Investigating the Effect of Market and News Data on Stock Movement

Chen Xu and Royce Yang*

March 22, 2019

Abstract

While previous literature on the heated topic of stock prediction have demonstrated the significance of sentiment analysis (on news, social media, etc.) in predicting stock movement, few have combined the use of quality sentiment data with market data in trade simulation. Furthermore, with the emergence of machine learning, even fewer studies have taken a step back to consider traditional regression methods, which we prove to be actually more effective than modern machine learning approaches, with an accuracy of nearly 70%. Yet, translating predictions into trading simulations does not Sharpe ratios higher than 1.

1 Introduction

In this paper, our goal is to identify a model that best predicts the binary (up or down) movement of specific collection of stocks during the 6.5 hours in which the New York Stock Exchange and the Nasdaq Stock Market normally operate. We take on three modeling approaches: a logistic regression, SVM, and random forest, and rank the models according to their out-of-sample predictive accuracy. Using the best model among the candidates, which we find to be the logistic model, we then simulate trading procedure using two trading strategies, compute our returns, and discuss the implications.

Thus, our research is broken into two sections:

In the first section, we perform the classification task in which we sequentially expand the training dataset as well as the set of covariates, making an attempt to identify the variables that are key to our models and dropping variables that do not make a significant contribution to the predictive accuracy. We then compare the model performance on the hold-out validation data, choose the model with the highest prediction accuracy, and apply that model onto the

*Xu: Department of Economics, University of Chicago, cx971111@uchicago.edu. Yang: Department of Economics, University of Chicago, royceyang@uchicago.edu. Acknowledgements: Professor Ali Hortaçsu, TA Francisco De Asis Del Villar Oritz Mena. All data and code can be found at https://github.com/cx971111/ECONpaper

test data for the actual performance check. Note that we chose to hold out an additional subset of our data for final testing to prevent over-fitting on the validation set as a result of excessive fine-tuning.

In the second part, we consider two trading strategies based on our model prediction. For the present purpose of this introduction, we suppress the details of the strategies and it is sufficient to know that both strategies trade only when the model predicts, with a given confidence threshold, that a given stock will have a higher closing price than opening price for a given day. Note that the news we use for this prediction is specifically limited to articles published during afterhours, after the previous day's close and before the given day's open time. We do this in order to strictly maintain the fact that the news is published before the movement in stock price. The main difference between our two strategies is that one strategy's daily investment are mutually independent while the other strategy compounds the amount of investment, dependent on the previous day's returns.

The roadmap of this paper is as follows: Section 2 describes our questions of interests and hypotheses and provides the background as well as literature review on the issue of stock prediction. Section 3 focuses on our data and explains the news data, the market data, and the feature engineering as well as the subsetting mechanism. Section 4 explains our methodology in the two project phases (model building and simulation). Section 5 presents and discusses our regression and simulation results. Section 6 tries to explain the anomalies and detail the implications of previous results. Section 7 concludes with summaries of paper contributions, limitations, and future works to be done.

2 Initial Thoughts

2.1 Research Question and Hypothesis

The main goal of this paper is to investigate the use of market and news data on predicting daily stock movement. Upon reviewing previous literature (described later) and upon discussing our topic with professors and peers, we formed three hypotheses:

1. Through training on historical data, the model can predict future stock movement with accuracy as high as 70%.

2. Machine learning models are better at predicting binary stock movements than traditional regression methods.

3. Trading simulations with the best predictive model can yield satisfactory Sharpe ratios, namely greater than 1.

2.2 Background

In general, better understanding the dynamics of the stock market is very relevant to investor decisions. For instance, accurate predictions of stock move-

ment would accrue to them high monetary value. More specifically, as news flood our daily lives and are broadcasted everywhere, investigating their potential impacts on stock movements would be valuable. Yet, competing arguments and studies exist regarding the possibility to do so.

Opposing opinions and groundwork – Efficient Market Hypothesis (Fama, 1991, Fama, Fisher, Jensen, Roll, 1969) & The Inefficacy of Accurate Prediction (Walczak, 2001) claim no more than 50% accurate prediction because of random walk pattern. Despite extensive previous works, there is no definite causal or correlational relationship between stock price and other factors that can aid investor decisions.

