Milestone Report: NLP Market Predictions

Luke Jordan
Machine Learning CS5824
bruceli@vt.edu

Abstract

The financial industry's use of indicators to forecast the movement of markets contains a fatal flaw. It forecasts purely based on stock data and not qualitative data. It is no argument that the news influences the market, and the news reports on the market. Most people make financial decisions through the news, the mob, personal experiences, and word of mouth–especially as investing is becoming more easily accessible. Thus, in this study, I tested various machine learning algorithms to forecast market movements before the market opens the following day. This report will specifically cover a binary classification on whether a stock will be red or green, negative or positive, the following day once the market closes. This report clearly states the current problem statement, industry limitations, the environmental setup, the results, the analysis, and the further work. This report shows that despite the initial algorithm's good performance with familiar articles, algorithms struggle when faced with articles that are unfamiliar. The root cause of this failure is poor training data.

1 Overview

The aim of this project is to develop an effective and unique trading bot algorithm. A trading algorithm is one that can make reasonable decisions dependent on the stock market or cryptocurrency trades in order to garner a profit. It will utilize machine learning in order to make its predictions, and its success will be measured on correct financial decisions (i.e. positive or negative or how much % growth). As of right now, this algorithm will explore a facet of market analysis that is hardly touched upon, which is natural language processing in order to determine the trend of the market. The Overview section will go over the problem statement, industry limitations, and solution statement.

1.1 Problem Statement

Investing has never been more accessible than it has been now, combined with the craze of cryptocurrency and the news of stock market success stories, it has created a rush of young investors looking to secure themselves a brighter future. However, the stock market can be turbulent and unforgiving. There are hundreds of financial indicators that forecast trends utilizing stock market data, but most people have neither the time nor know-how to take full advantage of such indicators. Instead, most people rely on the news, the mob, their own experiences, and the word of mouth to determine which stocks to invest in. The industry's indicators fail to take into account this qualitative data, and instead focus on quantitative data. As a result, current algorithms fail to predict the prologue of market movements and can only predict market trends when they are already set in motion.

1.2 Industry Solution Limitations

Industry solutions often use "indicators" to predict market trends. An indicator is defined as a measurable characteristic of some data that can reliably forecast trends. For instance, the Ichimoku Cloud is an indicator that gives a prediction of when to buy and sell stock. Its smaller indicators forecast resistance, momentum, and trend. The problems surrounding this approach are threefold:

1. Indicators focus on quantitative data, and completely ignore the qualitative data, such as news articles, which can provide a wealth of information before the market reacts.

2. Indicators forecasts all sorts of things that can be overwhelming and time consuming. For instance, indicators include but are not limited to the Moving Average, which forecasts possible future patterns, the Stochastic Oscillator, which determines whether a market is overbought or oversold, and the Bollinger Band, which forecasts long-term price movements. There are hundreds more, and analysis of all these variables can be time consuming.

3. There exists hundreds of indicators to analyze market trend, and most people have neither the time nor knowledge to choose and analyze indicators to come to a decision on whether to buy or sell stock. These types of people tend to rely on the news, feelings, and word of mouth to help them indicate whether to buy or sell.

In general, the limitations of indicators is that it is purely quantitative and not qualitative. Current solutions cannot predict the prologue to such a market movement because the prologue to market movements stem from the news, the mob, personal experiences, and word of mouth.

1.3 Solution Statement

1.3.1 This Report's Solution Statement

Due to time constraints, this report will only tackle a fraction of the end-goal (See Section 1.3.2). The goal for this report is to reliably determine whether the market the following day or same day is in the green or red–up or down–positive or negative. News articles are the only source of information that the algorithm may use to determine the market trend. If the algorithm could reliably do this, then there are some patterns in the training data that can reliably lead to the development of a mutliclass classifier that can give a range of how much the market may change.

1.3.2 The End-Goal Solution Statement

This study aims to train a model using a sample of historical news articles about a particular stock, in order for the model to give financial advice that will inform the user about potential market trends. The final model will ultimately provide some range of expected percentile change for the next day (multiclass classification). Reaching this point in the research would be considered a success, as an average person can make an informed decision whether to buy or sell a stock if given a range of how much a stock is expected to change the next day. If there is enough time, an even better model would be able to take information about its qualitative assessment and incorporate it into existing indicators in order to determine the percentile change. Instead of being purely qualitative, the model would then be a mixture of qualitative and quantitative traits, perhaps making the forecasts more accurate.

The project's solution statement is to determine whether or not it is possible to predict a stock going up or down in growth (binary classification) the same day and next day. If given more time, further work would include building a multiclass classifier that gave a range of percentile change. More on this later.

