

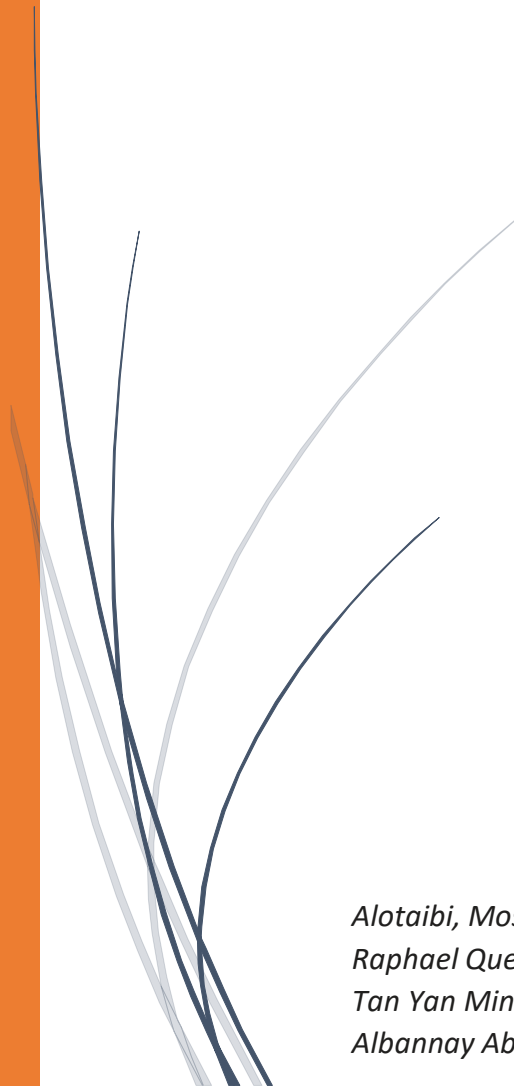


10/6/2019

IND5003 Data Analytics for Sense-making Final Project

Stock Price Movement Prediction Using Stocks Market

Data and Sentiment Analysis on Stocktwits Data



Alotaibi, Moshabbab Sowailem O (A0206910N)
Raphael Quek Hao Chong (A0139716X)
Tan Yan Ming (A0094598H)
Albannay Abdullah Abdulghani A (A0206922J)

Abstract

This paper shed some light on the thought process and an implementation method of stock price movement prediction using both stock market data and related opinions sentiment data. A brief description and comparison of some tools that are utilized at each stage of the data processing pipeline are presented. The data is aggregated and subsequently fed into different machine learning models to obtain a trained model that can be used to test the hypothesis of sentiment score implication on the stock movement. The results suggest that sentiment analysis improves the prediction and could achieve 80% accuracy of prediction using gradient boosting and deep learning models.

Key Words: Stock Market, Sentiment Analysis, Machine Learning, Web Scraping.

Table of Contents

Abstract	1
Introduction	3
Problem Statement.....	3
Main Aim	3
Hypothesis	3
Overview.....	3
Data Collection	4
Data Pre-processing.....	4
Sentiment Analysis Model Comparison	5
VADER.....	5
Training Vector Formation	6
Prediction.....	6
Results	7
Conclusion	7
Future Contribution	7
References.....	8
Additional Resources	9

Table of Tables

Table 1 Web Scraping Tools Analysis ^{3,4,5}	4
Table 2 Sentiment Analysis Models Comparison ^{8,9,10}	5
Table 3 Heuristic Rules for VADER ¹²	6

Table of Figures

Figure 1 Project Overview.....	3
Figure 2 Machine Learning Models Evaluation	7

Introduction

Sentiment Analysis refers to the use of NLP to identify and categorize emotions or tones expressed in any given piece of text ¹. Most people in business, and even politicians, often employ such a technique to guide their strategic planning by analyzing reviews, feedbacks, or comments that were targeted towards their services, products, or events from social media platforms such as Twitter or Facebook. Such a powerful approach could also be exploited to evaluate stock market sentiment via the use of StockTwits.com². StockTwits is a social media platform designed for sharing ideas between investors, traders, and entrepreneurs².