Supporting opinions – However, with the advancement of technology and computing power, more detailed and efficient analyses of stock markets claim the contrary: Some tried to predict stock price using only historical prices (Cervell-Royo, Guijarro, Michniuk, 2015). Others added media sentiment analysis to the historical prices model and thereby consistently improved prediction accuracy (Bollen, Mao, & Zeng, 2011).

Therefore, with a good set of longitudinal data across a long time, we aim to further investigate the possibility of stock prediction using available market or news, with the ambition to aid investor decisions by conducting simulations based on model performance.

2.3 Literature Review

Based on our approach to the problem, we evaluate past studies using logistic regression and machine learning models to predict stock return. We also evaluate works that intend to assess the impact of non-financial information (such as social media comments) on stock price. Overall, we believe our study has several advantages:

- Comparing to studies that aim to pick a good predictive model via model selection, our data are more expansive as we consider not only financial data but also news data that are often ignore by these studies.
- Comparing to studies that incorporate non-financial data, their data often come from social media contexts saturated with anecdotes or rumors that appeal more to investor emotion rather than to rationality. In comparison, our data sources are more reliable and influential, as these news are reported by big news providers that screen the contents carefully before releasing them to the public.
- Furthermore, while some of these papers end with a proposed model and performance, our study attempts to understand the model performance in greater detail. As we simulate trading strategies that are routinely used by investors, we could better understand the monetary value of these models, which motivates this paper in the first place.

On using social media and market data and machine learning models: Nguyen, Shirai, & Velcin (2015) integrates the sentiments in social media for the prediction of stock price movement. Using JST-based and Aspect-based sentiment analysis, its proposed SVM model with a linear kernel can predict the stock price movement with more than 60% accuracy for a few stocks, and performs much better than other methods for the stocks that are difficult to predict with only past prices.

On using logistic models with market data: Hussain et al. (2018) uses logistic regression model to predict the performance by accounting and financial variables of the nonfinancial firms listed in Pakistan stock exchange. Specifically, sales growth (percentage change in sales), return on equity, debt to equity, current ratio, earning per share and book to price ratio were used. The model results reaches 88.7 percent level of accuracy.

On using sentiment scores of news stories to predict stock returns through cross-sectional regression: Heston & Sinha (2016) makes use of cross-sectional regression to discover the news attention effect, that firms with neutral news outperform firms without any news, and that negative sentiment has longer-lasting impact than positive sentiment.

On algorithmic trading with sentiment analysis and reinforcement learning: Kaur (2017) formulates a Markov Decision process that is solved using a reinforcement learning algorithm. He demonstrates that by looking at sentiment in the headline of news articles alone, machine learning algorithmic trading methods could achieve a Sharpe ratio of 0.85.

3 Data

One of the key distinguishers of our study from related works is the quality of our dataset. We make use of sentiment analysis on news data, provided by Thomson Reuters, that ranges from 2007 to 2016 in line with general market data, provided by Intrinio, spanning the same time frame.

3.1 News Data

Our news data is provided by Thomson Reuters News Analytics (TRNA), and the key variables we make use of include noveltyCount, volumeCount, firstMentionSentence, relevance, and several sentiment score indicators. Our news data is a diverse collection of news articles of many sources, topics, and companies, in the period of 2007 to 2016. We have a total of 9,328,827 observations in this dataset.

Note that although we are directly using data provided for us by Thomson Reuters, we have gone through the process of reproducing this data ourselves in order to better understand the dataset. Namely, we have conducted sentiment analysis[1] on the headlines of the news articles and found a correlation of 0.93

[1]We use VADER (Valence Aware Dictionary and sEntiment Reasoner) to conduct sentiment analysis in Python.

with the scores given by Thomson Reuters.

noveltyCount is a way of discerning whether a news article is breaking news. It determines the uniqueness of a news article within a configurable history period, such as 24 hours. This is made possible through TRNAs linking system that, when given articles mentioning the same asset, determines the similarity distance between the two articles and thereby decide whether or not to classify the pair as linked articles. In other words, two articles are linked if they are similar enough. A low noveltyCount of 0 indicates that there are 0 linked news articles and that the one at hand is breaking news.

volumeCount simply counts the number of news articles that mention the same asset, within a configurable time period, such as 24 hours.

firstMentionSentence provides the index of the first sentence, counting from the headline as 1, in which the scored asset is mentioned. A value of 0 indicates that the name of the asset is not mentioned in the news headline or body text. A value of 1 indicates mention in the headline, a value of 2 indicates mention in the first sentence of the body, a value of 3 indicates mention in the second sentence of the body, etc.