2 Environmental Setup

2.1 Aggregating Training Data

The objective of this report is to determine whether or not news articles can forecast negative and positive growth (binary classification) for the following day. I chose to use NVIDIA's historical data (collected from: https://www.nasdaq.com/market-activity/stocks/nvda/historical) as the price of its stock has been turbulent the past six months (as of April 2022). The information in this data set is then expanded to include news articles. News articles were collected from the front page of Google News about NVIDIA. Google News can conveniently filter based on time of creation, and it can features a wide range of sources that contain a wealth of information about the stock. With each article on the front page, raw HTML paragraphs from the front page websites were gathered. To circumvent the issue of paid media or advertisements, HTML paragraphs that did not contain the word "NVIDIA" were ignored.

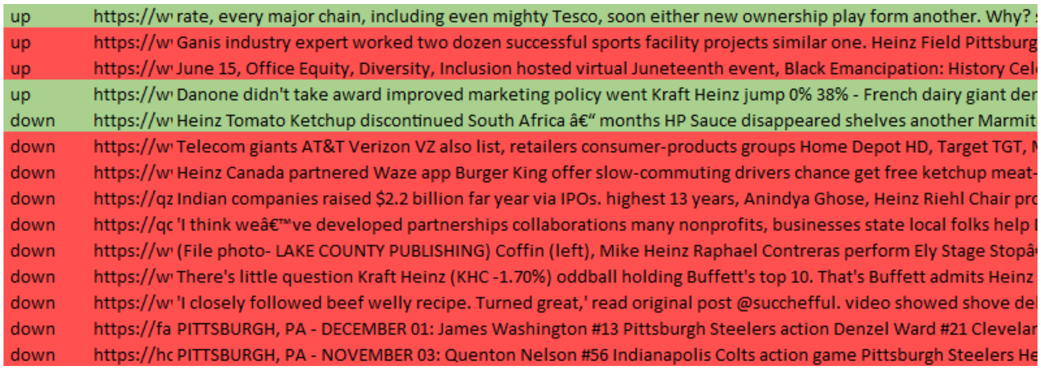
2.2 Training Datasets

Many training datasets were made, but the report will only reference these three datasets. From left to right, I have put the dataset name, the day it is predicting, and the description of the dataset.

Given Data Set / Test and their Descriptors		
Test Name	Pred. Day	Description
test_same_day	Same	Same day articles predicting same day movement.
test_skip_fridays	Next	Does not include closed days (Weekends) and Fridays.
test_include_closed	Next	Includes Fridays and Weekends. Their test value is the following Monday.

Figure 1: Given Data Set / Test and their Descriptors

It came to my attention late into the project that my training data was incredibly unreliable. This is foreshadow to the results that you are about to see. As a bit of an explanation as to why the training data is unreliable, one must understand the nature of Google News. Firstly, Google does not know I only want news pertaining to the markets, and even if I type in "NVIDIA stock", Google will return any news article that contains those keywords. This leads to a plethora of unrelated news articles, and since most financial media requires a paid subscription, a lot of the data I sought were unaccessible to me.



up	https://w	rate, every major chain, including even mighty Tesco, soon either new ownership play form another. Why?
up	https://w	Ganis industry expert worked two dozen successful sports facility projects similar one. Heinz Field Pittsburg
up	https://w	June 15, Office Equity, Diversity, Inclusion hosted virtual Juneteenth event, Black Emancipation: History Cel
up	https://w	Danone didn't take award improved marketing policy went Kraft Heinz jump 0% 38% - French dairy giant der
down	https://w	Heinz Tomato Ketchup discontinued South Africa â€" months HP Sauce disappeared shelves another Marmit
down	https://w	Telecom giants AT&T Verizon VZ also list, retailers consumer-products groups Home Depot HD, Target TGT,
down	https://w	Heinz Canada partnered Waze app Burger King offer slow-commuting drivers chance get free ketchup meat-
down	https://qz	Indian companies raised $2.2 billion far year via IPOs. highest 13 years, Anindya Ghose, Heinz Riehl Chair pro
down	https://qc	'I think weâ€™ve developed partnerships collaborations many nonprofits, businesses state local folks help
down	https://w	(File photo- LAKE COUNTY PUBLISHING) Coffin (left), Mike Heinz Raphael Contreras perform Ely Stage Stopâ
down	https://w	There's little question Kraft Heinz (KHC -1.70%) oddball holding Buffett's top 10. That's Buffett admits Heinz
down	https://w	'I closely followed beef welly recipe. Turned great,' read original post @succheffull. video showed shove de
down	https://fa	PITTSBURGH, PA - DECEMBER 01: James Washington #13 Pittsburgh Steelers action Denzel Ward #21 Clevelar
down	https://hc	PITTSBURGH, PA - NOVEMBER 03: Quenton Nelson #56 Indianapolis Colts action game Pittsburgh Steelers He

