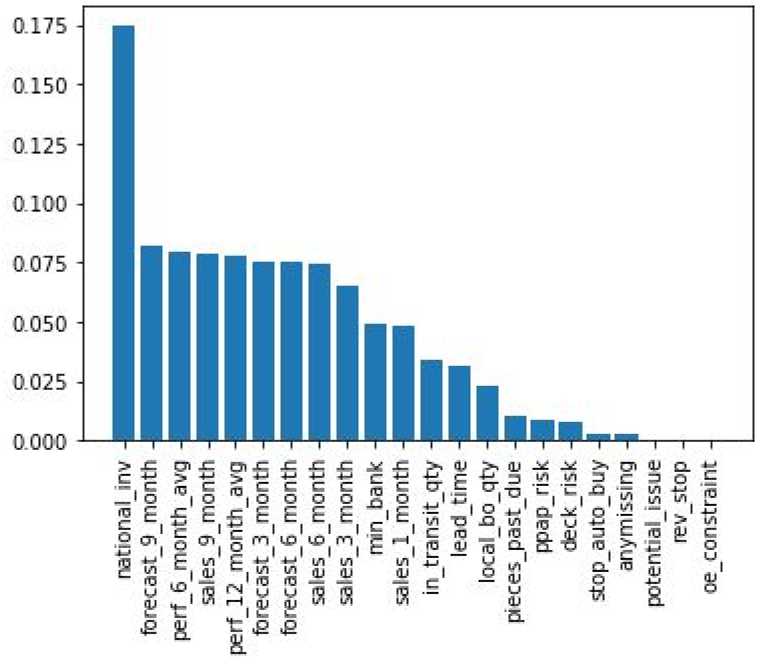

And the Precision - Recall curve is given as:




Thus, we clearly see the difference in the results. The overall accuracy is 95.5%, which is different.

After applying all the models and various sampling techniques, we come to a conclusion that the best method for sampling is undersampling and we leave it upto the user for this model to decide what suits him the best.

We further go ahead and check the importance level of each of the variable in the analysis, which is very well shown using the following graph:




6. Conclusion:

Observing the ROC curves and the Precision - Recall curves above we see that they vary a lot in terms of the precision and recall values. While we guessed that the reduced features should work the best given the correlation between data, but that is not the case. While we apply our model on the reduced data, we get a lesser value of AUC as compared to the AUC for the model applied on the entire dataset of 22 features. Also, there will always be a time complexity factor associated with the domain that needs to use this model. Thus, we leave that part for the experts to decide to costs associated with misclassification in a particular quadrant of the confusion matrix.

**References**

1. http://images.fedex.com/us/retail/backorders.pdf
2. [http://multichannelmagic.com/03/just-one-thing-the-backorder-effect-on-marketing/](http://multichannelmagic.com/03/just-one-thing-the-backorder-effect-on-marketing/)
3. R. B. de Santis, E. P. de Aguiar and L. Goliatt, "Predicting material backorders in inventory management using machine learning," *2017 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, Arequipa, 2017, pp. 1-6.
4. Production and Operations Analysis Sixth Edition Steven Nahmias
5. Galit Shmueli, Peter C. Bruce, Inbal Yahav, Nitin R. Patel, Kenneth C. Lichtendahl Jr.-Data Mining for Business Analytics Concepts, Techniques, and Applications in R-Wiley
6. SALES ANALYTICS: HOW TO USE MACHINE LEARNING TO PREDICT AND OPTIMIZE PRODUCT BACKORDERS (http://www.business-science.io/business/2017/10/16/sales_backorder_prediction.html)

7. https://www.tradegecko.com/learning-center/introduction-to-backordering