relevance is used to discern whether the company is the focus of the article, one of many mentioned in the article, or just a passing mention in the article. The relevance score compares the number of times the company is mentioned in the article, compared to the number of times that other companies are mentioned. A relevance score close to 1.0 denotes that a company is the focus, a relevance score of 0.2 to 0.8 denotes that the company is one of several mentioned substantively, and a relevance score of 0 to 0.2 denotes that the company is simply a passing mention of the article.

sentiment is the most complicated variable in our dataset. Our measure of sentiment only reflects the authors judgement of the company independent of market sentiment. TRNA uses a three step process for sentiment analysis: preprocessing, feature extraction, and classification. In preprocessing, sentences are properly divided and formatted into meaningful lexical tokens. In feature extraction, words are classified as one of: adjective, adverb, intensifier, noun, or verb and complex structures such as negation and intensification are simplified. In classification, the attribute of positive, negative, or neutral is assigned to the article based on the number of positive and negative features.

3.2 Market Data

Our market data is provided by Intrinio, and our variables of interest are open and close price, daily trade volume, and market-residualized returns from the previous day. We have panel data (collected every day) from 2007 to 2016 of US-listed instruments. This means that we do not have data on foreign stocks. Each of our variables are specific to a single asset (stock):

open and *close* represent the open price and close price for the day, respectively.

volume is the trading volume, in shares, for the day.

returnsClosePrevMktres1 is the return of the previous day, calculated close-to-close and market-residualized. Market residualized means that we account for the general movement of the market leaving only movements inherent to the instrument. More specifically, this variable is defined as the asset's excess return minus beta times the benchmark excess return (i.e. S&P 500). Despite having no explicit criteria based on which $Mkres$ is calculated, some investigations conclude that it is subtraction of a per-stock beta adjusted market return from the raw return.

3.3 Feature Engineering

We expand the dataset by creating two new variables that are key to our regression:

The first variable is *marketcap*, which we define to be the product of volume and closing price.

The second variable is *up*, a binary variable that is 1 if the closing price is greater than the opening price and 0 otherwise.

3.4 Subsetting

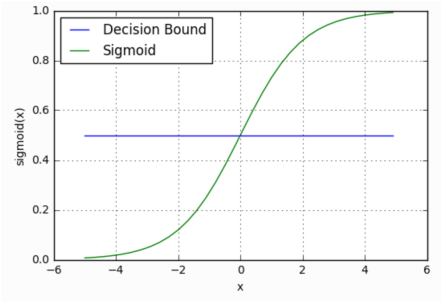
We have a lot of data, but it may be more efficient and reasonable to only use a subset. Specifically, we intend to eliminate the effect of the 2008 financial crisis on our stock returns so that we tried to use data from only 2012 to 2016. Although some may argue that the effects of the Great Recession had dissipated as early as 2010, we chose to start from 2012 just to be safe, and because we have sufficient number of observations in our data to do so. Also, we chose companies primarily based on their market capitalization and the number of post-2012 positive/negative news targeted at their stock. This is because we wanted to make sure the stocks for our simulation are liquid (so our investment would not affect its price) and that they have sufficient relevant observations for a reasonable standard error.

Specifically, our data have 15 companies in total, with companies with the top 5 market cap, as high market cap stocks make up 91% of the US equity market and therefore are often the key players in portfolio investments, as well as the top 5 companies with the most negative / positive sentiments (so in total of 10). Taken overlapping into consideration, we consider the company with a lower ranking (e.g. top 6th) according to our criteria if a company belongs to the top 5 in multiple categories.

4 Methodology

As described earlier, this paper is separated into two phases, the first being model selection based on prediction accuracy and the second being simulation upon knowing the predictive results. We briefly describe the methods we adopted in each phase below.

Figure 1: Logistic Prediction



4.1 Prediction

Since we chose to use binary response variables, this regression task is equivalent to classification in machine learning, hence solvable by logistic regression as a classic example of generalized linear model and by Support Vector Machine (SVM) and Regression Trees (Random Forest) as more advanced machine learning techniques. We briefly describe and formulate the models here. All the technical details, including parameter estimation procedure, prediction, evaluating losses, etc. can be found in the *mathematical appendix* in the end.

