Taking Stock with Social Media

An Integration of Financial and Social Data Analysis

Eliott Chapuis

Final Report

CS 6365 Spring 2017

Contents

Motivation

Over the past decade, social media has grown tremendously, becoming a hub for a variety of content as well as a major medium for public discourse. Companies such as Facebook, Twitter, and Google Plus boast millions to billions of monthly active users, with terabytes of content being generated each second. Given the huge scale of data being created in real-time, much of the information hidden within these networks is largely untapped, and the potential to glean meaningful insights from this data grows as these services become more and more ubiquitous. This project aims to tap into some of these insights within social data in order to better predict and understand market sentiment and how it relates to the performance of financial instruments.

The purpose of this project is twofold: to gain a better understanding of the social data currently available through public APIs and to apply this understanding towards financial insight. From the academic perspective, the scale of the data alone presents a tremendous challenge. This challenge will provide a significant learning opportunity, specifically in terms of data preprocessing, visualization, algorithmic efficiency, and data storage. Additionally, the development of large amounts of software using various tools and languages – ranging from statistical processing languages to web-based data collection tools – will provide a breadth of meaningful learning experiences.

Secondly, and perhaps more importantly, the application of social data towards the prediction of financial instruments may not only aid in the evaluation of market sentiment but also provide support for corporate investment in social media presence. In this way, one of the primary goals of this project will be to evaluate the relevance of social data (specifically Twitter) with regards to the financial performance of highly liquid stocks. It is strongly believed that, given the proper data preprocessing and quantitative evaluation of social data, an effective financial model could be used to predict stock prices with a certain level of accuracy.

Related Work

There has been a variety of work performed trying to link social data to the prediction of prices for financial instruments, notably using data from Twitter. The reason for this is the fact that Twitter is incredibly open with its data and offers fairly lenient rate limits to access its API. However, the complexity of this task – ranging from the automated gathering of data to natural language processing (NLP) of posts – is certainly not trivial. Therefore, I have split my research into three categories: data mining and processing, sentiment analysis/NLP methods, and application-based social media analysis. The focus will be on gathering and understanding meaningful social media data; there will be limited financial analysis, although simple quantitative measures will be used as benchmarks [17].

Data Mining and Processing

The core objectives of data mining in the context of this problem will be to gather and process large amounts of relevant social media data. This involves filtering out non-relevant material, understanding the context and meaning of large amounts of qualitative data, and processing the data in a way that allows for effective visualization [2,3]. The rise of social media introduced a huge new source of real-time data and new areas of research, including work on group detection, influence propagation, and behavior analysis [2]. In particular, many of these insights are derived from methods such as sentiment analysis, which involves extracting high-level emotions from text. Another significant portion of data mining involves extracting sentiments from the social media data [10].

Sentiment Analysis/NLP Methods

One of the primary tasks of this project will be to extract and analyze sentiment behind social media data in order to use it to predict market sentiment. Natural language processing (NLP) is the broader definition encompassing sentiment analysis, applying machine learning techniques such as naïve Bayes, maximum entropy classification, and support vector machines to text analysis [11]. Other NLP techniques that have been studied include Lexicon-based methods that gauge sentiment based on pre-classified words [13] and sentence architecture models that analyze sentence structure and the sentiments of key words such as the subject and verb [8]. Finally, more advanced methods capture greater meaning from words using vector embeddings which are trained using trained neural networks [4].

Application-Based Social Media Analysis

Given the tremendous range of information being shared over social media, there are naturally many applications that benefit from social media insights. Here are some examples of some of the existing literature:

“Stock Prediction Using Twitter Sentiment Analysis” [9]

This is one of the cornerstone papers introducing the use of social data in predicting stock market variations over time and has gained significant attention in the media. The primary premise of the paper is to explore the predictive amplitude of public mood on economic indicators. Overall, the paper was significant in that the researchers were able to make significant predictions on aggregate stock (i.e. the market) movement.

“The Predictive Power of Online Chatter” [5]