Figure 2: A Significant Chunk of Training Data is Unreliable (Heinz)

Take for instance this chunk of news articles gathered from Google News pertaining to Heinz stock. Google News does not know I only want stock data, and even if I search for "Heinz stock", Google News will return data pertaining to sports articles. The reason why sports is related to Heinz stock is because there is a Heinz sports stadium. This battle between me and Google, where I wanted one thing but Google recommended me another, was frustrating. Putting up filter words still lead to unrelated articles leaking into the training algorithms. .

3 Process

I specifically wanted to train the data with the Naive Bayes (NB) model, as Naive Bayes performs extremely well at classifying text. I hypothesized that this logic would carry over to determine whether a stock's growth would be positive or negative, as determining stock trends from news articles uses textual input and a binary classification. Furthermore, textual data can contain an enormous and unknown distribution, and Naive Bayes tackles this by weighing certain known words with the hopes that even with unknown words it performs accurately. Demonstrated by Prof. Jin in class, email spam/ham classification can have an accuracy of up to $99\%+$ using the Naive Bayes classifier.

I also was curious about the performance of other algorithms in performing binary classification. I additionally tested LGBM Classifier, Linear SVC, Logistic Regression, and Random Forest Classifier. I wanted to test these algorithms to see how well they'd perform in conjunction with NB.

My experiment went in two phases. The first phase was training the NB model and collecting its training accuracy. In other words, I wanted to train the model and check the accuracy against data it has already seen. I wanted to see how well the model can perform when classifying from a known distribution.

The second phase was training the NB model and collecting its test accuracy. In orther words, I wanted to train the model and check the accuracy against data it has *never* seen. I wanted to see how well the model can perform when classifying from an *unknown* distribution. Furthermore, as stated above, I trained other models based on different algorithms and tested their accuracy with an unknown distribution.

Each phase and each model I trained had the same preprocessing of the news data. The text was cleansed of unnecessary punctuation and stop words, and then converted to a matrix of token counts. The matrix was then transformed to a normalized term-frequency representation. This representation is known to be good at text classification. Finally, the Multinomial Naive Bayes model was trained with the term-frequency representation as the X and the market movement as the Y. A confusion matrix was then used to test the model.

4 Results

Each data set is its own unique test. See Figure 1 for more details on what each data set tests. These results were gathered from a Confusion Matrix. The Correctness follows the formula $Correct = \frac{\text{True Pos.+True Neg.}}{\text{num of all data points}} * 100\%$. The Error follows the formula $Error = \frac{\text{False Pos.+False Neg.}}{\text{num of all data points}} * 100\%$

4.1 Phase 1: Training Accuracy

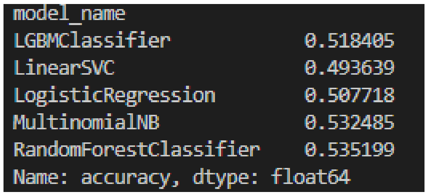
In Phase 1, only the Naive Bayes algorithm was trained to fit the data. I was curious about the training accuracy that could be achieved.

Given Data Set / Test and their Results						
Test Name	True Pos.	True Neg	False Pos.	False Neg.	Correct (%)	Error (%)
test_same_day	205	271	52	2	89.8	10.2
test_skip_fridays	180	222	24	1	94.15	5.85
test_include_closed	251	316	44	3	92.35	7.65

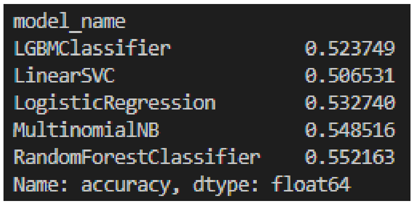
Figure 2: Naive Bayes Model Training Accuracy with Given Data Set and its Result

4.2 Phase 2: Test Accuracy

In Phase 2, many models were trained and tested with an unknown distribution.