4.1.1 Logistic Regression

Graphically, the decision boundary and sigmoid function of the logistic function look like:

Overall, logistic regression is one of the effective linear methods for classification via setting up a decision boundary. More specifically, it arises from the desire to model the posterior probabilities of the K classes via linear functions in x, while at the same time ensuring that they sum to one and remain in [0, 1].

4.1.2 SVM

As a generalization of linear decision boundaries for classification, the support vector machine produces nonlinear boundaries by constructing a linear boundary in a large, transformed version of the feature space.

Regarding formulation, it looks like:

$$\min_{\|\beta\|} s.t. \begin{cases} y_i \left(x_i^T \beta + \beta_0\right) \geq 1\xi_i, \forall i \\ \xi_i \geq 0, \sum \xi_i \leq constant \end{cases}$$

where $\sum \xi_j^*$ is the total distance of points on the wrong side of their margin and β are parameters to be estimated.

4.1.3 Regression Tree

Random forests (Breiman, 2001) are an ensemble learning method for classification that builds a large collection of de-correlated decision trees, and then averages them. On many problems the performance of random forests is very similar to boosting, and they are simpler to train and tune. As a consequence, random forests are popular, and are implemented in a variety of packages.

The idea in random forests algorithm is to improve the variance reduction of bootstrapping by reducing the correlation between the trees, without increasing the variance too much. This is achieved in the tree-growing process through random selection of the input variables.

We also describe the sequence of variables used in our model prediction here. For simplicity, we did not consider higher-order basis vector such as the smoothing spline, wavelet, etc, that may make the prediction better but are also prone to overfitting and opaque interpretations.

We resorted to previous literature on variable selection. Motivated by section 4.1 in *Thien, 2015*, we decided to expand and shrink the size of our predictor in an iterative fashion, with the model that only include *close* as the baseline model. Specifically, the sequence of choices we make was:

First, close price as the only market-specific predictor.

Then, include $returnsClosePrevMkres1$ as another market-specific variable.

Then, include all the news-specific variables, such as $sentimentscore$ of news (as a continuous variable), $noveltycount, volumecout, firstMentionedSentence$, etc.

Lastly, exclude all market-specific variables to assess the pure predictive power of news variables.

4.2 Simulation

In this second phase, once we have decided upon a model based on its prediction on the hold-out 3-month test data, we simulated trades based on two simple strategies.

For succinctness, we present in this section only the two mechanisms of the strategies. We will present trading results, explain our rationale behind trading based on these strategies, the pros and cons for each strategy, and compare the trading results in the next section.

4.2.1 Strategy 1

Buy and sell 1 share only when model predicts 1 and do nothing when it predicts 0. Upon using the validation data with decision boundary 0.5, 0.65, and 0.8, pick the best decision boundary among these. Then, trade on the test data with the optimal decision boundary. Output the threshold, daily average return rate, standard deviations on daily returns, and Sharpe ratio for each simulation.

4.2.2 Strategy 2

Unlike strategy 1 that makes independent strategies (in terms of daily investment budget), strategy 2 starts with a fixed amount of money on day 1, and on day i, invests all of its current budget (which equals to budget on day $i-1$ + net gain on day $i-1$) into the companies whose stocks are predicted by the model to rise on that day. The proportion of investment into each stock on each day depends on the prediction results favorable to that company on that given day (in terms of the number of up movements predicted for stock k on day i). Output the threshold, daily average return rate, standard deviations on daily returns, and Sharpe ratio for each simulation.

5 Results and Discussion

5.1 Prediction

5.1.1 Logistic

Upon merging the data 5 market cap, 5 negative market and 5 positive market (in total 15 companies) and using the sequence of covariates described in the earlier section, We found that the model with all the news covariates and all the market covariates perform on par with the model with only market covariates. We decided to use the larger model (model with column number 3 in the table below) since the estimates are all significant. This model predicts with 65% accuracy on the validation data and 69.8% on the test data, when the decision boundary is set to be 0.5.

Upon analyzing the table, which is shown on the next page, we can see that both market-specific predictors and several news-specific predictors are significant. However, the magnitudes of the estimates, except that of $returnClosePrev Mktres1$, are uniformly small, making it unlikely to affect the probability of prediction much.

The meaning of estimates relates to odds between probabilities of predicting up and predicting down. For example, the estimate of $returnClosePrevMktres1$ at 39 means that $P(up)/P(down) = e^{39}$ when $returnClosePrevMktres1$ increases by 1, which almost guarantees that the model predicts 1, since the odds are so high.