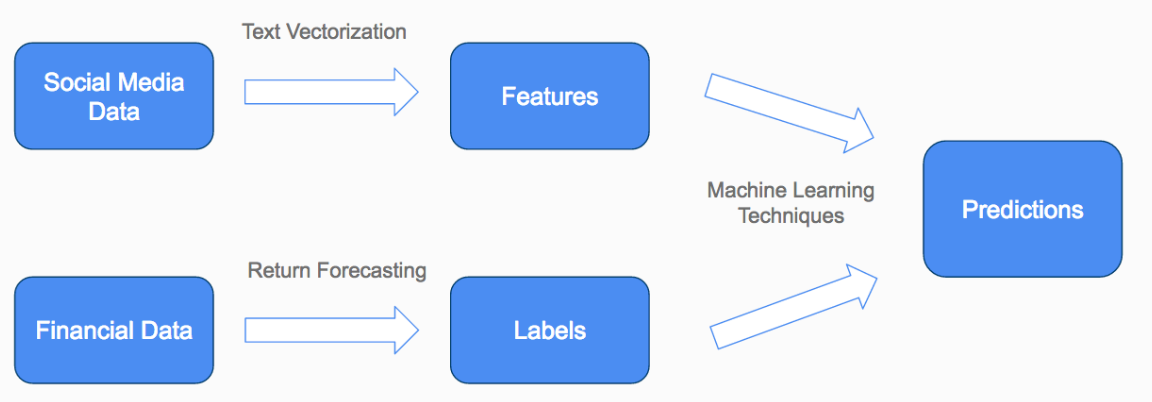
The notion that social media has become one of the cornerstones of public information is discussed in this paper, and the authors use millions of ranked reviews of 2,340 books to predict spikes in sales rank. The paper analyzed the frequency of blog, media, and web page mentions to understand the effect on overall sales.

“Predicting the Future with Social Media” [1]

This paper introduces yet another application for social media data: forecasting box-office revenues for movies. While this topic seems somewhat offhand from the financial stock analysis presented in this paper, it has significant implications for the research being pursued. Firstly, this paper demonstrates the predictive ability of social media. Second, it demonstrates that there are financial corporate benefits to catering social presence to the public, two of the major goals of this project.

Project Scope

The primary purpose of this project is to demonstrate how quantitative analysis of text can be used to generate predictions using stock data. In short, the procedure that is followed is best summarized in the following flow chart:



In other words, in this project, I have gathered social media data in an attempt to demonstrate its usefulness in predicting fluctuations in stock prices. To do this, I used a variety of text processing techniques to refactor the social media data into feature vectors (i.e. numerical representations). Then, I used some basic forecasting techniques to transform financial data into labels that could be used in a machine learning model. Finally, combining these features and labels, I was able to algorithmically discover the relationship between the features (social media data) and the labels (financial data), allowing me to make predictions based on new instances of social media data.

In this way, my project proposes to make the following contributions:

- Provide a detailed description of effective text processing and vectorization techniques
- Introduce basic financial modeling concepts
- Demonstrate the viability of using social media data to predict stock prices

While individually these contributions may not directly add to the current state of the art, together they provide a good introduction to financial forecasting and sentiment analysis while conveying several interesting insights.

Implementation

Data preprocessing, word2vec, kmeans clustering, random forest classifier

Data Sources and Scope

For my project, the time period that was observed was from 2010 to 2013, with the 2010-2012 time frame used for training the models and the 2012-2013 time frame used for testing. The social media data was gathered from Twitter and Reddit. Specifically, only the tweets using #financialnews, #financialtimes, #nasdaq, #nyse, and #sp500 were used for gathering Twitter data, and subreddits r/finance, r/investing, r/market, and r/stocks were used for gathering comments from Reddit. These hashtags and subreddits were selectively chosen based on their relevancy to the context of this project: predicting stock prices. Additionally, the financial data was gathered using Yahoo! Finance. Daily closing prices for stock indices NASDAQ, Dow Jones Industrial, and the S&P 500 were used to generate the feature labels used for classification tasks.

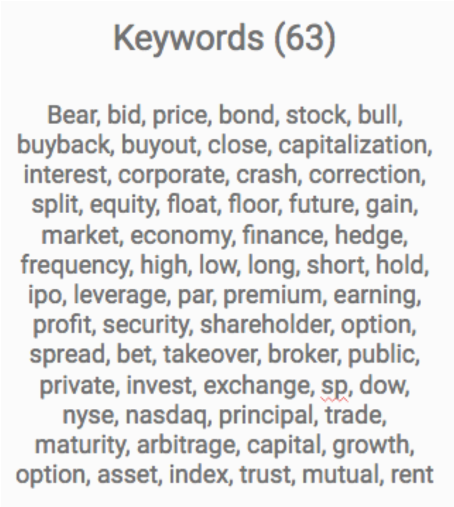
Data Pre-Processing

In order to remove noise inherent to social media data, several methods were used to filter and process the data gathered. First, the tweets/comments were filtered based on language so that only English text was used. Next, sentences were deconstructed into lists of words and stop words, including words such as 'the','is', and 'at', were removed. Finally, I used Porter stemming to stem each of the words in these lists. This process involves reducing words to their root form. For example, "argue", "argued", and "argues" would all be reduced to the stem "argu", where the stem itself is not necessarily a word but a root.

