Stock Market Trend Prediction with Neural Net and Natural Language Processing

Eliska Kloberdanz, Jeremy Roghair, Matthew Burke
Iowa State University, Department of Computer Science
eklober@iastate.edu, jroghair@iastate.edu, mburke@iastate.edu

Abstract

Machine learning is currently on of the most researched areas in AI amongst both academia and industry. It is a very popular AI technique that can be applied in many fields to solve different problems. In this paper, we explore the application of machine learning in the field of finance. In particular, we are predicting the U.S. stock market daily trends using two machine learning models, a neural net and natural language processing, and we simulate trading decisions based on these predictions - essentially creating an automated/algorithmic trading platform.

The two main financial models used to predict future asset prices are technical analysis and fundamental analysis. Technical analysis exploits historical data such as price and volume to come of with patterns, whereas fundamental analysis also considers intrinsic value, expectations, and economic and political climate. In this paper, we implement the technical analysis with a neural net model to predict whether the next day's S&P 500 index closing price will go up or down that we trained on historical price data and various technical indicators. The fundamental analysis is implemented through a natural language processing model that analyzes news articles published by Wall Street Journal to gauge the daily market sentiment and evaluates it either as positive or negative. We then combine the outputs of these two models to generate a trading signal to buy or sell or hold. If the neural net model predicts that the price will increase or if the natural language processing model says that the market sentiment is positive, a buy signal is generated. The results show that the returns from our trading strategy outperform the market returns. However, it should be noted that our strategy is to both buy and sell, which is relatively risky. When the strategy is changed to buy and hold, the returns are only slightly above the market. Nevertheless, this is expected since higher risk means higher return.

Ⅰ. Introduction

Financial predictions with regards to market changes and performance have historically been challenging. A wide assortment of techniques and methods have been developed in order to determine how an investor should expect the market to change and provide an indication of whether they should either buy, sell or hold certain assets. Two of the most important financial models used to predict future asset prices are technical analysis and fundamental analysis. Automating these analyses and using computer programs to trade originated in 1970's and is referred to as automated or algorithmic trading. (Huang et al., 2018). Since then algorithmic trading as been growing and in 2011 it accounted for 73% of U.S. equity trading volumes. (Treleaven et al., 2013) However, it was only within the last decade or so that automated trading expanded to include machine learning algorithms which have proven to be more robust at making accurate and reliable predictions for market performance.

A major theory in finance that describes the behavior of asset prices is the efficient market hypothesis (EMH) states that asset prices reflect all available information. The hypothesis has three forms: markets can be weak efficient, semi strong efficient, or strong efficient. The weak form of EMH assumes that past prices are already fully incorporated in asset prices. (Fama, 1970) The general consensus of the finance academic community is that markets are usually at least weak efficient, which is why our model incorporates not only past historical price data, but also current news into the the stock price prediction. Even though the finance academia believes that the markets are at least weak form efficient and that trading strategy based on technical analysis does not yield abnormal returns, technical analysis is the most common model implemented on algorithmic trading platforms in practice. (Huang et al., 2018) In addition to that, the EMH assumes that all market participants are equally informed, which is not necessarily a realistic assumption. (Ruta, 2014) Therefore, we believe that there is a merit to out machine learning model utilizing technical analysis.

Ⅱ. Methods & Approach Overview

The basis of our approach stems from the idea of making trading decisions to buy or sell financial assets depending on the output of a model comprised of various inputs that are believed to influence the economy and subsequently financial markets. To this end, we developed a strategy for determining this, which leverages two different machine learning models into a single trading platform.

The first model is a neural network which is used to evaluate the financial aspects of the market. In particular we input daily historical prices of the S&P 500 index with the thought of it being a good indicator of market performance on a particular day. Additionally, we derive several different financial indicators from this data including simple moving averages, exponential moving averages, momentum, bollinger percentages, daily high-low price changes and daily open-close price changes. The model outputs a binary prediction based on today's financial inputs, indicating whether the market will rise or fall tomorrow.

The second model utilizes a naive Bayes classifier for natural language processing in order to extract sentiment analysis from news articles for a particular day. The model is trained using movie reviews which generate a binary response of either positive or negative sentiment. News articles related to the financial markets from the past five years were then gathered and ran through the model. The output produced was used to determine the overall sentiment for financial markets for a particular day.

