Machine Learning Engineer Nanodegree

Capstone Proposal

Hui-yu Yang
July 21st, 2017

Predicting Backorders

Domain Background

Backorders occur when supply cannot keep up with the demand. The customers' orders can't be filled or shipped due to the lack of inventory available. It seems more desirable than surpluses at first glance, but it could cost lowering satisfaction of customers, or even losing loyal customers. It is more of a problem for necessity than luxuries goods. The customers will likely purchase similar products from competitors for necessities, but they will be able to wait for luxuries goods since backorders create greater desirability due to either exclusivity or popularity.

There are many factors contribute to backorders, such as:

- incompetent management
- miscommunication
- inability to manufacture enough items to service the demand
- lack of data to accurately forecast the demand
- company strategy.

Problem Statement

We would like to prevent backorders by accurately predict which product will be backordered. When the product is a luxuries good, whether there is backorder or not is not important. Therefore, given that the products are all necessities, we need to determine the important features and a good model to prevent backorders. This problem is quantifiable, measurable, and replicable as this is a binary classification problem. We need to accurately classify whether there will be a backorder on the products or not. An accurate prediction of backorders can prevent competitor taking advantages of the situation, or potentially losing loyal customers.

Datasets and Inputs

The data for this capstone project comes from the Kaggle website Can you predict product backorders? (https://www.kaggle.com/tiredgeek/predict-bo-trial/feed). The training dataset has 1687861 samples with 22 features and one target variable. The list of variables in the dataset are:

- **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
- **went_on_backorder** - [Target variable] Product actually went on backorder.

All of the features are related to either inventory, sales, performance, or risk flag that may signal a backorder.

Solution Statement

Several supervised machine learning algorithms will be applied for this binary classification problem. The algorithms include:

- **Gradient Boosting Classifier**: a principled method of dealing with class imbalance by constructing successive training sets based on incorrectly classified examples.
- **Random Forest classifier**: bagging-based ensemble method.
- **Decision Tree classifier**: this method is based on 'gini index' or 'entropy,' which doesn't rely on accuracy.
- **AdaBoost classifier**: AdaBoost can achieve similar classification results with much less parameter tuning.
- **Bagging classifier**: ensemble method.

All of these methods are either emsemble methods or good at dealing with class imbalance.

Benchmark Model

The benchmark model is just a very naive model that always predict no backorder as the backorder rate is very low in the dataset. Due to the low backorder rate, the accuracy of this benchmark model will be high. Therefore, we need to examine its kappa value instead of accuracy.

Evaluation Metrics

Due to the highly imbalanced dataset with 1:143 ratio, we cannot rely on accuracy as it is misleading Because we have really imbalanced dataset with 1:143 ratio, we know we cannot rely on accurcy as it's misleading due to the accuracy paradox. Accuracy only reflects the underlying distribution when the data is imbalanced.

The metrics that we will examine are:

- **confusion matrix**: 2 by 2 table showing types of correct and incorrect predictions
- **precision**: measure of classification exactness
- **recall**: measure of classification completeness
- **F1 score**: weighted average of precision and recall
- **Cohen's kappa**: classification accuracy normalized by the imbalance of the data
- **ROC curve**: graphical examination of sensitivity vs 1-specificity.

Project Design

The project will split into 4 parts: data exploration, data preprocessing, model selection, and model evaluation.

Data Exploration

We will first explore the data by getting summary statistics for numerical and categorical features separately. After examining the features, we will explore outcome variable and see the exact backorder rate in the training set. Any variable that contain abnormal values will be further investigated. We will also examine the features within outcome groups and see if there are any feature that show a huge difference between products with and without backorders.

Missingness exploration will be performed as well in order to see if there are any missing data in the dataset, or if a particular feature contains more missing values than the rest of the features. We will also perform outlier detection by using Tukey Fences. Any data that are outlying for more than 1 feature will be removed, but we have to pay attention to the amount of data loss from this. We have to ensure the data loss is not substantial, or we can always apply a technique that is less affected by outliers in the analysis phase.

Data Visualization

In this section, we will make a boxplot of numerical features to examine their distribution. We will also construct a scatterplot matrix of the features, as well as the heatmap (correlation matrix) of the features. The variables exhibit a somewhat linear relationship in the scatterplot matrix will be further investigated, as well as the ones that are highly correlated in the heatmap. The variables that require special attention from these two plots should match.

Data Preprocessing

This is a straightforward preparation step for data analysis. We will convert any text data into dummy variables, such as Yes/No into 1/0. We will also impute the missing values that we found earlier with their median or mean based on the feature's distribution. We will also normalize the numerical variables before applying any machine learning techniques, so all the features have an equal variance and will not cause any ill-condition for methods such as PCA.

Model Selection

We will apply five supervised machine learning techniques on the processed dataset, and compare their performance based on the ROC curve, f1 score, kappa score, precision, recall, and confusion matrix. After we chose the specific machine learning technique, we will go through parameter tuning by first fix several parameters and improve them one at a time with grid-search and cross-validation.

Model Evaluation

We will compare the performance of the tuned model with the benchmark model by the performance metrics mentioned above.

References

Backorder. (2017, July 19). Retrieved from http://www.businesspundit.com/encyclopedia/general-business/backorder/ (http://www.businesspundit.com/encyclopedia/general-business/backorder/)