Word2Vec

In order to transform the text data into meaningful numeric representations, I used Google's Word2Vec model, which, as the name suggests, transforms words into vectors. Word2Vec does this by learning associations between words using n-grams (collection of 'n' number of words) and a collection of shallow neural networks (i.e. autoencoders). I trained my Word2Vec model using all of the text in my training data. As parameters, I used 300-length feature vectors with a 5 word context. What this basically means is that each vector representation generated will consist of 300 numbers, and the representations were learned using 5-grams.

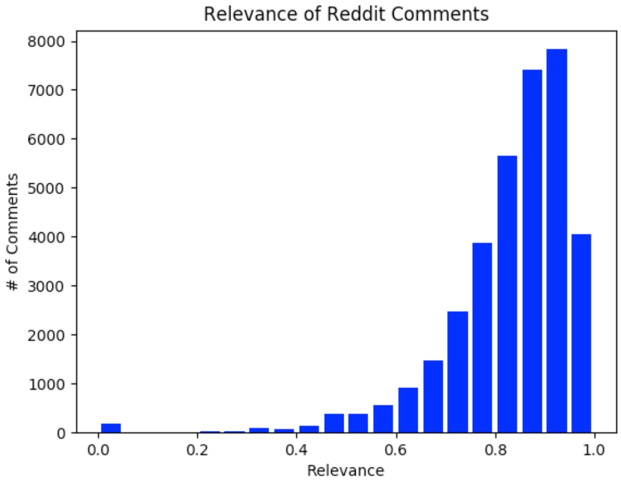
Something interesting that you can do with these representations is determine the similarity between words. Although it was not used specifically for this project, one application may be to determine the relevance of a specific text according to certain keywords. For example, if you use the following keywords:

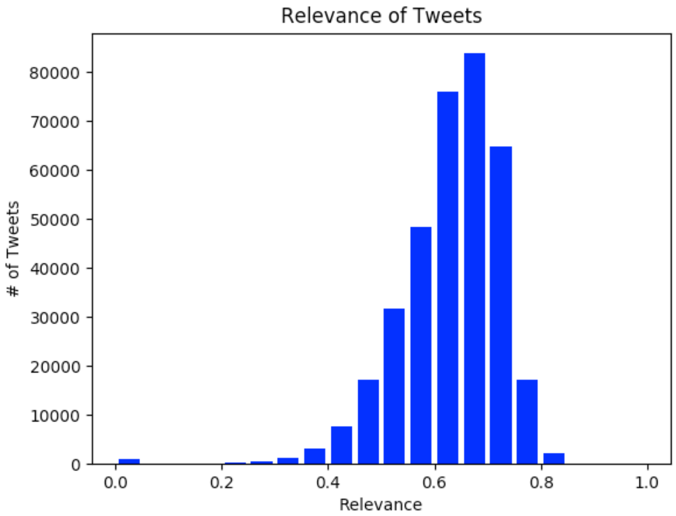


Keywords (63)

Bear, bid, price, bond, stock, bull, buyback, buyout, close, capitalization, interest, corporate, crash, correction, split, equity, float, floor, future, gain, market, economy, finance, hedge, frequency, high, low, long, short, hold, ipo, leverage, par, premium, earning, profit, security, shareholder, option, spread, bet, takeover, broker, public, private, invest, exchange, sp, dow, nyse, nasdaq, principal, trade, maturity, arbitrage, capital, growth, option, asset, index, trust, mutual, rent

You can examine each of the tweets or comments and determine their relevance to the general topic – ie stocks or finance – that these keywords are intended to represent. The way that I determined this relevance was by evaluating the pairwise similarity between each of the words in the text and each of the keywords. Then I normalized the result to a 0-1 scale based on how many words were found to be similar to a keyword compared to how many were not. The results are fairly interesting:

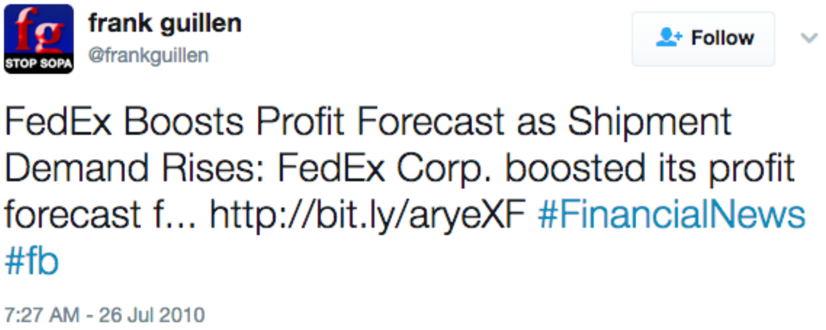




Here we see that the Reddit comments tend to be more relevant than the Twitter tweets. This may be due to the fact that, on average, tweets contain about one sentence per tweet, whereas Reddit comments average about 3.5 sentences per comment. Moreover, the difference in relevance may also be due to the interface itself. Namely, that Redditers post data that is more relevant, whereas Twitter has more irrelevant data.