In this project, the objective is to apply sentiment analysis techniques from StockTwits' posts on Apple Inc. stock price to determine whether the general sentiments analyzed could be used to predict the price movement of the next day.

Problem Statement

Main Aim

The main aim is to predict if Apple's stock price would go up or down based on previous Apple's stock price data and to provide a prediction of next day market movement (up or down) of a particular stock that is traded on stocks market. Apple Inc. stock was selected to illustrate the concept. The prediction is obtained using stock market trading data in addition to the sentiment of StockTwits posts about Apple.

Hypothesis

The accuracy of the prediction of Apple's stock price movement would be improved using the sentiments expressed on StockTwits about Apple. The assumption is that the sentiments expressed in the StockTwits posts about Apple would be linked to Apple's stock price. This is based on the reasoning of how optimistic or pessimistic traders and investors feel about Apple's stock could reflect the state of the demand and supply which directly affects stock price.

Overview

The approach is taken consists of the following stages:

- Data Collection
- Data Preprocessing
- Sentiment Analysis
- Feature Vector Formation
- Prediction

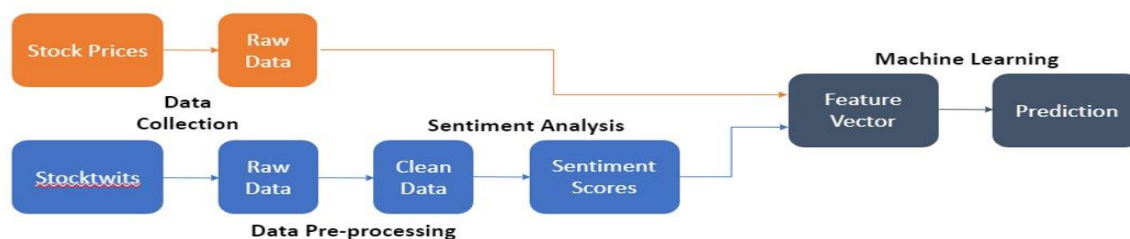


Figure 1 Project Overview

Data Collection

The team has evaluated several methods to draw data from StockTwits.com, such as the use of python script to access StockTwits' API and other existing tools or software for web scraping. However, scraping the data using python script is not an option anymore due to some unknown reasons as the API was no longer available to the public since a year ago. The other option to consider is using the tools available to scrape data from stocktwit.com. The team evaluated the available options and analyzed each tool strengths and weaknesses, how much data can we scrape using each tool. Below is the selection criteria matrix we used to select the best available scraping tool.

Tool/Software	Requires Installation	Subscription	Accuracy %	Operation System
<i>ScrapeStorm</i>	Yes	FREE	100%	All
<i>ParseHub</i>	Yes	Yes	100%	All
<i>webscraper</i>	No	Free for 1000 URLs	100%	Cloud

Table 1 Web Scraping Tools Analysis^{3,4,5}

Based on our project requirements, the team selected ScrapeStorm software. This software is developed by former Google search technology team³. It is a straightforward and powerful tool to scrape any data web page by identifying all the objects within this web page; also, user can use point and click on web page elements wants to extract it³. The tool can export the data to various formats like HTML and CSV. A total of 37259 Apple twit's data was scraped from Stocktwits.com for the period between 29Jul19 to 01Oct19. A list of scrapped fields is: (message, datetime, user).

For Apple Stock Prices data, we used a python library (*yfinance*) to get the data since Yahoo Finance decommissioned its historical data API⁶. This library is scraping data from Yahoo Finance return them into DataFrame Format. Data collected using this library covering the period from the for the period from 01Jan16 to 01Sep19.