Furthermore, many predictors are significant under $\alpha = 0.05$ but do not affect predictive accuracy much (upon viewing Figure 2 which plots confusion matrices for model 2 and 3). Noticeably, the variable $returnClosePrevMktres1$ has the most significant impact on the predictive probability. However, we lack the complete understanding of what it is, although we concluded before that it is the subtraction of a per-stock beta adjusted market return from the raw return.

	Dependent variable:			
	Price	P+Market	P+M+News	News
	(1)	(2)	(3)	(4)
close	-0.0001**	-0.0002***	-0.0002***	
	(0.00004)	(0.00004)	(0.00004)	
returnsClosePrevMktres1		34.766***	34.951***	
		(0.405)	(0.405)	
sentimentNegative			0.140***	0.054**
			(0.025)	(0.024)
sentimentPositive			0.012	0.039
			(0.029)	(0.028)
relevance			0.045***	0.055***
			(0.017)	(0.016)
noveltyCount24H			-0.003***	-0.004***
			(0.001)	(0.001)
volumeCounts24H			0.002***	0.001***
			(0.0002)	(0.0002)
firstMentionSentence			-0.0001	0.0001
			(0.0005)	(0.0005)
Constant	0.027***	0.037***	-0.074***	-0.068***
	(0.007)	(0.007)	(0.022)	(0.021)
Observations	146,823	146,823	146,823	146,823
Log Likelihood	-101,761.200	-96,675.920	-96,603.490	-101,729.700
Akaike Inf. Crit.	203,526.400	193,357.800	193,225.000	203,473.300

Note: *p<0.1; **p<0.05; ***p<0.01

Also, we present two plots that show the predictive results of the four models above on the validation data, where the first plot is for bear stocks (those with close price < open price) and the second is for bull stocks

The plot on the right shows the predictive across across four models when the actual response is up. We can see that model predicts upward trends with more uniform accuracy than it does for downward trends. This should be helpful to investors who want accurate predictions of up more than of downs.

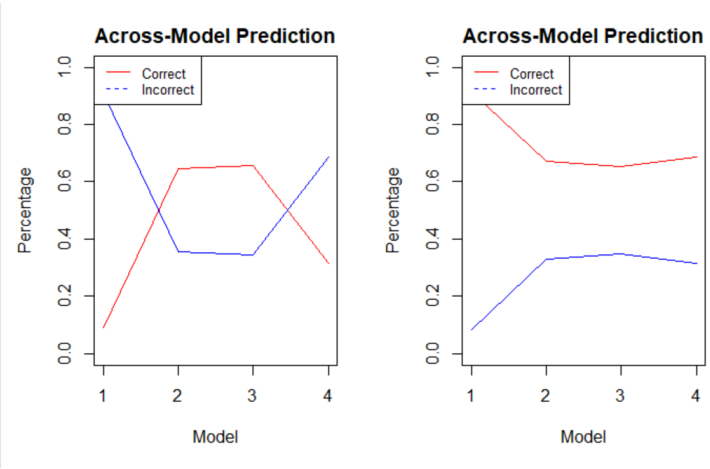


Figure 2: Plotting of Predictive Accuracy Across Predictor Combinations

5.1.2 SVM and Regression Tree

For the machine learning models, it is hard to get a good regression table as above, so we decided to just report the overall predictive accuracy onto the validation data, with the set of covariates we decided to use. In particular, we used the covariates in the three model including news and market data variables to train the models. The decision boundaries are set to be 0.5 in both models. The figure on the left is the prediction using SVM and the one on the right is the prediction using Random Forest

```
Accuracy : 0.6819
  95% CI : (0.6705, 0.6931)
```

```
Accuracy : 0.6823
  95% CI : (0.6709, 0.6936)
```

Indeed, all three models seems comparable in terms of their predictive accuracy of the stock movement upon predicting on test data. In addition to testing the simple accuracy by matching with the true outcomes, we also compare the confusion matrices at given decision boundaries, focusing on the accuracy of the "predicted to rise" column, since in our simulation we do nothing when we are not confident enough that the stock price will rise. Lastly, we make a comparison of the R^2 values when regressing the predicted outcomes on the true outcomes. There are no clear advantages in using machine learning models that can be harder to interpret and require significantly more computational power. Thus, for our simulations in part 2, we decide to use the logistic model since it is more easily understood.