Clustering (K-Means)

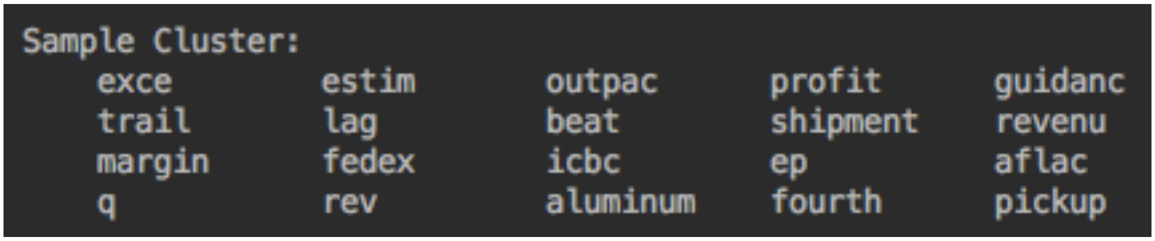
Once I was able to determine effective vector representations for words using Word2Vec, I performed K-Means clustering to group words together. For this task, I used 100 clusters and kept the best clustering assignment out of 10 random initializations. Once I determined which words were in each cluster, I was able to redefine the texts (i.e. tweets and comments) into a bag of centroids representation. A bag of centroids is simply a 100-length (number of clusters) vector where each number represents the number of words from the text belonging to that specific cluster. Using this bag of centroids representation, I was able to combine different tweets and comments from the same day into a single vector representation by adding them together. Finally, to account for a varying number of tweets/comments per day, I normalized the vector. This representation essentially represents the percentages of the text for that day coming from each of the clusters. If the clusters are insightful enough then this effectively reduces gobs of text to a small numerical representation which can be used to predict stock prices.



frank guillen
@frankguillen

FedEx Boosts Profit Forecast as Shipment Demand Rises: FedEx Corp. boosted its profit forecast f... http://bit.ly/aryeXF #FinancialNews #fb

7:27 AM - 26 Jul 2010

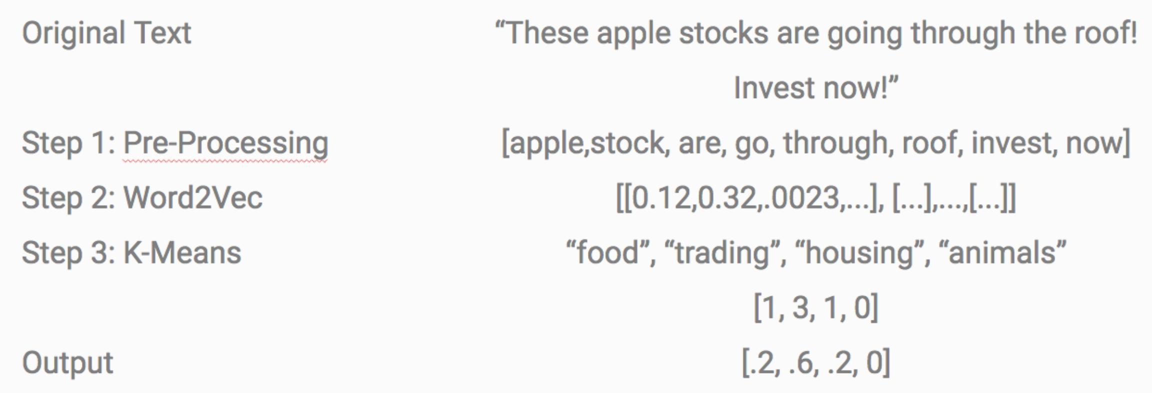


```
Sample Cluster:
     exce           estim          outpac         profit         guidanc
     trail          lag            beat           shipment       revenu
     margin         fedex          icbc           ep             aflac
     q              rev            aluminum       fourth         pickup
```

Using a bag of centroids representation, this example tweet was discovered to be the most relevant tweet to this specific cluster. As you can tell from the words in the cluster, one subject in the cluster has to do with profit predictions for shipment companies such as Fedex. This tweet talks about exactly that, making it very relevant for this cluster. In the stock predictions, having many of these kinds of tweets on a given day would result in a larger number for the value of this cluster in the feature vector.