Data Pre-processing

VADER was the model chosen to generate sentiment scores from each StockTwits post in our dataset. Appropriate pre-processing was performed the raw data to suit how VADER works. It considers the punctuation, letter case, emoticons, and degree modifier words when generating the sentiment score. As such, the pre-processing aimed to preserve these so that VADER could work as designed. The following pre-processing was done on the extracted raw StockTwits posts:

- Converted HTML character entity references to corresponding Unicode characters
- Removed links (starting with 'www.', 'http://', 'https://')
- Removed user handle tags (e.g. '@name123')
- Removed ticker symbols (e.g. '\$AAPL')
- Removed numbers
- Removed special characters ('@', '\$', '%', '/', '\', '_', '-')
- Replaced multiple spaces with a single space

Sentiment Analysis Model Comparison

Sentiment analysis on the StockTwits posts was chosen to be performed by pre-trained models. The bases of the decision are the lack of labeled StockTwits data to train a model. To optimize the results, different models were compared and the most suitable one in terms of accuracy and computation time was selected. There were four different models identified and tested as per the below table.

Since the analysis is based on StockTwits data, we applied them on a pre-labeled dataset of 3424 tweets from Twitter⁷. The tweets to be analyzed had a basic pre-processing done on them first. Using each model, the sentiment score for each tweet was computed. The accuracy was then calculated by taking the percentage of correctly labeled samples over the total number of samples. The time taken to generate the sentiment scores were measured as well. This was then divided by the total number of samples processed to get the average time per sample.

Model	Description	Accuracy (%)	Avg. Time/Sample (s)
<i>TextBlob</i>	Python library for processing textual data. It is a Naïve Bayes based classifier.	48.0	10.64
<i>VADER</i>	Lexicon and rule-based sentiment analysis tool adapted for social media	60.2	0.0005476
<i>Stanford CoreNLP</i>	Recursive neural network trained on a sentiment treebank which associates phrases from sentences with sentiment labels	72.0	2.92
<i>SentiWordNet</i>	Lexical resource that associates each word to a numerical score	32.8	0.009168

Table 2 Sentiment Analysis Models Comparison^{8,9,10}

The Stanford CoreNLP model achieved the highest accuracy of 72.0%. However, it took an average of 2.92 seconds to generate a sentiment score for one data sample. Hence it was not feasible to use the Stanford CoreNLP model to compute all the sentiment scores for the full dataset of 37259 Stocktwits posts. Ultimately, VADER, which had the second-highest accuracy of 60.2%, and an acceptable average time taken to compute a sentiment score for one data sample of 0.0005476 seconds, was selected as the sentiment analysis model to use.

VADER

VADER is an acronym for Valence Aware Dictionary for sEntiment Reasoning and is a pre-built sentiment analysis model included in the NLTK package that is specifically attuned to examine sentiments expressed in the short texts from social media platforms^{11,12}. It is also included in the NLTK package that we learned in lesson 4 and could be imported into the script to be used like any other packages.

Unlike most sentiment analysis models, VADER's lexicon is entirely built by humans instead of using codes and machines¹. Since it is extremely labor-intensive to construct a lexicon manually, a group of individuals was independently hired in the crowdsourcing e-platform, Amazon Mechanical Turk, to examine around 9000 words that are usually found in several social media platforms¹². Each of these human raters was then tasked to create a dictionary of their own by appending a sentiment value for every 9000 words, ranging from -2.5 and 2.5, to describe their respective sentiment polarity and intensity¹². In attempt to reduce any form of biases or human elements, each of these dictionaries were then aggregated to create the final VADER lexicon with a length of 7500 words¹².

Besides, the VADER sentiment analysis model was built to follow five heuristic rules to determine the sentiment of any given text (Table 3). On top of those five rules, VADER model also considers other types of non-words features, such as emojis, slangs, emoticons, that are usually removed for other sentiment analysis models¹². Because of such acceptances, a more accurate sentiment representation of a text can be obtained, and there is no need to clean any raw text extensively to form a proper English sentence. Hence, the extracted StockTwits raw data were cleaned using the conditions mentioned in the section "Data Pre-processing."

