Machine Learning in Intraday Stock Trading

Art Paspanthong
Stanford University
methinp@stanford.edu

Nick Tantivasadakarn
Stanford University
nantanic@stanford.edu

Will Vithayapalert
Stanford University
nopsinth@stanford.edu

Abstract

This paper aims to explore how the predictive power of machine learning models can reap financial benefits for investors who trade based on future price prediction. This project focuses on binary classification problem, predicting the next-minute price movement of SPDR S&P 500 trust and acting upon the insights generated from our models. We implemented multiple machine learning algorithms including: logistic regression, support vector machines (SVM), Long-Short Term memory (LSTM), and Convolutional Neural Networks (CNN) to determine the trading action in the next minute. Using the predicted results from our models to generate the portfolio value over time, support vector machine with polynomial kernel performs the best among all of our models.

1 Introduction

The ability to precisely predict the price movement of stocks is the key to profitability in trading. Many investors spend time actively trading stocks in hope of outperforming the market, colloquially referred to as a passive investment. In light of the increasing availability of financial data, prediction of price movement in the financial market with machine learning has become a topic of interests for both investors and researchers alike.

Insights about price movements from the models could help investors make more educated decisions. In this project, we aim to focus on making short term price movements prediction using the time-series data of stock price, commonly used technical-analysis indicators, and trading volume. Such predictions will then be used to generate short-term trading strategies to capitalize on small price movements in highly liquid stocks.

2 Related work

With the increase of available financial data investors can access to, as suggested by Hegazy, Osman, Soliman, Omar S. and Salam, Mustafa A (2013), machine learning techniques have been applied to create a powerful trading strategy that helps traders more likely make right decision in buying or selling the assets. As mentioned in Chen, Sheng and He, Hongxiang (2018), Neural Network models including both Long Short Term Memory (LSTM) and Convolutional Neural Network can successfully capture the micro-change of time-series data, resulting in accuracy higher than 70% across many different datasets. In addition to deep learning methods, a more traditional model as Support Vector Machine (SVM) is also effective to do the classification work, which predicts whether the price movement will be up or down.

Despite numerous deep learning applications in stock price prediction, only few research focuses on actual profits generated by ML-driven trading. We decided to further explore how the accuracy of predictions from various machine learning models are correlated with the profits that we would obtain based on predicted results.

CS229: Machine Learning, Spring 2019, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

Therefore, the main goal of this paper is not only assessing statistical performance of machine learning in forecasting future price movements but also effectively the evaluating the results in terms of actual profits.

3 Dataset and Features

The dataset is the the SPDR S&P 500 trust (NYSE: SPY) with 1-minute intervals from March 1st until May 24th 2019. The data is available on the IEX Trading website.

The features includes price and trading volume. We also used technical indicators including Simple Moving Average (SMA), Exponential Moving Average (EMA), Crossovers, consecutive price trends with 5, 10, 12, 20, 26, 50, 100, 200 days lookback window). These indicators represent volatility, momentum, and trending strength of price movement. Details of each technical indicator can be found in the appendix.

4 Methods

Code avialable at https://github.com/nantanick/cs229

4.1 Feature Selection

We used Lasso regularization method to select more statistically significant features by shrinking their corresponding coefficients towards zero. Variables selected by Lasso includes:

- Original Features: Volume and Price.
- Simple Moving Averages: SMA5, SMA15, SMA20, SMA200
- Crossovers: SMA5Cross, SMA10Cross, SMA15Cross, SMA20Cros, SMA50Cross, SMA100Cross, SMA200Cross
- Consecutive price trends: Up.Down10, Up.Down15, Up.Down50

4.2 Models

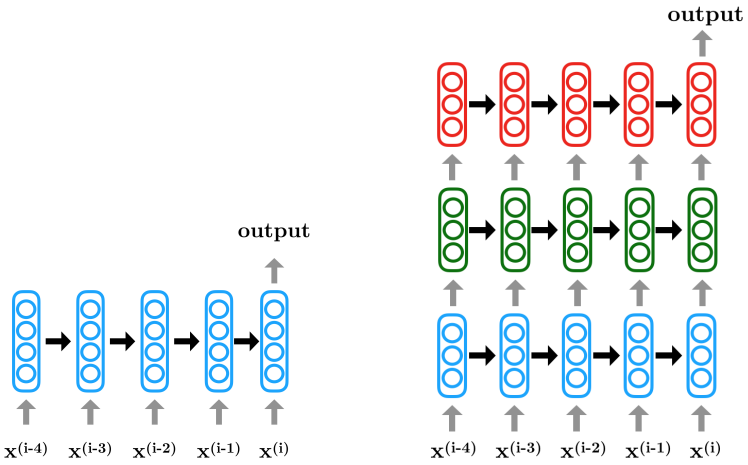
Since the goal of the project is to predict the next-minute price movement (binary classification), we use logistic regression as our baseline model. In order to evaluate how complicated the problem is as well as to understand the structures in the data, we explore the data set with the following models:

1. Baseline Model: We built logistic Regression with and without regularization, as reference to the baseline models. Regularization used includes Ridge and Lasso methods.
2. Support Vector Machine
   For the support vector machine model, we started off by exploring different kernels, including Linear, Polynomial (degree 3), Sigmoid, and Radial Basis Function kernel. After training our simple model, we adjust the model by varying the cost of constraint violation. Specifically, we adjust the "C" part of the optimization problem below.

$$\min_{\mathbf{w},\xi,b} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi^{(i)}$$

$$\text{such that} \quad y^{(i)}\left(w^T x^{(i)} + b\right) \geq 1 - \xi^{(i)}; \quad \forall i \in \{1,\ldots,n\}$$

3. RNN models: We tested Single-layer LSTM, Multi-layer LSTM (Figure 1a), and Multi-layer GRU (Figure 1b) with 128 hidden units in each layer with RELU activation function. We tested multiple lookback windows, and discovered that having a lookback window of 5 works best. Regularization includes early stopping and dropping out parameters.
4. Convolutional Neural Network model: Single layer CNN with a linear layer. The models has access to five data points, to match the RNN model. More complicated CNN models were considered, but was not completed due to time constraints.



(a) Single Layer RNN (b) Multi Layer RNN

Figure 1: RNN models

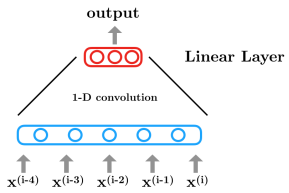


Figure 2: NN model

As our task is a binary classification, we chose sigmoid activation function in the last layer of all Neural Network models (RNN and CNN). Accordingly, our loss function is binary cross-entropy loss.

$$\ell(\theta) = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i \log\left(p_i\right) + \left(1 - y_i\right)\log\left(1 - p_i\right)\right]$$

4.3 Neural Network Hyperparameter Tuning

Once all models were successfully implemented, we also tuned relevant parameters and hyperparameters to improve the model performance by using Grid Search method. Especially for Sequence models that incorporates historical information, we varied different values of lookback window (5, 10, 15, 30 days) and how they affect the statistical performance. Other relevant hyperparameters we tuned include learning rate (across log scale), batch size, number of hidden units, and number of epochs. The results of best hyperparameters are listed as following: Learning rate = 0.0001 Batch Size = 64 Number of hidden layers = 128 and number of epochs: 10 (low due to early stopping)

5 Experiments/Results/Discussion

Model	Accuracy (Training Set)	Accuracy (Test Set)	AUC (Test Set)	Profit/Hour (Test Set)
Baseline (Logistic)	0.4988	0.4899	0.4876	$-0.11
SVM (Linear)	0.5354	0.5341	0.5355	$0.68
SVM (Polynomial)	0.5452	0.5449	0.5449	$1.61
SVM (RBF)	0.5433	0.5384	0.5393	$0.79
SVM (Sigmoid)	0.5024	0.4983	0.5021	$-0.69
GRU	0.5141	0.5096	0.4928	$0.51
LSTM (Single-Layer)	0.5011	0.4983	0.4989	$-0.20
LSTM (Multi-Layer)	0.5127	0.5110	0.4817	$0.47
CNN	0.5130	0.4889	0.5000	$-0.46

Table 1: Statistical Performance and Cumulative Profits of all models.

5.1 Statistical Performance

According to Table 1, the accuracy of training set is generally higher than that of test set, as the models tend to more overfit to the training set. When generalized in the test set, the performance slightly dropped.

From all of the models, the support vector machine with polynomial kernel has the best performance in all metrics. Most of the SVM methods outperform the deep learning methods (LSTM, GRU and CNN) as we expected due to the size of our dataset (approximately 20,000 datapoints). All models except SVM (Sigmoid Kernel), Single-layer LSTM, and CNN models post positive profits.

5.2 Portfolio Performance

The use of model with high statistical performance is not necessarily a successful trading strategy, as the predicted results are only directional, not reflective of actual profits. Therefore, we implemented another algorithms that incorporates directional prediction to generate portfolio value over time. In our case, we would long the asset if the predicted probability is above the upper threshold, short the asset if the probability is below lower threshold, and continue holding our position, if the predicted probability lies in between the two thresholds. The results shown in Table 1 correspond to the 0.48 - 0.50 threshold range (lower - upper: 0.48 - 0.50).

To compute the cumulative profits from our ML-based strategies, we execute trading in our test set under the condition where the principal amount of cash is $1000 with no leverage. Furthermore, in this project, every trade is fully long/short decision, which means that we would spend all cash buying as many shares as possible when we decide to long and short selling as many shares as possible when we decide to short. The results presented above show our profit in zero-transaction cost environment. The result was summarized in the form of how many dollars we can make per hour (as opposed to per minute) so as to illustrate the profit in reasonable unit.