Text Featurization – An Example

Many of the steps described may be hard to understand and are perhaps not altogether intuitive. For this reason, I've developed an example to help illustrate the process from text to vector:



Original Text	"These apple stocks are going through the roof! Invest now!"
Step 1: Pre-Processing	[apple,stock, are, go, through, roof, invest, now]
Step 2: Word2Vec	[[0.12,0.32,.0023,...], [...],...,[...]]
Step 3: K-Means	"food", "trading", "housing", "animals" [1, 3, 1, 0]
Output	[.2, .6, .2, 0]

In this example, the original text is first reduced to a list of words. Notice that the words “these” and “the” were removed because they are considered as stop-words. Also, “stocks” was changed to “stock” and “going” to “go” using Porter stemming. Next, these words are vectorized using Word2Vec. Each word is converted to a 300-length numerical vector, so in this case you would have 8 300-length vectors to represent each of the words gathered from pre-processing. Next, using K-Means on the feature vectors, you can determine which words are close to one another and group them into clusters. “Food”, “trading”, etc. are used as representative words for each cluster, but there are actually many more words in each cluster. Next, to get the bag of centroids representation you count the number of words from the text (ie using Step 1) that are in each cluster. In this example, “apple” falls naturally into the “food” cluster and “roof” into the housing “cluster”. “Stock” and “invest” belong in the “trading” cluster, and, perhaps because the clusters aren’t super specific, either “now” or “go” may have also been counted for that cluster. Of course, this isn’t a perfect example, but not all words are guaranteed to fall into a single cluster. Finally, you would sum up these bag of centroid representations by day and then normalize the results to get your final feature vector.

Formatting the Financial Data

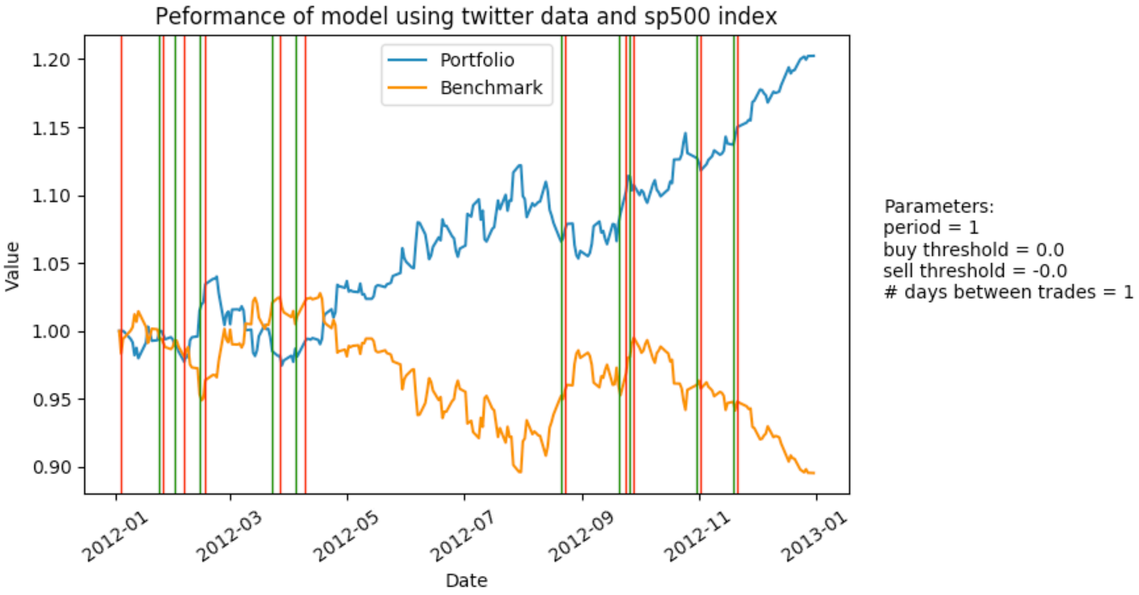
In order to get appropriate labels for my machine learning algorithms, I also had to process the financial data a bit. Given daily stock prices, I first calculated the n-day returns for each day. N-day returns are simply the expected return from looking ‘n’ days into the future. So if the stock is currently priced at $1, but is valued at $1.2 in 10 days, then the 10day return would be .2, or a 20% return. Based on these returns and specified buying and selling thresholds (ie how much of a return you need to expect before you want to buy or sell), I was able to get labels of -1,0,1 which represent the position that you want to be in on a given day. In other words, if you expect the stock price to go up the next day (1-day return) then the current day would have a value of 1, which means that you want to buy the stock so that you benefit from the price increase.