No	Criteria	Example
1	Punctuation	<i>"I'm hungry vs I'm hungry!!"</i>
2	Capitalization	<i>"I'm hungry!! vs I'M HUNGRY!!"</i>
3	Degree modifiers	<i>"I WANT TO EAT!! VS I REALLY WANT TO EAT!! "</i>
4	Conjunctions	<i>"I love pizza, but I really hate Pizza Hut."</i>
5	Preceding Trigram	<i>"Canadian Pizza is not really all that great."</i>

Table 3 Heuristic Rules for VADER ¹²

However, before the cleaned texts were used to generate their sentiment scores, we took an additional step to update the VADER lexicon with other external sources of lexicons that are often used in financial applications or businesses, such as the Loughran-McDonald Financial Sentiment Word Lists¹³. The update adjusted the values of most words in the existing VADER lexicon and increased the length of words from 7500 to 8600.

After the lexicon was updated, the VADER model was used to generate the polarity scores for all the pre-processed data. The scores are represented in the range of -1 to 1 , where the sign characterizes the polarity as positive or negative emotion, while the absolute value implies the intensity¹². However, it was noticed that those texts that return exactly 0.0000 scores are just gibberish and were thus removed. The final dataset has a length of 34875 which is approximately 93.6% of the raw data extracted from StockTwits. The sentiment scores were subsequently aggregated into averages of every hour for each day. These processed data, along with the stock price data extracted from *yfinance*, were then passed on to train several prediction models.

Training Vector Formation

The training vector was built using the data obtained from *yfinance*, the previous five days closure values and sentiment analysis value obtained by VADER aggregated per hour. Subsequently, features are passed to multiple machine models to train them to predict next day market movement for the stock (up/down). Different features and different periods were passed to the models to test the effect of including and excluding sentiment analysis on prediction accuracy.

Prediction

Prediction using different machine learning was performed on the test sets, and the results were compared to select the best models and to form the future contribution assumptions. The feature vectors were split into training and test data and were used to train various models. The models were also trained on just stock price data alone for comparison.

Results

The below figure summarizes the prediction accuracies obtained from the different machine learning models. The full period includes the stock market features only without aggregated sentiments per hour, while the partial period includes all the features. The full period is extended over three years while the partial is covering 59 days only due to the lack of StockTwits for the full period.

As can be seen the highest performance for full period was obtained by deep learning model (Sequential with 3 layers). However, the partial period with sentiment compared to without sentiment showed improvement once sentiment is incorporated in training for all models except for KNN with the default number of neighbors. The best performing models with sentiment analysis scoring included were gradient boosting model (GBM) and deep learning.

