Samuel Burden

August 19, 2017

Capstone Project

Product Classification Using Machine Learning

# Definition

## Project Overview

The ability to correctly classify data is a desire and need for any successful business. Unclassified data makes it extremely difficult for a business to glean very much useful information about their customers, competitors, employees, etc. The task of classifying data by hand though is an extremely large task that proves too costly and time consuming. This is where the power of machine learning is able to help both small and large businesses gain better insights into their data.

Product classification is one data classification problem that most companies face, especially with the proliferation of e-commerce where companies can sell millions of different products. The ability to categorize the products is necessary for financial reporting as well as making it easier for customers to find the products they need. If a customer cannot find a product because it was misclassified this is potential lost revenue for the company. Machine learning makes it possible to look at the various features or attributes of a product and make a prediction about the appropriate product category.

This project will specifically look at the product data of an e-commerce company name The Otto Group. The have over 200,000 products and need help with classifying them into appropriate categories. The origin of this project comes from a [Kaggle](https://www.kaggle.com/c/otto-group-product-classification-challenge#description) competition.

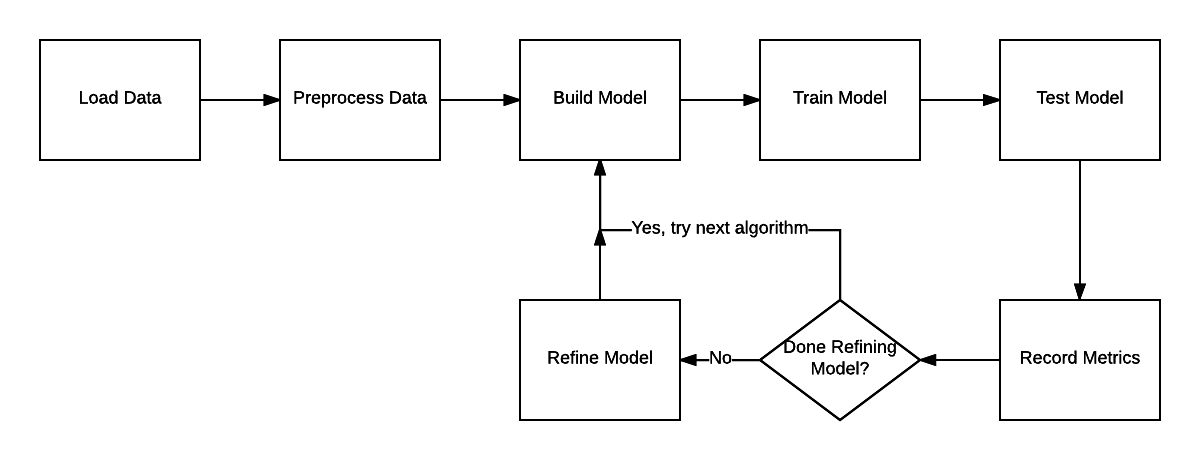
## Problem Statement

The Otto Group is a large e-commerce company that is looking to properly classify their products into nine different categories. The problem is how to take data from a subset of their product data and build a system for classifying all their products correctly. Machine learning classification algorithms are potential solutions to this problem.

The following machine learning classification algorithms will be evaluated to attempt to find the best model for classifying the products.

* XGBoost
* AdaBoost
* Neural Network

Each algorithm will provide metrics that will be used to compare each algorithm to determine the best model. A discussion of the specific metrics can be found in the next section of this document. A very defined workflow will be followed to evaluate each of these algorithms to arrive at the best model. A diagram showing this workflow can be found below:



Here are the steps in the above workflow:

1. Load the data from the training data provided by Otto Group.
2. Preprocess data by making training and test splits. The target category for a product also needs to be separated from the features. Also, any data normalization needs to be performed.
3. Build the model for one of the algorithms being evaluated.
4. Train the model on the training data split.
5. Test the model on the test data split.
6. Record the log loss/error and accuracy metrics.
7. If the model needs refining adjust the parameters of the algorithm and go back to step 3. If no further refining is needed go back to step 3 and build the model for the next algorithm.

## Metrics

Since this was a Kaggle competition, we have a clearly defined metric that was used to evaluate the various submissions. The log loss metric is one standard metric used in machine learning to help determine how well a model is performing. Generally, the lower the log loss the better the model. Below is the mathematical representation of log loss when multiple classes are being used in classification.



N = the number of products

M = the number of product categories, in this case 9

yij = 1 if product i is in category j

pij = the predicted probability that product i belongs to category j

Another simple metric that will be examined is accuracy. Accuracy can simply be calculated by summing all the correctly classified products and dividing by the total number of products in the dataset.

