Machine Learning Engineer Nanodegree: Capstone Proposal

Brian Campbell

September 8, 2019

Proposal

Domain Background

Many economists have claimed that one cannot predict individual winners and losers in the stock market and cite the Efficient Market Hypothesis to justify this claim (Investopedia). These economists believe that because one cannot pick individual stocks, investors should simply spread their risk amongst many stocks, i.e. investing in Exchange Traded Funds (ETF). While these economists are correct in identifying that traditional statistical methods have failed to identify individual winning and losing stocks, advancements in machine learning could make it possible to identify the winners and losers. Furthermore, there are a couple of benefits to investing in individual stocks than spreading risk in ETFs. First, investors can realize larger returns. Second, identifying financially sound businesses could encourage investors to take on more risks for social causes. For example, an investor might be willing to put more capital in a company that is developing the cure to Pancreatic cancer if a machine learning algorithm could reliably predict whether the pharmaceutical company was financially sound.

There is some research to suggest that machine learning can identify which companies are financially sound (Mattsson, Steinert and Dzemski, 2017). Bjorn Mattsson tested gradient boosting (GB), random forest (RF), and multilayer perceptron (MLP) machine learning algorithms on a dataset of Polish companies from 2000 to 2013 (Mattsson, Steinert and Dzemski, 2017). 64 different financial features were used to predict the binary target variable, bankruptcy (Mattsson, Steinert and Dzemski, 2017). Out of the three algorithms, the GB algorithm was best able to predict whether a company declared bankruptcy (Mattsson, Steinert and Dzemski, 2017).

Problem Statement

The problem to be solved is whether the MLP algorithm can be modified to perform better. Mattsson et al found that the RF and GB algorithms consistently outperformed

the MLP algorithm (Mattsson, Steinert and Dzemski, 2017). While Mattsson et al modified some of their hyperparameters, their MLP algorithm included weight decay in the activation function and they applied dropout to their hidden layer nodes, it is unclear whether they modified the hyperparameters beyond these measures (Mattsson, Steinert and Dzemski, 2017). This analysis would seek to improve the performance of the MLP by modifying parameters, e.g. the number of hidden layers and nodes, and the hyperparameters, e.g. the activation functions.

Datasets and Inputs

The dataset consists of financial indicators from polish manufacturing companies from the years 2000-2013 (Mattsson, Steinert and Dzemski, 2017; Tomczak, 2016). 64 financial indicators, such as net profit / total assets, total liabilities / total assets, working capital / total assets, retained earnings / total assets, EBIT / total assets, sales / total assets, total assets / total liability, gross profit / total assets, gross profit / sales, etc, are recorded for each company in the dataset (Mattsson, Steinert and Dzemski, 2017; Tomczak, 2016). These financial indicators are the traditional key performance indicators (KPIs) used by financial analysts to assess the health of companies. The target variable is a binary bankruptcy variable, 1 = bankrupt and 0 = not bankrupt (Mattsson, Steinert and Dzemski, 2017; Tomczak, 2016). The dataset is further broken down by the year in which the financial indicators for a company were recorded, consisting of five subsets total (Mattsson, Steinert and Dzemski, 2017).

Furthermore, the dataset is highly unbalanced. The number of companies that did not go bankrupt far exceed the number that did (Mattsson, Steinert and Dzemski, 2017). To correct this, Mattsson et al used the oversampling technique, meaning they provided the classifier with an equal number of non-bankrupt and bankrupt companies to the classifier (Mattsson, Steinert and Dzemski, 2017). This analysis will use the oversampling technique.