Machine Learning (Random Forests)

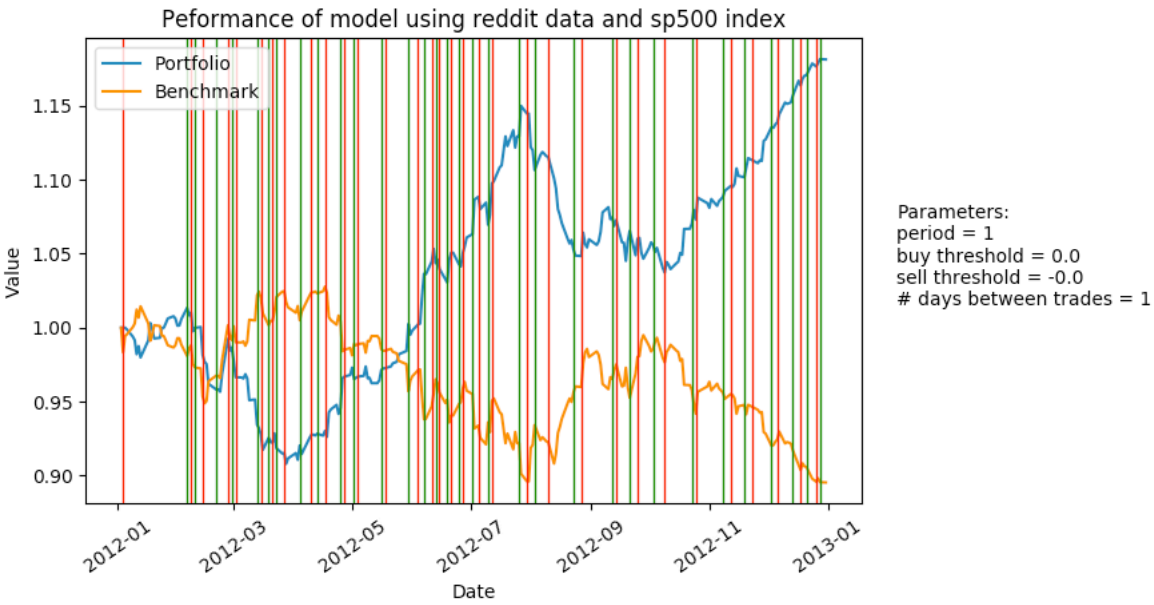
Finally, once the features and labels are appropriately calculated, I fed them into a Random Forest Classifier, which was able to determine if, on a given day, you should buy, sell, or hold (no position) a stock. For my forest, I used a collection of 100 tree with a sample split size of 5 and the minimum samples per leaf being 1. The random state was maintained throughout testing to ensure consistent results.

Results

Overall, the results were fairly surprising. While I had anticipated some rate of success, the effectiveness of using social media data proved to be much greater than I anticipated. For my experiments, I first adopted very basic assumptions: trading of a stock is allowed once per day and returns were forecasted for a single day only. In other words, this equates to predicting whether or not the stock will go up or down the next day and being able to trade on it the day before. Here are some of my results:

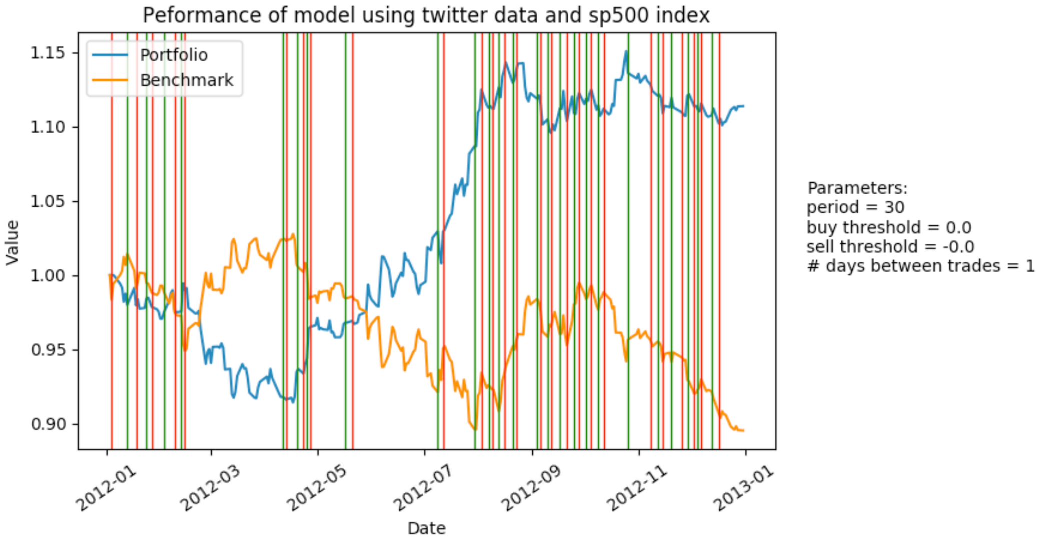


In this graph, the portfolio based off of the model that I developed is the blue line and the orange line indicates the benchmark stock index performance. Additionally, each red vertical line indicates selling the stock and each green line indicates a buying order. Here you can see that, using twitter data to predict the performance of the S&P 500 stock (SPY), I was able to significantly outperform the market index as well as get a return of roughly 20% over one year! This is a pretty amazing return if it is sustainable, and it importantly demonstrates that there are informative properties to social media data.



