Machine Learning Engineer Nanodegree

Capstone Proposal

Adarsha Badarinath

March, 2017

Proposal

Based on historical data predict backorder risk for products

Domain Background

Part backorders is a common supply chain problem. Working to identify parts at risk of backorder before the event occurs so the business has time to react.

What is backorder?

Popular items may sell out quickly and temporarily be on backorder. This means that the items are currently out of stock but that there are shipments on their way to re-stock our warehouses.

Why is it helpful to predict it??

Whenever a backorder situation occurs it is very easy for a competitor to woo the customer to their product and even sell it at a premium taking advantage of the scarcity. Also sometimes businesses have to give discounts to retain the customer. It affects the bottomline of any business.

Problem Statement

Based on historical data predict backorder risk for products. The data is taken from kagel dataset. Using the dataset we predict 'went_on_backorder' column value given other input parameters. Explore different methods and different techniques to solve the problems. We want to answer the following questions on the dataset in the end

1. Given all the columns in the dataset predict 'went_on_backorder' column. Which technique is the best to do this?
2. Sometimes in real-world we donot have all the information available. So randomly erase a few features for the input and now try to predict 'went_on_backorder' column. Which technique is the best?

3. Find out which are the top 5 features which has the most impact on our predictions

Datasets and Inputs

The data set is taken from Kaggle dataset. The dataset is already divided into test and training set. It has the following columns/features.

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

- 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 - Product actually went on backorder. This is the target value.**

Solution Statement

Our solution is to predict *using supervised machine learning* with highest possible accuracy if an order can go into backorder using the given features. We want to build a model which will handle this prediction and also be general enough to predict similar inputs in the future with same accuracy.

Benchmark Model

There are many existing theories about backorder prediction few are listed below.

- Backorder prediction can make use of inventory Forecasting and Aggregate Planning. I have liked to a material from UT Dallas.
- A paper from Indiana University where they conclude that it was possible to predict backorders statistically.

However given above, I plan to develop my own methods using the machinelearning techniques developed in the class

Evaluation Metrics

We can evaluate my prediction % using the test data given the dataset

Project Design

My intended workflow

1. Import and explore the data set.
    a. Catch for any missing data
    b. Make sure it is not skewed in anyway so that learning algorithms get biased
    c. Compare it to common sense in real world
    d. Visualize different aspects of the data during exploration
2. Build multiple models of prediction - when all features are present
    a. Pipeline to prepare data for input into different models to do things such as normalization or adding new derived features (to reduce number of features).
    b. Explore multiple models of supervised learning and build models
    c. Predict using the models on the test data and pick the best model
3. Using the above models, build a more robust model when multiple features might be not not available in real world where data is messy.
    a. RANDOMLY drop many values in the training set
    b. Pipeline to prepare data for input into different models to do things such as normalization or adding new derived features (to reduce number of features).
    c. Explore multiple models of supervised learning and build models
    d. Predict using the models on the test data and pick the best model

4. Explore and answers questions on the data such as what is our prediction %. Also explore what a realworld company can do using this given information and also knowing that the model may not be 100% accurate.