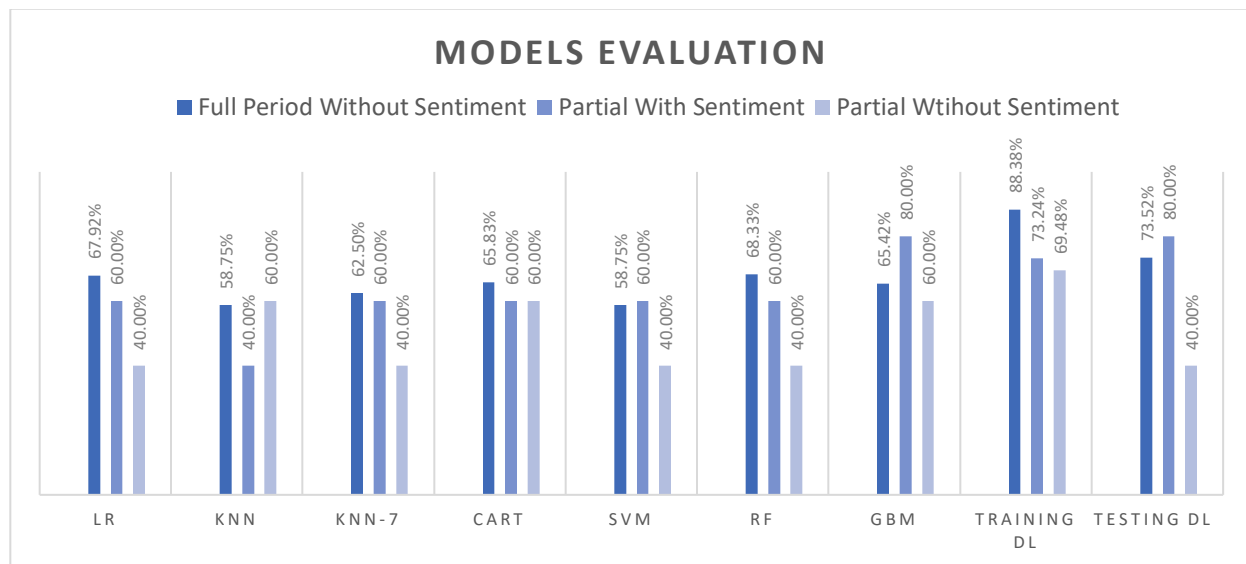


Figure 2 Machine Learning Models Evaluation

Conclusion

The concepts of sentiment analysis were explained along with highlights on the empowering tools to perform such a task. These tools included web scraping tools used for web data extraction, trained sentiment analysis models, stock prices available libraries, models utilized to evaluate the accuracy of predictions. The final results showed the stock movement could be predicted with accuracy of 80%. This claim needs to be further evaluated with more data. However, it is showing promising results with a comparatively small test and training data.

Future Contribution

The lesson learned from the project provided the team with an insight on what to improve in the next possible iterations. Firstly, improving the data acquisition process to include more data that covers wider scope such as more stocks, more markets and extended period for StockTwits texts. Secondly, some processes in the workflow can be automated such as automating the extraction, cleaning, evaluation, and aggregation. This also can be improved using one pipeline to perform the whole process in one go. Thirdly, time-series forecasting can be used to predict the actual price on the following day(s) instead of prediction the movement solely. Finally, more specialized machine learning models that can remember previous closure data can be utilized to improve the storage parameters and to improve the accuracies.

References

1. Mark, Beccue. (2017). *New Frontiers in Natural Language Processing: Sentiment Analysis Is the Key to New Insights*. Retrieved from <https://www.tractica.com/artificial-intelligence/new-frontiers-in-natural-language-processing-sentiment-analysis-is-the-key-to-new-insights/>
2. Huang, D. (2015). *Retail Traders Wield Social Media for Investing Fame*. *Wall Street Journal*. Retrieved from <https://www.wsj.com/articles/retail-traders-wield-social-media-for-investing-fame-1429608604>
3. ScrapeStorm. (2019). Introduction to ScrapeStorm. Retrieved from <https://www.scrapestorm.com/tutorial/introduction-to-scrapestorm/>
4. ParseHub. Parsehub Tool Mainpage. Retrieved from <https://www.parsehub.com/intro>
5. Webscraper. Webscraper Tool Mainpage. Retrieved from <https://www.webscraper.io/documentation>
6. Pypi. (n.d.). yahoo-finance 1.4.0 Project Description. Retrieved from <https://pypi.org/project/yahoo-finance/>
7. Prelabeled twitter data. (n.d.). Retrieved from https://github.com/zfz/twitter_corpus
8. Advanced Usage: Overriding Models and the Blobber Class. (n.d.). Retrieved from https://textblob.readthedocs.io/en/dev/advanced_usage.html#sentiment-analyzers
9. Deeply Moving: Deep Learning for Sentiment Analysis Stanford NLP. (n.d.). Retrieved from <https://nlp.stanford.edu/sentiment/>
10. SentiwordNet. (n.d.). Retrieved from <https://github.com/aesuli/sentiwordnet>
11. Loper, E., & Bird, S. (2002). NLTK: the natural language toolkit. *arXiv preprint cs/0205028*.
12. Hutto, C., & Gilbert, E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Eighth International Conference on Weblogs and Social Media (ICWSM-14).
13. Loughran, T., & McDonald, B. (2011). When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks. *The Journal of Finance*, 66(1), 35-65.

Additional Resources

1. All project files and codes are available at github in the following repository:
<https://github.com/IND5003-Project/main/tree/master/FINAL%20FILES%20TO%20SUBMIT>
2. Additional Resources from the literature related to the stock market with sentiment analysis can be found under our github repository:
<https://github.com/IND5003-Project/main/tree/master/Resources>
3. Additional Twitter open source data can be found on the below resource:
<https://github.com/shaypal5/awesome-twitter-data>