5.2 Simulation

5.2.1 Strategy 1

Upon deciding to use the logistic model, we apply it first onto the validation dataset with different thresholds to decide on a threshold that generated the best Sharpe ratio and rate of return. In particular, we converted the 3-month risk free rate, which was at 3.5%, into daily risk free rate by the formula $r_1 = (1 + r_{90})^{1/90} - 1$. As a result, the optimal threshold is 65%. We show the table of daily returns and volatility for several different threshold levels below:

threshold	daily.rate.of.return	daily.volatility	Sharpe.ratio	Val_Test
0.500	0.005	0.013	0.334	Val
0.650	0.012	0.021	0.576	Val
0.800	0.013	0.028	0.442	Val
0.650	0.019	0.024	0.767	Test

Notice that the daily volatility and rate of return simultaneously increase when the threshold increases. The Sharpe ratios fluctuate a lot but are all within a moderate range. However, there emerges the problem of unlimited total investment, since the strategy assumes no capping on total money available. This problem is resolved by strategy 2.

5.2.2 Strategy 2

The table below summarizes the results when we start with a fixed amount of $ 10000 on day 1 to invest rather than with an unlimited budget. We decide upon a threshold using the same approach as in strategy 1.

threshold	daily.rate.of.return	daily.volatility	Sharpe.ratio	Val_Test
0.500	0.005	0.010	0.497	Val
0.650	0.016	0.019	0.817	Val
0.800	0.024	0.030	0.799	Val
0.650	0.019	0.021	0.874	Test

Fortunately, strategy 2 generates a slightly higher Sharpe ratio on the test data at a threshold of 65%. The overall average daily rates of returns are comparable between two strategies but the second strategy reduces volatilities at several thresholds.

Overall, we believe that the Sharpe ratios between two models are not very promising because accurate predictions of binary stock movements cannot be directly converted into very promising returns, which involve continuous close prices and open prices that vary a lot by stock. In other words, the binary response may not capture the variability across stocks over days, which is reflected in the standard deviations of the daily returns. We further justify and explain the implications of our returns results in the section below.

5.3 Relate to our initial hypotheses

1. Machine learning methods do not yield significantly better results in terms of predictive accuracy. In fact, they even predict with 1-2% less accuracy than a simple logistic regression. Yet, we are currently unable to explain why these models are less accurate than the logistic model.

2. Predictive accuracy is as high as 69.8%. Although its exactness may be variable upon the training/validation split and the set of covariates we choose, we can be assured that our model makes decently good prediction, especially when large amounts of company data across years would typically introduce many confounding effects and interactions between stocks that make predictions harder.

3. The Sharpe ratios are reasonable but not satisfactory (still less than our hypothesized ratio of at least 1.0), indicating that our predicted results for stock movements are not yet suitable for generating promising returns.

6 Justification and Implication

In this section, we attempt to justify why our accurate prediction does not yield satisfactory Sharpe ratio under the two strategies we considered.

It is important to notice that to simplify the task involved in stock prediction, we decided to work with binary response as our response variable during the first phase of our project. Specifically, stocks with different unobserved volatilities and prices are considered the same in terms of the response value so long as their price differences have the same sign (both negative or positive).

However, during the actual simulation procedure, the calculation of Sharpe ratio requires knowing standard deviation of return and excess return, both of which require considering the exact magnitude of price change and how variable this magnitude could be. Doing so thus introduces complications that are beyond the capabilities of our model.

Moreover, even when different investors hold the same set of information (i.e. prediction of stock movement), developing trading strategies that maximize rate of return and minimize the standard deviation of returns involves careful and often sophisticated knowledge and practical experience. It is thus very likely that our simple strategies above were far from being optimal.

In summary, we want to acknowledge that under certain simplifications, such as predicting only price trend but not magnitude, our model predicts well and can be used as a benchmark for gaining insights into the relationship between some market-specific and news-specific predictors. Yet, it must be extended and amended if used in reality to generate profit. Noticeably, the discrepancy between good model prediction and poor return statistics further indicates that investors in reality need to be careful about depending their tradings on the superficial claims or insider information about the ups and downs of stock prices,

as more nuanced variables such as volatility and trading mentality are not easily predicted or characterized by them.