Solution Statement

The solution for the problem statement would be to test several MLP algorithms with varying parameters and hyperparameters to determine whether modifications to the MLP algorithm can improve performance. The modified parameters would include the number of hidden layers in the network and the number of nodes in these layers. The modified

hyperparameters will include modifying dropout on the hidden layers (adjusting the probabilities of a node dropping out), experimenting with different activation functions not mentioned in Mattsson et al's research, e.g. softmax, softplus, softsign, elu, and selu, and experimenting with different optimizers, e.g. stochastic gradient decent, adagrad, adam, and rmsprop (keras.io, 2019a; keras.io, 2019b; Mattsson, Steinert and Dzemski, 2017). Mattsson et al use the Area Under the Curve (AUC) method derived from Receiver Operating Characteristics (ROC) curves to measure the accuracy of their MLP algorithm, which is what this analysis will use to measure the efficacy of the MLP algorithms.

Benchmark Model

Mattsson et al's MLP model is the benchmark model (Mattsson, Steinert and Dzemski, 2017). The AUC scores for each of the five data subsets, one for each of the five years of their analysis, ranged from 0.83 - 0.92. This analysis would seek to improve the AUC scores.

Evaluation Metrics

The evaluation metric here is an AUC score. The AUC is the area under the two dimensional graph with the true positive rate as the y-axis and the false positive rate as the x-axis (Google Developers, 2019). A model that has predictions that are 100% correct has an AUC of 1, while a model that has predictions that are 100% wrong have an AUC of 0 (Google Developers, 2019).

Project design

First, the data will be cleaned and organized using Mattsson et al's methodology (Mattsson, 2017). Second, Mattsson et al's MLP algorithm will be imported to determine whether the data was sufficiently cleaned (Mattsson, 2017). If the algorithm returns different AUC scores, these scores will serve as a signal suggesting that the data was not prepared using Mattsson et al's methodology (Mattsson, 2017). Finally, several MLPs will be observed with varying parameters and hyperparameters to determine whether these MLPs can produce better AUC scores than the benchmark MLP.

Furthermore, the data will be divided into sub datasets based upon years. Eg. for individual years 2007-2011, bankruptcy prediction models will only be based upon data

in the individual years. 2007 will have its own model; 2008 will have its own model, etc. This method accomplishes two goals: first, it eliminates look ahead bias, and second, it allows this analysis to compare its results to that of Mattsson et al's (Mattsson, Steinert and Dzemski, 2017; Walimbe, 2017). Mattsson et al split their data by years and only applied models to an individual year (Mattsson, Steinert and Dzemski, 2017).

References

Google Developers. (2019). Classification: ROC Curve and AUC | Machine Learning Crash Course | Google Developers. [online] Available at: https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc [Accessed 6 Sep. 2019].

Investopedia. Efficient Market Hypothesis (EMH) Definition. [online] Available at: https://www.investopedia.com/terms/e/efficientmarkethypothesis.asp [Accessed 4 Sep. 2019].

Mattsson, B. (2017). bamattsson/neural-bankruptcy. [online] GitHub. Available at: https://github.com/bamattsson/neural-bankruptcy [Accessed 6 Sep. 2019].

Mattsson, B., Steinert, O. and Dzemski, A. (2017). Corporate Bankruptcy Prediction using Machine Learning Techniques. [online] Available at: https://pdfs.semanticscholar.org/cd5f/aff7c02bcba0b3f7f438d4d1c38c3d30d43e.pdf [Accessed 4 Sep. 2019].

Tomczak, S. (2016). UCI Machine Learning Repository: Polish companies bankruptcy data Data Set. [online] Archive.ics.uci.edu. Available at: https://archive.ics.uci.edu/ml/datasets/Polish+companies+bankruptcy+data [Accessed 5 Sep. 2019].

Keras.io. (2019a). Activations - Keras Documentation. [online] Available at: https://keras.io/activations/ [Accessed 5 Sep. 2019].

Keras.io. (2019b). Optimizers - Keras Documentation. [online] Available at: https://keras.io/optimizers/ [Accessed 5 Sep. 2019].

Walimbe, R. (2017). Avoiding look ahead bias in time series machine learning modelling. [online] Linkedin.com. Available at: https://www.linkedin.com/pulse/avoiding-forward-bias-time-series-machine-learning-rohit-walimbe-1 [Accessed 8 Sep. 2019].