In the real trading world, transaction cost is incurred in every execution order. To make our cumulative profits most realistic, we applied transaction cost in every minute the order was executed. Using transaction cost of 15 basis point, we found out that the transaction costs erode all of our profit, resulting in negative return for all strategies (models). This is primarily because transaction cost associated to a complete process of entering and exiting a position is 30 bps or 0.3%. However, as we are trading within minutes range, short-term price movements within a few minutes rarely goes above 0.3%. This means, even though we make correct predictions, we are still losing money.

5.3 Discussion

Although there are positive correlations between accuracy and profit, from our observations, we found out that higher accuracy does not necessarily translate to higher economic profits. For instance, our GRU model outperforms the multi-layer LSTM in terms of accuracy, but it generates less profit. Our correct predictions could correspond to time with small changes (less profit), whereas some incorrect predictions might correspond to time of large changes (huge loss). At this point, we were not able to control for that. Thus, the accuracy of model predictions does not reflect actual profits.

There are a few limitations to our study. First we simplified our problem to a binary classification, which resulted in low profits as discussed earlier. Second, we only tested our methods on the NYSE: SPY data set, which may not be representative of other stocks in the market. Considering other stocks would allow for a generation of a portfolio that better represent what investors do in real world.

6 Conclusion/Future Work

This paper explored the usage of multiple machine learning models to predict future prices of stocks. We first simplified our problem to a binary classification problem. Then we narrowed down our data, which consists of multiple indicators, to a smaller and more statistically significant subset. Finally we experimented with the different models and found out that SVM with polynomial kernel had the best performance on the dataset we have. Nevertheless, there are multiple limitations to this study, most notably the size of the dataset.

In the future we hope to modify our models to takes into account the magnitude of profit or loss. This includes multi-class classification that accounts for magnitude of price movements or even regression models predicting next-minute price. We also hope to expand our models to incorporate data from other stocks and gather more data to better train our deep learning models. We could also include portfolio optimization, weighing each assets based on predicted probability once we work with different stocks.

7 Contributions

Art Paspanthong: Data Collection, Data Preprocessing, Implemented SVM models, Evaluation Metrics (Trading Execution/Portfolio Generation/Profit Calculation), Report write-up

Will Vithayapalert: Implemented Baseline, RNN models, Hyperparameter Tuning, Report write-up

Nick Tanivasadakarn: Neural Network, Data Processing, Creating framework, Report write-up

References

[1] Chen, Sheng He, Hongxiang. (2018). Stock Prediction Using Convolutional Neural Network. IOP Conference Series: Materials Science and Engineering. 435. 012026. 10.1088/1757-899X/435/1/012026.

[2] Hegazy, Osman, Soliman, Omar S. Salam, Mustafa A (2013) Machine Learning Model for Stock Market Prediction. Faculty of Computers and Informatics, Cairo University, and Higher Technological Institute (H.T.I), 10th of Ramadan City, Egypt

[3] Hasselmo, M.E., Schnell, E. & Barkai, E. (1995) Dynamics of learning and recall at excitatory recurrent synapses and cholinergic modulation in rat hippocampal region CA3. *Journal of Neuroscience* **15**(7):5249-5262.

Appendix

Continuous variables

Trading Volume: This feature reflects how many shares of ETF are traded during the day, which can roughly indicate investor sentiment towards a particular security. If Volume is high, it can reflect high interest in that security.
Moving Average: this indicator measures trendiness of the price, widely used in trend-following strategies to determine both buy and sell signals. Moving Averages can be represented in two measures: Simple Moving Average (SMA) and Exponential Moving Average (EMA), which can be calculated as shown below. We also created SMA and EMA over many different lookback periods (5, 10, 12, 15, 26, 50, 100 days) to observe how long the future price is related to historical prices.

$$SMA_n = \frac{P_t + P_{t-1} + ... + Pt - n}{n} \quad n : \text{lookback window}$$

$$EMA_n = (P_t - EMA_{n-1}) \cdot \frac{2}{n+1} + EMA_{n-1} \quad n : \text{lookback window}$$

Moving Average Convergence Divergence (MACD): It is the difference between EMA-12 and EMA-26.

$$MACD_t = EMA_{12} - EMA_{26}$$

Categorical variables

Crossovers:

1. SMA Crossover indicator variables
2. EMA Crossover indicator variables
3. MACD Crossover indicator variables

The categorical variables were labeled at each timestep as +1 to indicate a crossover with buy signal, 0 to indicate no crossover, and -1 to indicate a crossover with a sell signal. They were calculated as asset price crossovers with all the SMA, EMA, and MACD indicator variables mentioned in the continuous variables section. In traditional trend following strategies, these crossover variables are important indicators of detecting upward or downward trends that can be ridden for profit. Our reasoning for feeding all of them into our models was to allow the algorithm to determine which ones are more accurate predictors of next day returns.

Consecutive price trends: An indicator whether the price have been rising or falling consecutively for the past n days.