7 Conclusion

7.1 Contribution and Value

By taking advantage of the large data set, we were able to discover significance of news-specific data in predicting the stock movement. By comparing machine learning models against the logistic model, we were able to quantify the predictive performance and based on the model predictions, simulate two trading strategies that yield unpromising but explainable results. Thus, our project is distinguished from other empirical paper on stock prediction as we not only build a satisfactory model but also extend its results to actual trading.

7.2 Limitations

We also acknowledge several limitations of our current work

- Cannot explain why return are very good but standard deviations are very high.
- Cannot provide causal inference (i.e. what caused changes in stock price).
- Did not test the robustness of the model (i.e. predict model on companies whose market and news data never appeared in our training set.)

7.3 Future Work

Regarding Prediction – we want to consider Ensemble Learning[2] of different models whose overall predictive performance can exceed individual ones. Furthermore, we want to, in greater detail, analyze headline sentiments to have better control over sentiment calculations. Lastly, we would want to incorporate more recent stock data and out-of-sample stock data.

Regarding Simulation – we want to implement other trading strategies that deal with time-series data and maximize profit but minimizing volatility at once. Furthermore, we want to implement and conduct a real-time simulation, and try to make real money!

References

Adnan Hussain, Irfan Lal, Muhammad Mubin, and Shahan Syed. 2018. "Prediction of stock performance by using logistic regression model: evidence from Pakistan Stock Exchange." *Asian Journal of Emperical Research.*

[2]We want to use multiple learning algorithms to obtain better predictive performance than any individual model could obtain.

Bollen J., Mao H., and Zeng X. 2011. "Twitter mood predicts the stock market." *Journal of Computational Science.*

B., Qian, and Rasheed K. 2007. "Stock market prediction with multiple classifiers." *Applied Intelligence.*

Cervell-Royo R., Guijarro F., and Michniuk K. 2015. "Stock market trading rule based on pattern recognition and technical analysis: Forecasting the DJIA index with intraday data." *Expert Systems with Applications.*

Eugene F. Fama, Lawrence Fisher, Michael C. Jensen, and Richard Roll. 1969. "The Adjustment of Stock Prices to New Information." *International Economic Review.*

Fama, Eugene F. 1991. "Efficient Capital Markets." *The Journal of Finance.*

Heston, Steven L., and Nitish R. Sinha. 2016. "News versus Sentiment: Predicting Stock Returns from News Stories." *Finance and Economics Discussion Series.*

Kaur, Simerjot. n.d.. "Algorithmic Trading using Sentiment Analysis and Reinforcement Learning."

Rechenthin M., Street W.N., Srinivasan P. 2013. "Stock chatter: using stock sentiment to predict price direction." *Algorithmic Finance,.*

S., Walczak. 2001. "An empirical analysis of data requirements for financial forecasting with neural networks." *Journal of Management Information Systems.*

Thien Hai Nguyen, Kiyoaki Shirai, and Julien Velcin. 2015. *Sentiment analysis on social media for stock movement prediction.*

Vu T.T., Chang S., Ha Q.T. Collier N. 2012. "An experiment in integrating sentiment features for tech stock prediction in Twitter." *24th international conference on computational linguistics.*

A Mathematical Appendix

A.1 Logistic

- Mathematical Formula

  With two classes, 1 and 0, the predicted probability satisfies:

  $P(class = 1) = S(z) = \frac{1}{1+e^{-z}}$

  $\cdot s(z) =$ output between 0 and 1(probability estimate)

  $\cdot z =$ input to the function (usually $z = x^t\beta$ as the linear predictor)

  $\cdot e =$ base of natural log

- Objective Function

  Instead of Mean Squared Error, we use a cost function called Cross-Entropy, also known as Log Loss. Cross-entropy loss can be divided into two separate cost functions: one for $y = 1$ and one for $y = 0$.

  $J(\theta) = \frac{1}{m} \sum_{i=1}^{m} cost\left(h_\theta\left(x^{(i)}\right), y^{(i)}\right)$
  $cost\left(h_\theta(x), y\right) = -\log\left(h_\theta(x)\right) \qquad if y = 1$
  $cost\left(h_\theta(x), y\right) = -\log\left(1 - h_\theta(x)\right) \qquad if y = 0$

  If compressed into 1, the loss function is:

  $J(\theta) = -\frac{1}{m} \sum_{i=1}^{m} \left[y^{(i)} \log\left(h_\theta\left(x^{(i)}\right)\right) + \left(1 - y^{(i)}\right) \log\left(1 - h_\theta\left(x^{(i)}\right)\right)\right]$