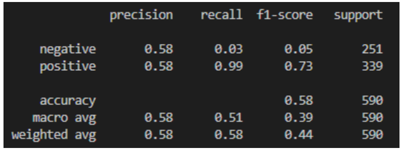


```
model_name
LGBMClassifier            0.518405
LinearSVC                 0.493639
LogisticRegression        0.507718
MultinomialNB             0.532485
RandomForestClassifier    0.535199
Name: accuracy, dtype: float64
```



```
model_name
LGBMClassifier            0.523749
LinearSVC                 0.506531
LogisticRegression        0.532740
MultinomialNB             0.548516
RandomForestClassifier    0.552163
Name: accuracy, dtype: float64
```

Figure 3: Test Accuracy, Same Day Pred. | Figure 4: Test Accuracy, Next Day Pred.



```
              precision    recall  f1-score   support

    negative       0.58      0.03      0.05       251
    positive       0.58      0.99      0.73       339

    accuracy                           0.58       590
   macro avg       0.58      0.51      0.39       590
weighted avg       0.58      0.58      0.44       590
```

Figure 5: Naive Bayes Stats Predicting the Following Day

5 Analysis

Section 5.1 covers performance accuracy of Naive Bayes (NB) against a known distribution–data it has seen during training. Section 5.2 covers performance accuracy of models against an unknown distribution–data it has never seen during training. Section 5.3 covers further work needed.

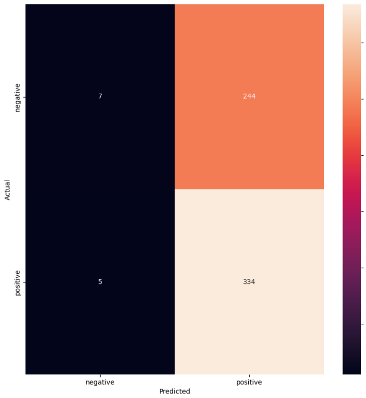
5.1 Analysis of Phase 1

Training the Naive Bayes model and testing it with a *known* distribution resulted in respectable accuracies ranging from 89.8% to 94.15%, seen in Figure 2. The accuracy for testing same day market movement had the lowest accuracy at 89.8% correct. The accuracies were higher when predicting the following day's market movement. When Fridays as well as Weekends were skipped, the accuracy was 94.15%. When Fridays and Weekend data was included, the accuracy was 92.35%. These accuracies are not close to the 99%+ of spam classification using the same algorithm. However, the Naive Bayes model did find a strong trend in the training data, so long as the data it is predicting has been seen before.

5.2 Analysis of Phase 2

This is when the results took a significant turn. Training the Naive Bayes model and testing it with an *unknown* distribution lead to significantly diminished accuracies when compared to that of Phase 1. This makes sense. Typically the training accuracy is higher than the testing accuracy. However, I did not expect such a significant and huge difference from the training accuracy. I initially through that the Naive Bayes Model was perhaps overfitting. To test this hypothesis, I wanted to fit four other algorithms to the training data and gather their test accuracy. The results show that all of the algorithms, when trained with the training data and tested with an unknown distribution, had an accuracy that was equivalent to a weighted coin flip. The most accuracy I was able to get was with the Naive Bayes Model at 58% accuracy.

Interestingly, Figure 3 and 4 show that predicting the following day's market trend has a greater accuracy than the same day's market trend. Figure 2 of Phase 1 agrees with that sentiment. Another interesting observation to note is that the Naive Bayes algorithm had a negative f1-score of 5% and a positive f1-score of 73%, shown in Figure 5. In other words, the Naive Bayes algorithm significantly favored predicting positive growth, and performed absolutely horrendous at predicting negative growth.



To make Figure 5 more easily understandable, I created the confusion matrix you see on the right. There were 244 times where the NB model predicted positive yet it was negative. Out of the 590 articles tested, only 11 were thought to be negative. These results are odd, and Figure 2 from Phase 1 foreshadowed such results, because even it struggled to classify negative data–however, not to this extent.

As stated in Section 2.2 on aggregating the training data, I believe that the training data is partially responsible for the unexpected results. Upon further examination of what Google News returned to me, I realized that an estimated 30%-40% of the data provided was unrelated and at some points contained ads. This leads me to my next section on further work

5.3 Further Work

Now that I have analyzed the results from Phase 1 and Phase 2, I want to discuss further work. One of the biggest issues I had training and testing the models was the training data. The models are only as good as the quality of the training data. Unfortunately for me, Google News happens to contain a plethora of unrelated articles and ads. Even with filtering a majority of such text out, many still somehow end up in my dataset. As seen in Figure 2, Google provides news articles for a wide range of topics, including sports, which can hurt the training algorithm if its focused on finances. Future work should include gathering data from handpicked reliable sources, which include those that have paid subscriptions. That way it bypasses Google's desire to recommend all types of articles.

6 Conclusion

I was not able to reliably determine whether the market would go up or down. However, I was able to identify that a large source of error came from the training data. The model is only as good as the quality of training data provided. I believe that despite a 58% accuracy and the unexpected results, it is possible to have that accuracy increase significantly if trustworthy reliable training data is gathered.

7 Contributions

1. Luke Jordan