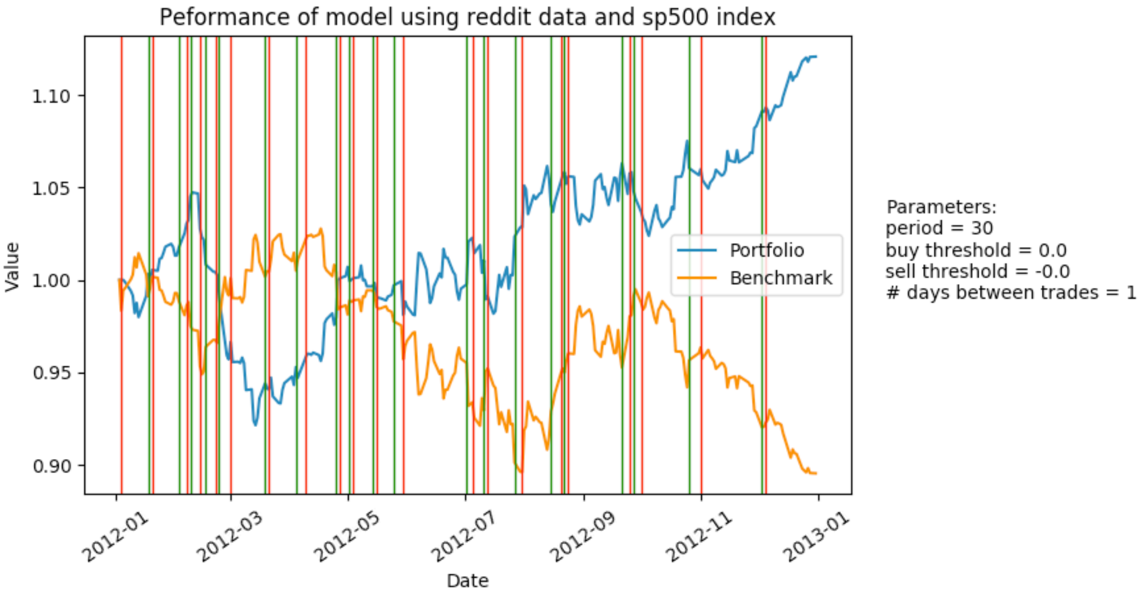
Parameters:
period = 1
buy threshold = 0.0
sell threshold = -0.0
# days between trades = 1

To compare against twitter data, I also tested using Reddit data. While the Reddit-trained model did not perform as well as the Twitter model, it still significantly outperformed the benchmark. However, it is important to note that this model performed many more trades than the other model. This could be due to a variety of reasons. The most likely reason is that there is not enough Reddit data to support the feature weights, and as slight variations in the feature weights can have a large impact on the performance of the classifier, the Reddit model will trade more frequently because there are more variation in the feature weights than in the Twitter model. This may be an important thing to consider, as each trade can cost a certain amount of money, so having fewer trades is preferable.



Parameters:
period = 30
buy threshold = 0.0
sell threshold = -0.0
# days between trades = 1

Additionally, I tested to see if the number of days used for the return forecast would affect the performance of the classifier. Determining the number of days to forecast returns an important factor to look at when training the machine learning model, as it affects the labels used. Overall, I found that performance varied as the range of days forecasted increased.


Peformance of model using reddit data and sp500 index

Parameters:
period = 30
buy threshold = 0.0
sell threshold = -0.0
# days between trades = 1

For both the Twitter and Reddit models, performance decreased as the range increased.

Discussion

Overall, the results gathered, while not necessarily conclusive, support the notion that relevant financial information can be gleaned from social media data to make relatively accurate stock price predictions. While these results are exciting, they have several limitations. The benchmark indices used for predictions are market-wide metrics, so it may be the case that social media data can only forecast large market movements rather than be used to predict whether the stocks of specific companies will change or not.

Additionally, several factors can significantly impact the performance of the model. First, the specific parameters used in the model can have a large impact on the performance. These parameters range from the text featurization paramaters to the testing parameters used. In particular, the testing parameters, such as the return forecasting period, the buying/selling thresholds, and the number of days needed to wait between trades have a much larger impact than other parameters because they directly influence either the labels or the trading decisions.