# Analysis

## Data Exploration

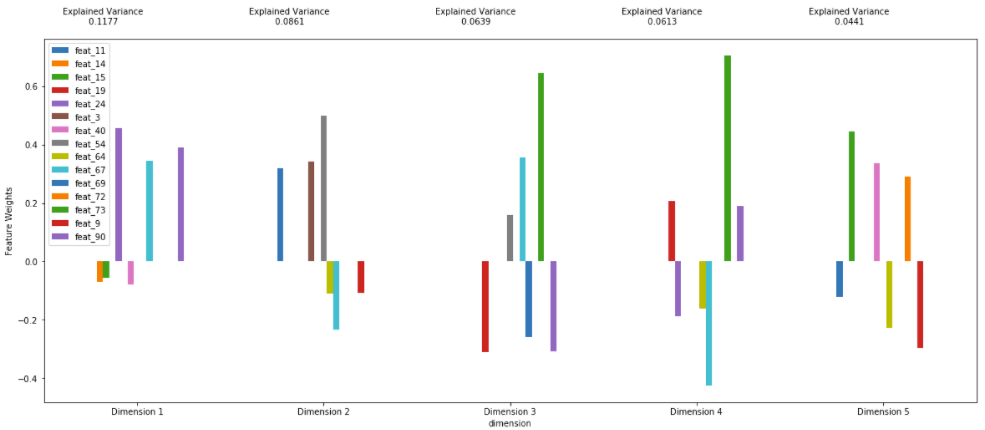
The Otto Group has provided some training data of over 60,000 products they currently sell. A link to the data can be found on [Kaggle](https://www.kaggle.com/c/otto-group-product-classification-challenge/data). Each product has 93 features that represent a relevant count for that feature in the product. The feature names have all been obfuscated to protect their competitive advantage, therefore the names of the features are simply feat\_1, feat\_2, …, feat\_93. Each of the products also has a target field that contains the product’s correct category. Since the data has been anonymized it is difficult to perform a deeper semantic analysis, but some overall statistics about the data have been provided below.

|  |  |  |
| --- | --- | --- |
| Statistic | Feature | Value |
| Highest Mean Feature | feat\_67 | 2.89765 |
| Lowest Mean Feature | feat\_6 | 0.025696 |
| Overall Max Value | feat\_73 | 352 |

All the features are numeric, specifically integers, as they are all counts and the range of values for all the features is between 0 and 352. One important point to notice when looking at the data is that it is fairly spare, with many of the features being 0 for a given product. Since every feature represents a count there is no need to do any feature transformation or normalization of each feature.

## Exploratory visualization

Principal Component Analysis(PCA) was performed to see if feature reduction was possible. After performing the analysis, it was determined that reducing the features didn’t provide a lot of benefit as the variance for the first component is only 0.1177 and the total variance for the top 5 components was 0.3731. In fact, it took the first 12 components to even reach an explained variance of 50%. Therefore, I decided to use all 93 features in the models being built. Below is a plot showing the first 5 components from PCA and how each individual feature contributes to its variance. We can see that some of the features that affect the variance the most are feat\_24, feat\_90, feat\_54, and feat\_73.



## Algorithms and Techniques

As stated previously, the three algorithms that will be evaluated in search of a solution are XGBoost, AdaBoost, and a Neural Network. Discussion of each of these algorithms can be found below.

### XGBoost

XGBoost is an ensemble supervised learning method that uses boosting to arrive at an optimized model[[1]](#footnote-1). One can think of boosting as taking a previous model’s errors and improving upon them by adding a new model. Models keep being added until no more improvements can be achieved. Finally, all the models are added together to arrive at a final model. XGBoost implements boosted decision trees as the models that keep being built sequentially. XGBoost is known for its execution speed and overall modal performance in being used as the algorithm of choice in many Kaggle competitons1. XGBoost performs well with sparse data and the dataset provided is fairly sparse for each product.

### AdaBoost

AdaBoost is another ensemble supervised learning method that uses boosting. It also uses decision tree models to sequentially improved upon until an optimal model can be reached. AdaBoost was the first really successful boosting algorithm developed for classification[[2]](#footnote-2). In fact, many of the current boosting algorithms build on the AdaBoost algorithm. AdaBoost works best on data that is void of outliers and noise. As all the columns in the product dataset are numeric it was easy to perform statistical analysis to test for outliers and look at the overall distribution of data. With little noise and outliers AdaBoost seems like a reasonable algorithm for finding an optimal solution.

### Neural Network

A neural network is a supervised learning technique where weights are applied to the input features to produce an output. The units/nodes that apply the weights and produce an output are called perceptrons. A neural network is just simply made up of several of these perceptrons and they together produce an output used for classification. Neural networks are great at learning non-linear models and support data in very high dimensions. Due to the number of features in the data and the potential that a linear model will not correctly classify the products, a neural network is an algorithm that should definitely be explored.

## Benchmark

Since this project was a Kaggle competition we have two different benchmark models and a metric for each benchmark has been provided which can be used to compare potential solutions. The metric provided in the Kaggle competition is the log loss of the model. The first benchmark is essentially just random guessing of the product category and has a log loss of 2.19722. The second benchmark is more sophisticated and uses the ensemble learning method of a Random Forest and achieved a log loss of 1.50241.

# Methodology

## Data Preprocessing

There was very little data preprocessing performed as the data is fairly clean since it was used for a Kaggle competition. As previously discussed, PCA was performed to see if the number of features could be reduced but based on the explained variance the decision was made to leave all 93 features as is. All the features are numeric counts therefore there was no need for normalization and since there are no categorical variables techniques such as one-hot encoding where not used either. The only preprocessing performed was to split the data randomly into training and testing splits, with 20% of the data being used for testing.

## Implementation

1. “A Gentle Introduction to XGBoost for Applied Machine Learning”, Jason Brownlee, http://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/ [↑](#footnote-ref-1)
2. “Boosting and AdaBoost for Machine Learning”, Jason Brownlee, http://machinelearningmastery.com/boosting-and-adaboost-for-machine-learning/ [↑](#footnote-ref-2)