The two models are then combined to form two different trading strategies: (1) Buy & Sell, and (2) Buy & Hold. In case of the first strategy, a decision to buy is made when either the neural net predicts that the next day price will increase or when the natural language processing model assesses that the market sentiment is positive. If neither condition applies, the model makes a decision to sell. The second strategy works the same way, except for it just holds a position instead of selling when neither condition applies.

Technical Analysis: Neural Network

Financial indicators are one of the most important factors that dictate future market performance. Therefore, creating a model that can accurately predict future performance is central to determining when it's best to buy and sell various assets. The neural network model we developed takes in 68 years of daily historical opening, closing, high and low prices for the S&P 500. From this data we derive several financial indicators, historically used to evaluate market performance. These include those listed below. Here P reflects the closing price unless otherwise notated:

- High-Low price differences
  - $(P_{high} - P_{Low})$
- Open-Close price differences
  - $(P_i - P_{open})$
- 3-Day Simple Moving Average (SMA)
  - Average previous 3-days closing price for a particular day
    - $\frac{Pi-3 + Pi-2 + Pi-1}{3}$
- 20-Day Simple Moving Average
  - Average previous 20-days closing price for a particular day
    - $\frac{Pi-20 + ... + Pi-2 + Pi-1}{20}$
- 5-Day Standard Deviation
  - Standard Deviation of the previous 5-day closing prices
    - $\sqrt{\frac{1}{5}\sum_{i=1}^{5}(P_i - \overline{P})^2}$
- 5-Day Exponential Moving Average (EMA)
  - $EMA_i = (P_i * \frac{1}{3}) + (EMA_{i-1} * (1 - \frac{1}{3}))$
- 50-Day Exponential Moving Average
  - $EMA_i = (P_i * (\frac{2}{51})) + (EMA_{i-1} * (1 - (\frac{2}{51}))$
- Bollinger Percent
  - $(P_i - (SMA - 2\sigma)) / ((SMA + 2\sigma) - (SMA - 2\sigma))$
- Momentum
  - $(P_i - P_{i-10})$

The neural network implemented uses a supervised learning methodology, where it infers the best fit to the data through a labeled response variable. The response chosen for our model was *price rise* which is set initially to a binary value of {0,1} for the historical data, indicating a rise (1), or fall (0) in next day's closing. Combing the *price rise* indicator with the historical prices, and financial indicators mentioned above constituted our base data set for the neural network model. The data was split with a 80/20 ratio, with 80% being reserved for training the model, and

20% as a hold out sample in which to test model performance. The model was trained using a multi-layer perception (MLP) classifier available within the sklearn python library. MLP consists of at minimum three different layers, an input layer, hidden layer and output layer. Each layer consists of multiple neuron nodes using a activation function to output an input into subsequent nodes which ultimately result in a classification in the output layer. MLP utilizes backpropagation in order to train the MLP model.

The model was tuned by using grid search from the sklearn library in order to optimize the parameters that best fit the model. Grid search uses a cross validation for part of its evaluation of various models. We set cross validation to three and five fold and the resulting tuned parameters for the multilayer perceptron were using four layers of size 20, 15, 10, 5, an activation function of *logistic,* the *adam* optimization function to minimize the loss function, and a learning rate = 0.001. The model displayed slightly high variance, so a small regularization parameter of = 0.001 was used to reduce overfitting and ensure the model converged.

Sentiment Analysis: Natural Language Processing

The stock market is driven by new information and also, according to behavioral economics theories, by people's emotions. (Huang et al., 2018) Our natural language processing model utilizes the nltk library and the naive Bayes classifier to assess whether news articles on a particular day were negative or positive.

Due to the unavailability of free market sentiment information on a daily basis, we trained our model on 2,000 positive and negative sentences relating to movie reviews that we obtained from the nltk library that includes various datasets. The training set was passed in as a list of tuples each containing a sentence and an indication if it is positive or negative. Then a dictionary containing words as tokens was created based on these sentences. The training set was then passed as input into the naive Bayes classifier, a probabilistic classifier which applies the Bayes' theorem and relies on strong independence assumptions between features. (Rish, 2001 & McCallum and Nigam, 1998)

In the next step we web scraped 5 years of daily Wall Street Journal (WSJ) articles from the WSJ archive website and saved them into separate files, where each file contained all headlines and article summaries for a particular day. The web scraping part of this research project was interesting, because we learned that many news websites obscure the text of their articles to prevent easy web scraping.