  In vectorized form, this looks like:

  $h = g(X\theta)$

  $J(\theta) = \frac{1}{m} \cdot \left(-y^T \log(h) - (1 - y)^T \log(1 - h)\right)$

- Parameter Estimation

  To find parameters β that minimize our cost, we use Gradient Descent. The iteration procedure in Gradient Descent looks like:

  $\mathbf{a}_{n+1} = \mathbf{a}_n - \gamma \nabla F\left(\mathbf{a}_n\right)$

  where a multi-variable function $F(\mathbf{x})$ is defined and differentiable in a neighborhood of a point $\mathbf{a}$

  In particular, the gradient of the logistic function is very easy to calculated, as given here:

  $s'(z) = s(z)(1 - s(z))$

  Which leads to an equally beautiful and convenient cost function derivative:

  $C' = x(s(z) - y)$

- Prediction

  Generally, with a decision boundary neutral to both 1 and 0, the prediction, where $p = S(z)$ for the sigmoid function defined above, satisfies:

  $p \geq 0.5$, class $= 1$

  $p < 0.5$, class $= 0$

A.2 SVM

- Mathematical Formula

  The use of SVM relates to the notion of margin, which is distance of the closest instance point to the linear hyperplane. This formulation is nice since noises in the data affect margin little if margin is large.

  Thus, to find a maximum margin classifier, we aim to find parameters $\hat{w}, \hat{w}_0$ such that

$\hat{w}, \hat{w}_0 = w \in R^d, w_0 argmax \min_i \frac{y^{(i)}\left(w \cdot x^{(i)} + w_0\right)}{\|w\|}$

Equivalent, we are solving the problem:

$\widetilde{w}, \widetilde{w}_0 = wargmin \|w\|^2$ s.t. $\forall i, \quad y^{(i)}\left(w \cdot x^{(i)} + w_0\right) \geq 1$

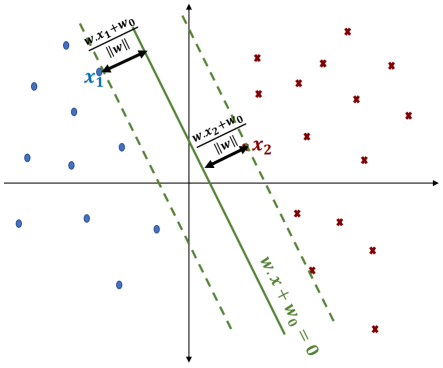


Figure 3: SVM Linear Classifier

- Objective Function
  Similar to the logistic regression, the surrogate loss looks like:
  $\ell^{01}(f(x), y) = 1[sign(f(x)) \neq y]$
  $= 1[sign(f(x)y) < 0]$.
  In other words, we are effectively calculating the number of cases whose predictions differ from the actual observation.

- Parameter Estimation
  Given the formulation above, which states
  $\hat{w} = wargmin\|w\|^2 \quad$ s.t., $\quad y^{(i)} w \cdot x^{(i)} \geq 1 \quad \forall i,$
  the following theorem holds. The proof is not given here.
  Representer Theorem: $\exists \{\beta_i\}$ such that $\widehat{w} = \sum_i \beta_i x^{(i)}$
  $\{\beta_i\}$ also satisfies $\beta_i = 0$ for all i such that $y^{(i)} \widehat{w}.x^{(i)} > 1$

- Prediction
  Prediction is determined by computing the distance of the new linear predictor using estimated parameters and compare it with the margin, classifying it accordingly.

A.3 Regression Tree

The following is adopted from Chapter 15 of Elements of Statistical Learning:

1. For $b = 1$ to B :
   (a) Draw a bootstrap sample $\mathbf{Z}^*$ of size N from the training data.
   (b) Grow a random-forest tree T_b to the bootstrapped data, by re- cursively repeating the following steps for each terminal node of the tree, until the minimum node size n_{min} is reached.
      i. Select m variables at random from the p variables.

ii. Pick the best variable/split-point among the m.

iii. Split the node into two daughter nodes.

2. Output the ensemble of trees $\{T_b\}_1^B$

To classify a new point x:

Let $\hat{C}_b(x)$ be the class prediction of the b^{th} random-forest tree. Then $\hat{C}_{\text{rf}}^B(x)$ =majority vote $\left\{\hat{C}_b(x)\right\}_1^B$