In addition to the potential for social media data to be used to effectively gauge changes in stock price, one of the contributions of this paper was to give a detailed explanation of how large groups of small texts, such as tweets and comments, can be aggregated into meaningful feature vectors used for machine learning. This, in particular, varies from traditional sentiment analysis and NLP methods that are focused on larger texts such as news articles and papers.

Future Work & Extensions

While the results from this project have shown some promise, there is still much more room for improvement. Extensions to this work may include:

- Using different analysis methods other than Word2Vec. For example, Latent Dirichlet Allocation (LDA) could be used to perform topic modeling on the text corpus, generating cluster representations similar to the process described in this project.
- Development of a comprehensive user interface in order easily test and tune hyperparameters. Refurbishing the code may help with building extensions to the current work.
- Integration of multicore processing to speed up computation. This extension will quickly become necessary with the addition of larger amounts of data.
- Real-time updates of the data. Ideally, it would be preferable to have a corpus of tweets and Reddit comments that expands over time, allowing for predictions in the present day.

With these extensions, this model will be more robust, facilitating more accurate stock predictions and social media insights. I hope to explore some of these extensions myself in my future work.

References

1. Asur, S., and B. A. Huberman. “Predicting the Future with Social Media.” *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*. Vol. 1. N.p., 2010. 492–499. *IEEE Xplore*. Web.
2. Barbier, Geoffrey, and Huan Liu. "Data mining in social media." *Social network data analytics*. Springer US, 2011. 327-352.
3. Borne, Kirk. “Top 10 Big Data Challenges – A Serious Look at 10 Big Data V’s.” *Converge Blog*. Web. https://www.mapr.com/blog/top-10-big-data-challenges-serious-look-10-big-data-vs.
4. Dos Santos, Cícero Nogueira, and Maira Gatti. "Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts." *COLING*. 2014.
5. Gruhl, D.,Guha, R., Kumar, R., Novak, J., and A. Tomkins, “The Predictive Power of Online Chatter,” in *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, New York, NY, USA, 2005, pp. 78–87.
6. Hootsuite, “125+ Essential Social Media Statistics Every Marketer Should Know,” *Hootsuite Social Media Management*, 30-Nov-2016. [Online]. Available: https://blog.hootsuite.com/social-media-statistics-for-social-media-managers/.
7. Himelboim, Itai; McCreery, Stephen; Smith, Marc. “Birds of a Feather Tweet Together: Integrating Network and Content Analyses to Examine Cross-Ideology Exposure on Twitter.” *Journal of Computer-Mediated Communication*, January 2013, Vol. 18, No. 2, 40-60.
8. Kim, Soo-Min, and Eduard Hovy. “Determining the Sentiment of Opinions.” *Proceedings of the 20th International Conference on Computational Linguistics*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2004. *ACM Digital Library*. Web. 6 Mar. 2017. COLING ’04.
9. Mittal, Anshul, and Arpit Goel. "Stock prediction using twitter sentiment analysis." *Standford University, CS229 (2011 http://cs229. stanford. edu/proj2011/GoelMittal-StockMarketPredictionUsingTwitterSentimentAnalysis. pdf)* 15 (2012).
10. Pak, Alexander, and Patrick Paroubek. "Twitter as a Corpus for Sentiment Analysis and Opinion Mining." *LREc*. Vol. 10. No. 2010. 2010.
11. Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. “Thumbs Up?: Sentiment Classification Using Machine Learning Techniques.” *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2002. 79–86. *ACM Digital Library*. Web. 6 Mar. 2017. EMNLP ’02.
12. Smith, Jerry. “Heilmeier Catechism: Nine Questions To Develop A Meaningful Data Science Project.” *Data Scientist Insights*. N.p., 11 June 2013. Web.
13. Taboada, Maite et al. “Lexicon-Based Methods for Sentiment Analysis.” *Comput. Linguist.* 37.2 (2011): 267–307. *ACM Digital Library*. Web.
14. “REST APIs.” *Twitter Developer Documentation*. Twiter, Inc. Web. https://dev.twitter.com/rest/public.

15. “Bloomberg API Documentation.” *Bloomberg Labs*. Bloomberg Finance L.P. Web. https://www.bloomberglabs.com/api/documentation/
16. “API Documentation.” *Reddit.com*. Reddit, Inc. Web. https://www.reddit.com/dev/api/.
17. “Technical Indicators and Overlays [ChartSchool].” N.p., n.d. Web. http://stockcharts.com/school/doku.php?id=chart_school:technical_indicators.