The last step was to use the trained natural language processing model to classify five years worth of daily WSJ articles as positive or negative and output a csv file with dates, articles, and sentiment as columns.

Ⅲ. Results & Discussion

Both Buy & Sell and Buy & Hold strategies yielded successful results in the sense that they both beat the average cumulative market returns over the five year period (10/24/2013 - 10/26/2018) studied, which is very desirable for investors. In particular, the average cumulative return for market was 20.67%, which the Buy & Sell and Buy & Hold beat by 4.23% and 2.12% respectively.

This result is very successful and could be further improved by upgrading from day trading to a higher frequency trading, which would allow the strategy to act upon the news quicker before all new information is already incorporated in prices.

Buy and Sell Strategy:

Buy and Hold Strategy:

Accuracy of Models

Neural Net

Average 5-Fold Cross Validation Accuracy on Training Set: 83.7%

Average 5-Fold Cross Validation Accuracy on Test Set: 82.8%

The 5-fold Cross Validation above is a special form of cross validation, sequential k-fold cross validation since our data is time series based. The model accuracy over the training set was relatively high at 83.7% demonstrating that our model does a reasonably good job of classifying the price rise changes. The test set 82.8% shows a marginal decrease in comparison to the training set, however, the relative performance for classification is still high.

Example Run - Confusion Matrix:

	precision	**recall**	**f1-score**	**support**
Price Fall (0)	0.78	0.89	0.83	2028
Price Rise (1)	0.89	0.77	0.82	2243

Precision & Recall Averages:

	precision	**recall**
Micro Average	0.83	0.83
Macro Average	0.83	0.83
Weighted Average	0.84	0.83

Precision and recall are relatively high here, indicating that the model is accurately predicting whether the market will rise or fall. The average precision of 83% indicates the percentage of positive (price rise) predictions that were classified correctly. Conversely, the average recall of 83% indicates the number of positive cases that were true positives.

Natural Language Processing

Model accuracy on a positive test dataset: 73.5 %

Model accuracy on a negative test dataset: 32.8 %

The model accuracy for the naive bayes suffered due to a limitation on the number of training examples that we had available to us and were computationally able to train on. The limitation stemmed from our decision to use the nltk library which is not optimized for parallel computations, so it limited the number of movie reviews that were able to be used for training. It could have been improved by using a larger training dataset and by using market sentiment data as opposed to movie reviews to train the model since they are highly biased towards them.

Ⅳ. Conclusion

The aim of this paper was to attempt to use machine learning to predict stock market trends. We have developed two machine learning models to predict whether the next day price of the S&P 500 index will rise or fall. The first model was a neural net based on technical analysis utilizing historical price data and various financial indicators such as the simple moving average, standard deviation, momentum etc. The second model was a natural language processing model that utilized the naive Bayes' classifier to analyze the sentiment of news articles. The output of the first model (price will rise or fall) was then combined with the output of the second model (market sentiment is positive or negative) to form two trading strategies: Buy & Sell and Buy & Hold. On average, both strategies yielded better returns than the market over the studied five year period between 2013 and 2018. Therefore, it can be concluded that machine learning techniques can help to successfully predict the stock market trends.

Ⅴ. References

Boming Huang, Yuxiang Huan, Li Da Xu, Lirong Zheng & Zhuo Zou (2018):
Automated trading systems statistical and machine learning methods and hardware implementation: a survey, Enterprise Information Systems

J.Nithiya Devi and G.Vijayabharathi (2014): An Automated framework for incorporating news into stock trading strategies international journal of research in computer applications and robotics

Philip Treleaven, Michal Galas, and Vidhi Lalchand (2013): Algorithmic Trading Review, communications of the acm, vol . 56 no. 11

Dymitr Ruta (2014): Automated Trading with Machine Learning on Big Data, IEEE International
Congress on Big Data

Andreas C. Muller and Sarah Guido (2017): Introduction to Machine Learning with Python

I. Rish (2001): An empirical study of the naive Bayes classifier, Watson Research Center:

Andrew McCallum and Kamal Nigam (1998): A Comparison of Event Models for Naive Bayes Text Classification

Eugene Fama (1970): Efficient Capital Markets: A Review of Theory and Empirical Work, Journal of Finance